Causal Investigations in Interactive Semantics

Pierre Clairambault

▶ **To cite this version:**

Pierre Clairambault. Causal Investigations in Interactive Semantics. Computer Science and Game Theory [cs.GT]. Aix-Marseile Université, 2024. tel-04523273

HAL Id: tel-04523273
https://hal.science/tel-04523273v1

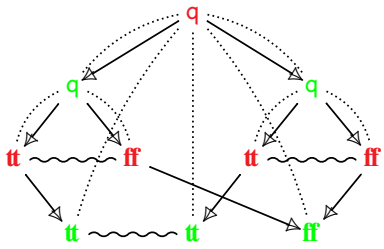
Submitted on 27 Mar 2024

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Distributed under a Creative Commons Attribution 4.0 International License

Causal Investigations in Interactive Semantics



Mémoire présenté par

Pierre Clairambault

en vue d'obtenir

l'Habilitation à Diriger des Recherches
(Informatique)

le 21 février 2024 devant le jury composé de:

Dan Ghica	Huawei, University of Birmingham	Rapporteur
Delia Kesner	IRIF, Université Paris Cité	Examinatrice
Ugo Dal Lago	University of Bologna	Examinateur
Damiano Mazza	CNRS, LIPN, Université Paris Nord	Président
Guy McCusker	University of Bath	Rapporteur
Paul-André Melliès	CNRS, IRIF, Université Paris Cité	Rapporteur
Laurent Regnier	Aix-Marseille Université	Examinateur

Aix*Marseille université
Socialement engagée

LIS LABORATOIRE D'INFORMATIQUE & DES SYSTÈMES

Remerciements

Tout d'abord, merci à tous mes collaborateurs passés sur ce formidable thème des jeux concurrents. Merci à Glynn pour m'avoir fait découvrir les structures d'événements et avoir accompagné mes premiers pas dans cette belle direction de recherche. Merci à Simon C.: nos séances de travail sur les jeux concurrents fins et l'innocence parallèle resteront parmis mes plus beaux souvenirs de collaboration scientifique. Merci à Aurore pour m'avoir accompagné sur les traces de Herbrand. Merci à Hugo pour une touche de probabilité, et pour m'avoir convaincu que les bicatégories, c'est après tout peut-être pas complètement inutile. Merci à Marc d'avoir insisté pour faire du quantique, menant à, à mon avis, une des plus belles applications des jeux concurrents. Merci à Lison et Lionel grâce à qui j'ai enfin pû comprendre le développement de Taylor (c'est de la sémantique des jeux). Merci à Simon F., grâce à qui on peut maintenant faire des jeux concurrents fins sans jeux concurrents.

Merci à tous les collègues de l'équipe Plume du LIP de m'avoir patiemment écouté parler de structures d'événements pendant toutes ces années, et en particulier Colin qui ne pouvait pas s'enfuir puisque nous partagions le même bureau. Merci à tous les collègues marseillais, du LIS et de l'I2M, d'avoir pris le relais, de m'avoir si bien acceuilli lors de mon arrivée à Marseille, et de contribuer à un cadre de vie et de travail exceptionnel – en particulier Raphaëlle, qui a brillamment pris la suite de Colin.

Merci à Dan, Guy et Paul-André d'avoir accepté de rapporter ce travail, malgré son volume déraisonnable: vous êtes sûrement les trois chercheurs dont le travail a le plus influencé ma vision scientifique, et je vous suis reconnaissant de votre relecture approfondie de ma contribution. Merci à Delia, Ugo et Damiano d'avoir bien voulu être membres de ce jury, et merci à Laurent de m'avoir guidé dans cette aventure de l'habilitation.

Merci à tous ceux que j'oublie et qui regrettent de ne pas trouver leur nom ici.

Et pour tout le reste, merci à Marie-Laure.

Résumé

La *sémantique des jeux* est un cadre mathématique puissant pour raisonner de façon compositionnelle sur les programmes. Contrairement aux méthodes plus classiques de sémantique dénotationnelle, elle ne voit pas les programmes simplement comme des fonctions mathématiques prenant une valeur en entrée et rendant une valeur en sortie, mais comme des processus dynamiques et interactifs. Elle capture de façon compositionnelle le flot de contrôle même pour des programmes disposants de primitives calculatoires complexes: ordre supérieur, état mutable, opérateurs de contrôle, etc.

Mais la sémantique des jeux n'est pas une *théorie mathématique*: c'est plus une constellation de modèles disparates sans réel liant autre que le folklore de quelques experts du domaine. L'un de ces modèles disparates est celui des *jeux concurrents*, qui hérite des jeux concurrents d'Abramsky et Melliès et des jeux asynchrones de Melliès. Dans sa formulation récente proposée par Rideau et Winskel, ce modèle est basé sur les *structures d'événements*, un modèle de *concurrence vraie* qui évite de supposer l'existence d'une horloge globale et représente plutôt les programmes par leur structure causale.

Bien entendu, les jeux concurrents permettent de modéliser la programmation concurrente. Mais dans ce manuscrit, je montre que plus qu'un modèle comme un autre, leur expressivité permet de mettre en avant des structures implicites dans plusieurs autres modèles de la littérature pour des langages concurrents, ou pas. Ce faisant, sur les traces de la vision d'Abramsky et Melliès, ils permettent une synthèse jusqu'alors absente et contribuent à faire de la sémantique des jeux une théorie mathématique.

Concrètement, ce manuscrit commence en Partie I par une introduction détaillée à différentes sémantiques des jeux traditionnelles *à pointeurs*: d'abord pour le langage fonctionnel paradigmatique PCF, son extension IA avec mémoire mutable, puis le modèle de jeux non alternants pour son extension concurrente $\mathsf{IA}_{/\!/}$. En Partie II, je donne une présentation complète des jeux concurrents et de leur extension avec symétrie. En partie III, je relie les jeux concurrents aux différents modèles de la Partie I – mais aussi d'autres, tels que le modèle relationnel – via des foncteurs préservant l'interprétation. Finalement, en Partie IV, je passe en revue d'autres développements en jeux concurrents, et conclus sur des problèmes ouverts et perspectives futures.

Abstract

Game Semantics is a powerful mathematical framework to reason compositionally on programs. In contrast to other more classical methods in denotational semantics, game semantics sees programs not merely as mathematical functions taking inputs (which may be functions themselves) and producing outputs, but rather, sees them as dynamic and interactive processes. It captures compositionally the control flow even for programs with complex computational primitives such as higher-order, mutable state, control operators, etc.

However, game semantics is not a *mathematical theory*: rather, it is a loose collection of separated technical settings, sometimes only loosely connected by the folklore of a handful of experts. One of those technical setting is *concurrent games*, inheriting from the concurrent games of Abramsky and Melliès and from Melliès' asynchronous games. In its recent technical set-up proposed by Rideau and Winskel, this model is based on *event structures*, a *true concurrency* model that avoids assuming the existence of a global clock, representing instead programs via their causal structure.

Of course, concurrent games can handle the semantics of concurrent programs. But in this monograph, I show that more than yet another new model, their expressivity reveals structures implicitly present in several other models from the literature, for concurrent languages, or not. In doing so, walking in the footsteps of Abramsky and Melliès' vision, they give us the means for a synthesis between various games or other denotational models, bringing game semantics one step closer to a mathematical theory.

Concretely, this monograph starts in Part I with a detailed introduction to several traditional *pointer* game semantics: first for the paradigmatic purely functional language PCF, its extension IA with mutable state, and the non-alternating variant for its concurrent extension $\mathsf{IA}_{/\!/}$. In Part II, I give a complete presentation of concurrent games and their extension with symmetry. In Part III, I link concurrent games to the different games models introduced in Part I – but also other models, such as the relational model – via interpretation-preserving functors. Finally, in Part IV, I survey other developments in concurrent games, and I conclude with some open problems and perspectives.

Contents

Preface

This document is my HDR – "Habilitation à Diriger les Recherches". As such, its principal aim is to motivate and present the heart of my research work so far.

Of course, I was not the only one in this adventure, this is the product of collaborations with a number of people. Firstly with Glynn Winskel: I started working with Glynn during my postdoc in Cambridge from September 2011 to September 2013, and our collaboration kept going strong long after that. My work owes a lot to him; many results presented here may not exist at all if not for his encouragements and insights.

Secondly, almost of my further work in this direction is collaborative, and was carried out with PhD students. First with Simon Castellan (2014 – 2017), with whom we constructed thin concurrent games and parallel innocence. This forms the core of the material that I have chosen to present in this monograph, though I believe the presentation has since then significantly gained in maturity. Then, with Aurore Alcolei (2016 – 2019) and Marc de Visme (2017 – 2020). Though the work done with them will not be covered therein, this document does greatly benefit from the intuitions gained with them – their work will be mentioned in the outline of further work in Chapter 13. Finally, the story currently continues with Lison Blondeau-Patissier (2021 – 2024).

No research is done in a vacuum and this work, as usual, stands on the shoulders of giants. First and foremost, the giants include the pioneers of game semantics: Martin Hyland and Luke Ong; Samson Abramsky, Radha Jagadeesan and Pasquale Malacaria; Hanno Nickau; and Thierry Coquand. Beyond those, the giants also include those who realized the strength of game semantics as a means to represent computational effects, for instance Guy McCusker, Jim Laird, Dan Ghica and Andrzej Murawski. And finally, the giants include those who kept deciphering the foundations of game semantics and maintained the sometimes thinning connection with linear logic: for instance, Pierre-Louis Curien, Laurent Regnier and Paul-André Melliès. The present work builds on their inheritance (and surely that of many others), and to an extent is a synthesis of their ideas. It is not at all obvious that such a synthesis was possible: it was first foreseen by Paul-André Melliès that this was possible, building on techniques from concurrency theory, and to a large extent this was the intent behind his line of work on *asynchronous games*. This line of work, which for simplicity (and although it can be slightly misleading) I will often refer to as *concurrent games*, inherits Paul-André's vision. Many of our technical steps parallels his, with the hindsight provided by 15 more years of work.

Beyond presenting this work, this document has two further objectives:

Firstly, I intend it as an entry point to my work on concurrent games. The technical underpinning of concurrent games is fairly elaborate: event structures are much more mathematically demanding than sequential structures (traces, trees, etc). The foundations are spread over a number of papers, written over the years. These papers rely on definitions and notations that are not quite consistent with each other, as our conceptual understanding progressed over time. In this manuscript, I present a complete and detailed construction of *Thin Concurrent Games*, a bicategory of concurrent strategies which plays a pivotal role in my further work – this is the purpose of Part II. I also present in Part III a detailed presentation of two developments at the core of the theory of Thin Concurrent Games, which I will introduce and motivate in Section 1.7.

Secondly, I needed an introduction to game semantics for programming languages, beyond merely concurrent games. In part, this is because I needed a precise set of definitions to refer to for the work presented in Part III. But beyond that, over the years I had come to dread the question "Where can I find a good introduction to game semantics"? While good introductory texts do exist, I felt that none gave a satisfactory answer. Indeed, many were written in the late 90s or earlier 00s and are not very modern or up-to-date. Others give a good survey, focusing on intuitions, but do not offer precise technical definitions; or they do offer a technical introduction, but to one particular technical setting, ignoring the wider picture. Hence, Part I is the result of my attempt at an accessible introduction to game semantics for programming languages.

Chapter 1

Introduction

Attempts to formally prove properties of the execution of programs (*e.g* functional correctness, safety, termination, *etc*) must rest on *formal semantics* turning the execution into a well-specified mathematical object. This is the main purpose of the field of *semantics of programming languages*, for which a reference textbook is [Winskel, 1993].

1.1 Operational Semantics

Typically, this is done via the methodology of **operational semantics**: by means of formal rules operating on syntax, showing how the source code is transformed throughout computation. For instance, assuming we have a value $\underline{n}$ for every $n \in \mathbb{N}$, the rules

$$\frac{e_1 \rightsquigarrow e_1'}{e_1 + e_2 \rightsquigarrow e_1' + e_2} \qquad \frac{e_2 \rightsquigarrow e_2'}{\underline{n} + e_2 \rightsquigarrow \underline{n} + e_2'} \qquad \frac{}{\underline{n} + \underline{m} \rightsquigarrow \underline{n+m}}$$

inductively define an evaluation relation $\rightsquigarrow$ between additive arithmetic expressions, specifying the operational behaviour of the sum: here, it has a left-to-right evaluation strategy: given a sum expression $e_1 + e_2$, we first evaluate e_1 to a value $\underline{n}$ – which may take many steps – then evaluate e_2 to a value $\underline{m}$. Once both operands are values, the expression $\underline{n} + \underline{m}$ is rewritten to $\underline{n+m}$. For instance, we have the reduction sequence

$$(2+3)+(7+(1+2)) \rightsquigarrow 5+(7+(1+2)) \rightsquigarrow 5+(7+3) \rightsquigarrow 5+10 \rightsquigarrow 15\,,$$

omitting the underlining. Each step above is justified by a derivation, *e.g.*

$$\frac{\dfrac{\dfrac{}{1+2 \rightsquigarrow 3}}{7+(1+2) \rightsquigarrow 7+3}}{5+(7+(1+2)) \rightsquigarrow 5+(7+3)}$$

for the second step. This would be usually called a *small-step operational semantics*; *big-step semantics* also exist, where where one axiomatises directly the evaluation relation $e \Downarrow \underline{n}$ of an expression to a value. More generally, there is a large body of literature studying variations of operational semantics; including a mathematical theory of operational semantics initiated by Turi and Plotkin [Turi and Plotkin, 1997].

Operational semantics is a powerful methodology for formal semantics of programming languages. It is very robust to the addition of syntactic constructs or programming features, and has proved capable of handling realistic programming languages. As it builds on simple inductive structures, it is amenable to mechanization in a proof assistant – as powerfully illustrated in the celebrated CompCert project by Leroy *et al* [Leroy, 2009, Krebbers et al., 2014]. However, operational semantics also has some drawbacks. By design, it is very much tied to syntax. More importantly, most frameworks are exclusively designed to reason on complete programs, operating in a closed world. They are ill-equipped to reason about *open* programs, that may include free variables or appeal to external libraries, and consequently are not compositional[1].

1.2 Toward a Mathematical Semantics

Another approach to the formal semantics of programs is that of **denotational semantics**, initiated by Scott and Strachey under the name of *mathematical semantics* [Scott and Strachey, 1971]. Following the methodology of denotational semantics, any program M is sent to a mathematical object $[\![M]\!]$, its *meaning* or **denotation**. It is usually an invariant of evaluation, in the sense that if $M \rightsquigarrow M'$, then $[\![M]\!] = [\![M']\!]$. Thus a closed program of ground type is typically sent directly to the value it returns, if any, giving no insight on the evaluation process itself (at least explicitely). In contrast with operational semantics, denotational semantics focuses on *open programs*. To those it also gives a proper meaning, often as a function, as in the example below:

$$\begin{array}{rcccc} [\![x + (x + 5)]\!] & : & \mathbb{N} & \to & \mathbb{N} \\ & & n & \mapsto & 2n + 5 \end{array}$$

In contrast with operational semantics, compositionality is at the core of the methodology of denotational semantics. A denotation is assigned to each syntactic construct, and the denotation of the program is obtained by replacing modularly each piece of syntax by its denotation, as illustrated by the computation below:

$$\begin{array}{rcl} [\![x + (x + 5)]\!](n) & = & [\![+]\!]([\![x]\!](n), [\![x + 5]\!](n)) \\ & = & [\![+]\!](n, [\![+]\!]([\![x]\!](n), [\![5]\!](n))) \\ & = & [\![+]\!](n, [\![+]\!](n, 5)) \\ & = & [\![+]\!](n, n + 5) \\ & = & 2n + 5 \end{array}$$

[1] There are, of course, formalisms for compositional operational semantics, used for instance to make CompCert compositional [Stewart et al., 2015]. By and large, those put into play structures strongly related to denotational semantics; for instance [Stewart et al., 2015] acknowledge similarity with Ghica and Tzevelekos' *system-level game semantics* [Ghica and Tzevelekos, 2012].

where $[\![+]\!] : \mathbb{N}^2 \to \mathbb{N}$ is the usual arithmetical addition operation.

The traditional core of denotational semantics is *domain theory*, in which types are represented by certain posets satisfying completeness properties useful to represent infinite data and partial computation, and open or higher-order programs are represented as (continuous) functions. But this traditional approach to denotational semantics has limitations, the focus on representing programs as functions being in tension with computational effects such as non-deterministic choice, probabilistic choice, references, exceptions, *etc* (though many of those effects can be accommodated via adequate monads [Moggi, 1991]). The modern understanding of *denotational semantics* is often taken more generally to mean a compositional interpretation of programs, usually structured categorically (typically as a cartesian closed category [Lambek and Scott, 1988]). In this wider understanding, denotational semantics may be regarded as the art of reasoning compositionally about program behaviour, which is of paramount importance for the purpose of having formal semantic techniques that scale [O'Hearn, 2015].

Over the years, a wealth of denotational models have appeared for programming languages, according to which programs are represented as other structures than functions. For instance, in the *relational model* [Girard, 1988], we have

$$\begin{array}{lcll} [\![x + (x + 5)]\!] & \subseteq & \mathscr{M}_f(\mathbb{N}) & \times \quad \mathbb{N} \\ & = & \{([n, m], & n + m + 5) \mid n, m \in \mathbb{N}\} \end{array}$$

where $\mathscr{M}_f(X)$ is the set of *finite multisets* of elements of X – in this semantics, we record that we may obtain value $n + m + 5$ provided we have *two* successful evaluations of the argument, one yielding value n, the other yielding value m. In contrast to the functional models mentioned earlier, this one is *quantitative*: we see in the interpretation that the program makes *two* calls to its argument, an information lost in domain semantics.

We also account for the behaviour of the program against a potentially non-deterministic environment: for instance, while in functional semantics the program $x\ \mathbf{xor}\ x$ will return $\mathbf{ff}$ for any value of x (writing $\mathbf{tt}$ and $\mathbf{ff}$ the two boolean values), we have

$$([\mathbf{tt}, \mathbf{ff}], \mathbf{tt}) \in [\![x\ \mathbf{xor}\ x]\!]$$

in relational semantics, accounting for an execution of $x\ \mathbf{xor}\ x$ in the context of a non-deterministic x. Note that here, $[\mathbf{tt}, \mathbf{ff}] = [\mathbf{ff}, \mathbf{tt}]$. There is no sense in which an element of the multiset may be unambiguously assigned to an occurrence of x – we do not know if the program returns $\mathbf{tt}$ because the first x returned $\mathbf{tt}$ and the second $\mathbf{ff}$, or the other way around. We can enrich the model to account for the fact that $([\mathbf{tt}, \mathbf{ff}], \mathbf{tt})$ is realized by these *two* executions, by assigning a *weight* to every point of the relational model

$$[\![x\ \mathbf{xor}\ x]\!]([\mathbf{tt}, \mathbf{ff}], \mathbf{tt}) = 2$$

and obtain in this way a *weighted relational model* [Laird et al., 2013], in which two terms of PCF are sent to the same representation iff they cannot be distinguished by PCF with probabilistic choice [Ehrhard et al., 2018]!

But the above still forgets much about the program, for instance it tells us nothing about the behaviour of the program under a stateful execution environment. For that,

we must yet again move up in the intensionality ladder, into the realm of *game semantics* [Hyland and Ong, 2000, Abramsky et al., 2000], and record the behaviour of the program over time as illustrated in the diagram below

$$
\begin{array}{ccccc}
[\![x \;\mathsf{xor}\; x]\!] & : & x : \mathbb{B} & \vdash & \mathbb{B} \\
 & & & & \mathbf{q}^- \\
 & & \mathbf{q}^+ & & \\
 & & \mathbf{tt}^- & & \\
 & & \mathbf{q}^+ & & \\
 & & \mathbf{ff}^- & & \\
 & & & & \mathbf{tt}^+
\end{array}
$$

breaking the symmetry between the two calls by explicitely showing them over time.

There are multiple ways to enrich denotational models with time, leading to many mathematical settings for game semantics: for instance, one can interpret a PCF program in Abramsky and McCusker's model [Abramsky and McCusker, 1996] which characterizes programs undistinguishable with ground state, in Berry and Curien's sequential algorithms which characterize indistinguishability in the presence of control operators [Cartwright et al., 1994], in Murawski's model for indistinguishability by state and control [Murawski, 2007], in Ghica and Murawski's model for indistinguishability by shared state and concurrency [Ghica and Murawski, 2008]... That is just the tip of the iceberg, and only listing game semantics models. Overall, a term of PCF can be interpreted into maybe a dozen different denotational models, sometimes similar and sometimes different, each recording a different aspect of its interactive behaviour[2].

In our tour of denotational models we should certainly also mention bicategorical models such as *generalized species of structure* [Fiore et al., 2008] or Melliès' *template games* [Melliès, 2019a], which assign to each point of the interpretation of a type in the relational model a *set* (or a category) of witnesses, presenting the different executions realizing that point – so rather than merely stating $[\![x \,\mathbf{xor}\, x]\!]([\mathbf{tt}, \mathbf{ff}], \mathbf{tt}) = 2$, the interpretation would yield a particular two-element set. Finally, all these developments phrased explicitely as denotational models fit into a more general landscape of tools and methods to approximate program behaviour (including idempotent or non-idempotent intersection types, resource or differential calculi, the Taylor expansion of programs, *etc*). There is an impressive breadth here: denotational semantics go way beyond the somewhat boring formalization of the "intended" denotation of a term as an official mathematical function, it is in fact the quite varied art of extracting compositional information from programs, and of reasoning about program identity.

[2]Game semantics models, along with others such as Girard's *geometry of interaction* [Girard, 1989], are often collectively referred to as *interactive semantics*. But this might be a misnomer: in reality, all the above semantics are interactive semantics in the sense that they all collect information about the interactive or compositional behaviour of the program, as opposed to the mechanics of its evaluation.

1.3 The Next 700 Denotational Semantics

It is fascinating that so many subtly different compositional accounts can be given, even if one considers only a single computational object such as PCF. What is also striking, is that there are comparatively surprisingly few results formally *relating* all these different models – even though many of those largely rely on similar intuitions, notably inherited from Linear Logic [Girard, 1987]. One explanation is that the community has focused more on building new models, taming new languages and effects, than on linking existing models. There is also a social aspect, individual researchers sticking with their favourite tools and methods. But besides those reasons, it remains true that such connections have in the past proved much more elusive than anticipated.

This is well-illustrated by the *relational collapse* of game semantics: we introduced above game semantics as relational semantics plus time, so it is only natural to attempt to send game semantics to the relational model by simply forgetting time:

$$
\begin{array}{lcll}
x : \mathbb{B} & \vdash & \mathbb{B} & \\
 & & \mathbf{q}^- & \\
\mathbf{q}^+ & & & \\
\mathbf{tt}^- & & & \\
 & & & \rightsquigarrow \quad ([\mathbf{tt}, \mathbf{ff}], \mathbf{tt}) \\
\mathbf{q}^+ & & & \\
\mathbf{ff}^- & & & \\
 & & \mathbf{tt}^+ &
\end{array}
$$

but this almost simple-minded idea hides two major conceptual hurdles: firstly, the traditional notion of plays in game semantics has time too wired in, and does not support an adequate notion of *position*. Secondly, composition of strategies also involves time; two interacting strategies may agree on a common position but not on a common *trajectory*, preventing synchronization. Understanding this prompted the work of many researchers, spanning over two decades [Baillot et al., 1997b, Melliès, 2005, Boudes, 2009, Calderon and McCusker, 2010, Clairambault and Paquet, 2021].

There are a few landmark results relating different models, such as Ehrhard's *extensional collapse* theorem [Ehrhard, 2012], which links the qualitative and quantitative views of resource replication. But by and large, a lot of seemingly simple questions turn out to hide a surprising amount of conceptual and technical depth[3].

Despite this relative rarity of actual formal connections, experts in the field do have a strong intuition as to what all these models record, and where they intuitively stand in relation to each other. After all, intuitively all those models are not independent: they all refer to, and present a part of, the interactive behaviour of the program. Over time, the working semanticist builds up an intuition of this full interactive behaviour, through the lens of their favourite technical tools. This "full interactive behaviour" is, arguably,

[3] As a more recent example, generalized species of structure are often presented – as we did above – as refining the weighted relational model, replacing mere cardinality information by an explicit of witnesses. This suggests that the coefficients computed by the weighted relational model are simply the cardinal of the set of witnesses computed by generalized species; but that is simply not true! (see Section 13.3).

the actual object of study of our field. In a scientific meeting on denotational semantics, there might be some talks about relational models, vectorial semantics, game semantics, Taylor expansion, categorical models, and others; but in essence, all these talks are about this common "full interactive behaviour". Yet the "full interactive behaviour" is not a formal object. It is intuitively there, but it seems that it can only be accessed indirectly through the various models – one cannot look at it in the eye.

But what if one could, at least to an extent? If nothing else, this would be a precious help in understanding rigorously how the models relate to each other, transporting results, *etc*. It would be a cornerstone of a unified theory of denotational semantics.

1.4 Full Abstraction and Intentional Intensionality

Of course, a model presenting the "full intensional behaviour" would be by nature extremely intensional[4], and representing the intensional behaviour of programs has not been the historical focus of denotational semantics. Instead, the driving question has long been that of *full abstraction*: the problem of capturing observational equivalence.

Let us make things a bit more precise. Informally, two programs M and N in a fixed language $\mathcal{L}$ are **observationally equivalent** if their behaviour cannot be separated within $\mathcal{L}$: there is no context $C[]$ – a program of $\mathcal{L}$ with a hole – such that, for instance, $C[M]$ converges while $C[N]$ diverges. In other words, within $\mathcal{L}$, M and N are completely interchangeable. In that case, we write $M \simeq N$. In many ways, observational equivalence appears to be the "ideal" semantic equivalence between programs; with implications in both theory and practice (for instance, one could hope that an optimization pass in a compiler replaces a program with an optimized, but observationally equivalent, variant). But observational equivalence is hard to establish, due to the infinitary universal quantification on contexts. Accordingly, the driving question in denotational semantics has long be to seek characterizations of observational equivalence: an interpretation $[\![-]\!]$ is **fully abstract** when we have $M \simeq N$ iff $[\![M]\!] = [\![N]\!]$.

In the 90s, much research on denotational semantics was driven by the *full abstraction* problem, of building a fully abstract model for the purely functional language PCF. This was not a practical question, but rather a quest to understand higher-order sequentiality. As an extensional solution remained elusive, this prompted the development of *game semantics for* PCF [Hyland and Ong, 2000, Abramsky et al., 2000, Nickau, 1994]: rather than to find conditions for functions to be sequentially computable, the idea became to find conditions for certain sequential interactive processes called *strategies* to compute *functions*; in other words, to find conditions on strategies to represent the interactive behaviour of purely functional programs. In Hyland-Ong games [Hyland and Ong, 2000], two were necessary: *innocence* and *well-bracketing*. It is debatable whether this did solve the full abstraction problem for PCF[5], but game semantics had a much better sur-

[4]Meaning, recording information about the *dynamics* of programs, which might not be observable by the execution environment – as opposed to the *extensional* behaviour, *i.e.* input/output.

[5]Nowadays, it seems that the consensus is that it *did not*: the additional intensional behaviour carried by strategies is eliminated by an undecidable quotient. On the other hand, it appears that we cannot do better: there is no effectively presentable fully abstract model for PCF, because observational equivalence in PCF is

prise in store: in the subsequent years, it quickly turned out that lifting *innocence* gave a model of PCF plus local references of ground type [Abramsky and McCusker, 1996, McCusker, 2003], lifting *well-bracketing* gave a model of PCF plus **call/cc**, while lifting both gave a model of PCF with those two effects combined [Murawski, 2007], all fully abstract *and* effectively presentable! In other words, we had *one single* semantic framework for all combinations of state and control, with each condition on strategies corresponding to (the absence of) a certain effect. This was not missed by the community, eventually acknowledged by the *2017 Alonzo Church Award for Outstanding Contributions to Logic and Computation* awarded to the authors of the three *full abstraction for* PCF papers, with the following excerpt from the announcement[6].

> "Game semantics has changed the landscape of programming language semantics by giving a unified view of the denotational universes of many different languages. This is a remarkable achievement that was not previously thought to be within reach."

To explain this success, we argue that what game semantics has stumbled upon here while seeking full abstraction for PCF, is a glimpse of this "full interactive behaviour" – *at least in a world where everything is sequential deterministic*. This made it possible to understand computational effects in terms of their induced intensional behaviour, and to present four fully abstract models, which could otherwise have appeared completely independent from each other, as four projections of this interactive behaviour.

Of course, game semantics did not stop at the sequential deterministic picture outlined above: there are also fully abstract models for non-deterministic programming languages [Harmer, 1999], probabilistic [Danos and Harmer, 2000] or concurrent languages [Laird, 2001b, Ghica and Murawski, 2008] among others; but they are defined by separate, seemingly incompatible sets of definitions, and they no longer fit in a unified framework as above. For instance the models for non-deterministic or probabilistic computation do not support a notion of *innocence* eliminating state; there is no clear path to a probabilistic extension of the game semantics of concurrent computation, *etc*.

So why do we get this appealing unified picture for sequential deterministic computation, only to lose it immediately after? Where does this come from?

1.5 Plays Considered Harmful

We argue that the limitation causing this is hidden in the very basic definition:

Definition 1.5.1. *Consider A a* game*, inducing a set of* moves M_A *and* ***plays*** $P_A \subseteq M_A^*$.
A ***strategy*** *on A is a non-empty, prefix-closed* $\sigma \subseteq P_A$*, subject to further conditions.*

This seems natural: the interpretation of a type specifies the valid executions, the *plays* of the game; and σ lists the paths that the strategy is prepared to take.

undecidable even with finite basic datatypes and without recursion [Loader, 2001].

[6] `https://eatcs.org/index.php/component/content/article/1-news/2473-the-2017-alonzo-church-award`

But this choice of representation has a cost.

Bounding Opponent's behaviour. When setting up a game semantics, one must usually start by specifying the plays. Plays form the basic backbone of a game semantics – strategies are formed out of plays, composition starts with operations on plays, *etc*.

This may not sound like much, but this choice of plays is perhaps *the* aspect that keeps the most game semantics settings apart and incompatible. Should plays be alternating? Or just locally alternating, as in AJM games? Should they be well-bracketed? Visible? Or even innocent? Some developments carry easily from one choice to another, for instance, essentially the same well-bracketed innocent strategies can be defined on many of those options. But in general, this choice has heavy consequences: it essentially specifies the computational effect available to the ambient environment. Picking a notion of plays is picking a computational universe, it is choosing the *programming language* that the *rest of the world* is written in. If a program was interpreted in a model based on well-bracketed plays, we can say nothing about its execution against **call/cc**. If a program was interpreted in a model based on alternating plays, then we can say nothing about its execution in a concurrent environment.

Should we just adopt once and for all the most general available setting, say based on non-alternating plays? That may be appealing in principle, but many key constructions in game semantics (such as the notion of *P-view*) only make sense for restricted plays.

Losing the branching structure. A play describes a *single* interactive execution, in isolation from all the other ways in which history could have unfolded. A strategy is a set of such plays, with an induced tree structure, that may *branch*. If the first moves that differ are Opponent moves, this simply corresponds to different evaluation choices by the execution environment. In contrast, if the first moves that differ are Player moves, then this witnesses a non-deterministic choice by the program. For instance, the plays

$$\begin{array}{c} \mathbb{B} \\ \mathbf{q}^- \\ \mathbf{tt}^+ \end{array} \quad , \quad \begin{array}{c} \mathbb{B} \\ \mathbf{q}^- \\ \mathbf{ff}^+ \end{array}$$

together inform a strategy $\mathbf{coin} : \mathbb{B}$ implementing non-deterministic choice.

But the point where branches of the tree of plays separate is only the point where a non-deterministic choice becomes observable, not necessarily where the choice actually happens! For instance, the two terms (for $\mathbb{U}$ a unit type with only one value)

$$x : \mathbb{U} \vdash \mathbf{if\ coin\ then}\ x;\ \mathbf{tt}\ \mathbf{else}\ x;\ \mathbf{ff} : \mathbb{B}, \qquad x : \mathbb{U} \vdash x;\ \mathbf{coin} : \mathbb{B}$$

allow exactly the same plays, which are all the prefixes of the one below:

$$\begin{array}{ccc} x : \mathbb{U} & \vdash & \mathbb{B} \\ & & \mathbf{q}^- \\ \mathbf{q}^+ & & \\ \checkmark^- & & \\ & & b^+ \end{array}$$

In reality, for the first program, at $\mathbf{q}^+$ the coin has already landed and the result is already determined, but we cannot see that. This has a number of consequences: firstly, being unable to *see* the point of non-deterministic branching means that we are stuck with an *angelic* view of non-determinism, where **if coin then** x **else** $\bot$ is equivalent to x and the possibility of divergence (with $\bot$ a looping primitive) is ignored[7].

Secondly, while this missing branching information turns out unnecessary to obtain a fully abstract model for a higher-order language with non-deterministic choice [Harmer and McCusker, 1999], this model no longer fits into a unified landscape as in the sequential deterministic case. In particular, attempting to ban state by a naive extension of *innocence* fails [Harmer, 2004]. Intuitively, *innocence* works by identifying a well-behaved fragment within the model that looks like syntax; but syntax *does* recall the branching information (as illustrated by the two terms above) while strategies as sets of plays cannot. There is no evident way to ban state in Harmer and McCusker's model [Harmer and McCusker, 1999] and obtain full abstraction for non-deterministic PCF; in this monograph we say that non-determinism and state are *entangled*.

While non-determinism is one possible branching structure, there are other "branching effects" that similarly change the very geometry of execution: two main examples are probabilistic choice and parallelism. As in the non-deterministic case, probabilistic choice and state are entangled in [Danos and Harmer, 2000], while parallelism and state are entangled in [Ghica and Murawski, 2008]; in either case there is no clear notion of innocence banning state while still allowing probabilistic choice and parallelism.

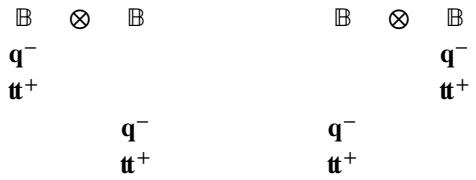
Prior to the line of work reported here, it had already appeared that adopting notions of strategies with explicit branching information, one could indeed obtain a notion of non-deterministic innocence which as desired captures a non-deterministic syntax [Castellan et al., 2014, Tsukada and Ong, 2015]. So it seems that extending the unified picture offered by game semantics beyond the sequential deterministic world is possible in principle, provided we replace plays with adequate mathematical objects giving a true account of various branching structures – we shall see later on that such a mathematical setting may be provided by *event structures* [Winskel, 1986].

Losing positions. Games are incredibly widespread beyond semantics; but on the surface their basic definitions often look rather different from Definition 1.5.1. Typically (say *e.g.* in *parity games* [Arnold and Niwiński, 2001] – or more generally, in games used in verification), a *game* is a graph of *positions*, which are split into *Player* and *Opponent* transitions. A *play* is a path in the graph starting in some initial position, and a *strategy* is a function that to any path ending in a Player position may associate a Player transition. A crucial notion is that of a *positional* strategy, whose behaviour does not depend on the whole play but only on the current position.

Curiously, this notion of *position* is missing from the very basic definitions of traditional game semantics, where instead the fundamental object is that of *plays*. While we

[7] Information about *must-convergence* can be retained by additionally recording *divergences*, as in [Harmer and McCusker, 1999] – but this does not extend to more elaborate resolutions of non-determinism, such as bisimulation or fair-testing equivalences.

may *a posteriori* observe that the two plays



(1.1)

reach the same *state*, *i.e.* the same set of moves where the two booleans have been evaluated and both have returned **tt**, the strategies used in semantics are typically not positional, and positional strategies are not stable under composition.

This is more than a peculiarity – at least if one hopes to link game semantics to other denotational models, since as hinted at in Section 1.3, positions are the key notion to establish a link with the relational model. Alternative, positional foundations for game semantics have existed for a while, such as the *graph games* of [Hyland and Schalk, 2002] and the *asynchronous games* of [Melliès, 2004a]. But by and large, they have not been developed beyond models of fragments of linear logic.

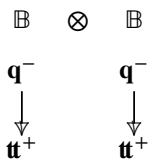
1.6 A Case for Causal Game Semantics

But if we are to remove plays, by what should they be replaced?

At first sight, *removing plays from game semantics* sounds a bit like removing malt from brewing, or removing wool from knitting. Of course committing to a notion of plays will constrain what you may do with them, but you have to live with that, after all plays *are* the basic material with which the game semanticist builds.

However, there is an alternative: *causal game semantics*. The developments presented in this monograph result from a paradigm shift: rather by presenting computational events in sequence, as they could be observed by a specific environment – however powerful it may be, we propose to organize them *causally*: one event appears before another only if this is imposed by a causal dependency, typically internal to the program.

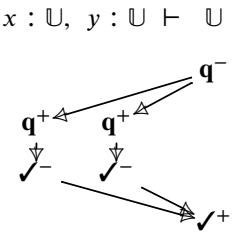
Recovering positions. For instance, the pair (**tt**, **tt**) would be represented by:



where below the first line that presents the two type components, we show the Hasse diagram of a *partial order*, read from top to bottom. In each type component, the program is prepared to return **tt**, under the unique cause that Opponent starts computation by playing $\mathbf{q}^-$. The comparison with the two diagrams of (1.1) is clear: we have removed *time* and the induced necessity of a choice for Opponent's scheduling. The diagram

only describes what actions the program is prepared to take, and under which conditions. The *positions* are easily accessible as certain down-closed (so *up-closed* in the diagram) sets of observable events; and we should see in the course of the monograph that this allows for a crisp link with relational semantics.

Recovering the branching structure(s). Adopting such partial orders allows us to record explicitly *parallel* branching information. The following diagram represents a strategy evaluating its free variables of unit type $\mathbb{U}$ (admitting only one value) in parallel

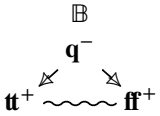


where the branching on the second line with the two $\mathbf{q}^+$ is *not* a non-deterministic choice (although it might lead to a non-deterministic choice if that parallelism is resolved by a scheduler), but a *parallel* branching: the events corresponding to the two threads for x and y are, as far as the intention of the program is concerned, *causally independent*.

A warning: this diagram does *not* mean that the program *enforces* the independence of x and y, it does not mean that the evaluation of x and y *will* be independent once the context for x and y is known. It only means that the program does not itself impose causal constraints between the two threads. It is possible that the calls to x and y trigger the access to a shared resource, making them interfere. But this interference is then due to the interaction of the program with its environment, not to the program itself.

Much of concurrency theory is based on *interleavings*: an execution of two processes in parallel $P \parallel Q$ is formalized by considering all interleavings of executions of P and Q, hence reducing concurrency to non-determinism by integrating the non-determinism from the scheduler. According to that interpretation, the program above looks non-deterministic as either x or y may be called first in a sequential execution. In contrast, for us (following the so-called *truly concurrent* approach to concurrency theory), this program is considered deterministic: all schedulings eventually reach the same state.

But if that is so, and if branching is parallelism, how can we express actual non-deterministic choice? We do so by adding a *conflict* relation, shown as a wiggly line



indicating that the two moves are *incompatible*, and cannot both occur in a single execution. Altogether, the partial order and the conflict relation form an *event structure* [Winskel, 1986], a well-established notion in concurrency theory.

Lifting constraints on Opponent. Those causal structures, which we call *concurrent strategies*, are developed in Part II. By design, concurrent strategies do not put *any* constraint on Opponent's behaviour: unlike all play-based game semantics, concurrent strategies do not record how a program is observed by a fixed environment, but only what observable actions the program is prepared to take, and under which conditions.

The purpose of concurrent games is to record very intensional information on the interactive behaviour of programs; information that is typically not observable, however strong the environment may be. In that, it is at odds with much of the earlier work on game semantics for programming languages, which focuses on obtaining full abstraction results and as such, on capturing the information about program behaviour that is observable by specific programming features. In adopting concurrent games, we exchange that for a crisp description of the *full interactive behaviour* of the programs from which all other denotational interpretations can, at least in principle, be obtained as certain projections – a claim that we shall aim to substantiate in this monograph.

Despite our overarching point about the necessity of capturing the full interactive behaviour, some will surely question the point of a semantics not designed to reason on observational equivalence, and recording interactive behaviour that is unobservable, even by a wildly powerful environment. To those, we oppose the following question: is it really reasonable to consider that a program may only interact with an environment written in a *single* programming language, with a fixed observational power? This "fixed world hypothesis", followed in much of denotational semantics, is in tension with the heterogeneous and distributed nature of modern software. Like any denotational semantics, game semantics is by nature modular with respect to program construction. But the fixed world hypothesis makes it *less* modular in another direction, that of programming language extensions.

One also could wonder if history will remember the full abstraction results as the most important contribution of game semantics. Arguably, there are already signs that probably not. In the recent past, a few developments have been proposed by groups without prior knowledge and experience in game semantics, but leveraging ideas or techniques for game semantics [Stewart et al., 2015, Xia et al., 2020, Chappe et al., 2023, Koenig and Shao, 2020, Vale et al., 2023]. By and large they are motivated by the need for more compositional structures in program certification – they need game semantics for its compositional description of interactive execution, *not* its full abstraction results!

Thus it seems healthy to part with the "fixed world hypothesis" – but it would be better if that did not mean throwing away decades of work and developments in denotational semantics, and the myriad of results and techniques developed. This is, we claim, what concurrent games may eventually achieve, and this monograph contributes some steps.

1.7 Contributions and Outline

In our "concurrent games" line of work, quite a few developments fit into that general line; we survey a few of those at the end of this monograph, in Chapter 13. To form the heart of this document, we had to select a precious few contributions:

Concurrent games. In the published literature, the technical construction of our "concurrent games" framework spans quite a few papers. We first build on Rideau and Winskel's original set of definitions for a bicategory where both games and strategies are event structures [Rideau and Winskel, 2011]. Our work is based on an extension of this bicategory with symmetry, a framework called *thin concurrent games*, that first appeared (without details and proofs) in a conference paper [Castellan et al., 2015] and then detailed in [Castellan et al., 2019], though depending on an earlier reconstruction of the basic bicategory of concurrent games in [Castellan et al., 2017a]. This makes the chain of dependencies of specific results and lemmas a bit challenging to track, especially given that the notations and terminology have matured over time.

Thus the first contribution of this monograph is a complete, self-contained and hopefully pedagogical presentation of thin concurrent games. It comprises a new presentation of the basic bicategory **CG** without symmetry, the central bicategory **TCG** of *thin concurrent games*; and the description of a number of additional constructions useful to define the define and study the interpretation of programming languages.

Disentangling parallelism and state. We use the above to give a causal version of one of the most expressive games models out there: Ghica and Murawski's nonalternating games, fully abstract for *Idealized Concurrent Algol* – a concurrent higher-order language with (local) ground state references [Ghica and Murawski, 2008].

In fact, *ICA* may be regarded as PCF enriched with two programming features: *parallelism* adds the ability to perform computations in parallel, a priori with no interaction or side-channel communication; and *state* is given by ground state local references (and semaphores). We regard *concurrency* not as a primitive computing feature, but as a phenomenon emerging from the combination of parallelism and state. Hence rather than *ICA*, we call the language *Idealized Parallel Algol* ($\mathsf{IA}_{/\!/}$), *i.e.* Idealized Algol plus parallelism – but it is indeed a concurrent language.

Resting on **TCG** we construct $\multimap\!\triangleright$-**Strat**, a cartesian closed category in which we describe an interpretation of $\mathsf{IA}_{/\!/}$. On strategies in $\multimap\!\triangleright$-**Strat**, we introduce two conditions: *sequentiality*, and *globularity*, which induce two sub-cartesian closed categories $\multimap\!\triangleright$-**Glob** (supporting the interpretation of *parallelism*, but not *state*) and $\multimap\!\triangleright$-**Seq** (supporting the interpretation of *state*, but not *parallelism*), and a third sub-cartesian closed category $\multimap\!\triangleright$-**SeqGlob** (supporting the interpretation of PCF only) where the two conditions are required. We construct **interpretation-preserving functors**:

$\multimap\!\triangleright$-**Strat**	$\to$	$\circlearrowleft$-**Strat**	(Theorem 9.3.8)
$\multimap\!\triangleright$-**Seq**	$\to$	$\downarrow\!\uparrow$-**Strat**	(Theorem 11.2.18)
$\multimap\!\triangleright$-**SeqGlob**	$\to$	$\downarrow\!\uparrow$-**InnWB**	(Theorem 11.3.10)

where $\circlearrowleft$-**Strat**, $\downarrow\!\uparrow$-**Strat** and $\downarrow\!\uparrow$-**InnWB** are the reference fully abstract models for $\mathsf{IA}_{/\!/}$, IA and PCF respectively, phrased in terms of traditional game semantics – so that $\multimap\!\triangleright$-**Strat**, $\multimap\!\triangleright$-**Seq** and $\multimap\!\triangleright$-**SeqGlob** are also fully abstract. Finally, we prove a new **finite definability result** for $\multimap\!\triangleright$-**Glob** with respect to PCF plus *parallelism*.

Altogether, these results **disentangle** parallelism and state, in the sense above.

Relational collapse. Setting up the tools required for this is quite the journey. A particularly significant step encountered along the way is that *visibility*, one of the constituents of *globularity*, ensures that the composition of strategies induces no deadlocks so that we have an **interpretation-preserving functor to the relational model** (Corollary 10.4.15) – a modern account of the link between game semantics with the relational model, in line with our motto of bridging denotational models.

A uniform presentation of pointer games. Of course, a challenge in constructing the interpretation-preserving functors above to traditional games models, is that they refer to models constructed in a number of different papers, working with different notations, definitions that differ in subtle but significant ways…To achieve our goals, it was necessary to first recast the various games models of interest, which we attempted to do in an introductive and pedagogical way.

Outline. In the next chapter, we introduce a few mathematical conventions and notations used throughout the monograph.

Part I contains our introduction to pointer games. It focuses on alternating pointer games, covering innocence, well-bracketing, the full abstraction results for PCF and Idealized Algol. Then, it also presents non-alternating pointer games and the fully abstract model for $\mathsf{IA}_{/\!/}$. We mention a few other topics along the way: *e.g.* the *semantic cube*, and the alternative approach to uniformity via *copy indices*.

Part II contains our detailed construction of concurrent games. We start with the basic bicategory **CG**, then add symmetry to obtain *thin concurrent games* (**TCG**). Finally, we construct additional structure aiming for (essentially) a cartesian closed category, as required by the interpretation of (call-by-name) programming languages.

Part III contains all the developments required to disentangle parallelism and state. It contains the interpretation of $\mathsf{IA}_{/\!/}$, the developments of *sequentiality* and *globularity* along with all the proofs of stability under composition. It contains all our interpretation-preserving functors, including that to the relational model. Finally, it contains a proof of finite definability for globular strategies.

Part IV surveys other developments in concurrent games fitting in our general pattern of bridging denotational models; as well as perspectives and open problems.

Finally, Part V contains some appendices.

Chapter 2

General Preliminaries

In this first technical chapter, we introduce some *preliminaries* to the *preliminaries*: first, we fix some of the general notations and conventions used in this monograph. Then, we introduce our programming languages of interest. Finally, we will recall definitions for a few of the basic denotational semantics concepts used later on.

2.1 Mathematical Language

To start off, let us introduce a few notations that we will use throughout this monograph.

Sets. If X is a set, then we write $\mathcal{P}(X)$ for the **powerset** of X, *i.e.* the set comprising all subsets of X. If X and Y are sets, we write $X + Y$ for the **tagged disjoint union** of X and Y, defined as $X + Y = \{(1, x) \mid x \in X\} \cup \{(2, y) \mid y \in Y\}$. We are careful to distinguish this with $X \uplus Y$ which is the usual set-theoretic union, when it is known or assumed to be disjoint – for instance, we may write $X + Y = \{1\} \times X \uplus \{2\} \times Y$.

Sequences. If X is a set, then X^* denotes the set of **words** or **finite sequences** of elements of X. We write $\varepsilon \in X^*$ for the empty sequence, and $\sqsubseteq$ denotes the **prefix ordering** on finite sequences.

Functions. If X and Y are sets, we write $f : X \to Y$ to mean that f is a function from X to Y (if X and Y are understood from the context to be objects in a specific category $\mathcal{C}$, this might mean that f is a morphism of $\mathcal{C}$ from X to Y instead). Likewise, we write $f : X \rightharpoonup Y$ to mean that f is a partial function from X to Y. In both cases, we write the application of f to $x \in X$ as $f(x)$ or sometimes simply as fx. If $f : X \to Y$ is a bijection, we write $f : X \simeq Y$.

Categories. We assume familiarity with the basic notions of category theory and categorical logic. We write **Set** for the category of sets and functions. If $\mathcal{C}$ is a category, we

write $\mathcal{C}_0$ for its set of objects; if A and B are objects of $\mathcal{C}$, we write $\mathcal{C}(A, B)$ for the set of morphisms from A to B. If $f \in \mathcal{C}(A, B)$ is an isomorphism, we write $f : A \cong B$. For an isomorphism in **Set**, we tend to prefer the term *bijection* and the notation $f : A \simeq B$, but use *isomorphism* and write $f : A \cong B$ in the presence of further structure – for instance if A and B are partial orders, and f is an order-isomorphism.

We write $\top$ for the terminal object of $\mathcal{C}$, if it exists. If $\mathcal{C}$ is a cartesian category, then we write $A\&B$ for the cartesian product of A and B. We write $\pi_A : A\&B \to A$ and $\pi_B : A\&B \to B$ for the two projections, and $\langle f, g \rangle : X \to A\&B$ for the pairing of $f : X \to A$ and $g : X \to B$. We will also use n-ary versions of these constructions.

If $\mathcal{C}$ is cartesian closed, we write $A \Rightarrow B$ for the arrow object of A and B and $\mathsf{ev}_{A,B} : (A \Rightarrow A)\&A \to B$ for the evaluation morphism. If $h : X \times A \to B$, we write $\Lambda(h) : X \to A \Rightarrow B$ for its currying.

Common abbreviations. We shall use the following abbreviations in the text: *iff* for "if and only if", *lhs* for "left hand side", *rhs* for "right hand side", *s.t.* for "such that" and *w.r.t.* for "with respect to".

2.2 $\mathsf{IA}_{/\!/}$ and its Fragments

The contributions of this monograph are not about a programming language in particular: we view them as methodological contributions in denotational semantics, that may apply, in principle, to a variety of languages – the past body of work on game semantics has indeed proved that the insights gained when studying programming features in a specific setting could, to a large extent, be transported to other settings (*e.g.* the notion of *innocence*, originally intended to ban state in a call-by-name language, turned out to apply just as well to ban state in a call-by-value language [Honda and Yoshida, 1999].

But we do need to fix target programming languages to illustrate our game semantics constributions, and to serve as target for our technical developments. For that, we settled on *Idealized Parallel Algol* ($\mathsf{IA}_{/\!/}$), a call-by-name concurrent higher-order programming language with ground local references. The main reason we adopted $\mathsf{IA}_{/\!/}$ is because among the languages that are well-studied in game semantics, this is one of the most expressive. Removing parallelism and/or state allows one to recover languages that are *also* very well-studied, though the corresponding models do not fit in a uniform semantic picture. If we want to prove full abstraction for the different fragments of $\mathsf{IA}_{/\!/}$, it "suffices" to relate to the existing fully abstract models, saving us some work.

The core of $\mathsf{IA}_{/\!/}$ is PCF, the prototypal higher-order sequential language introduced by Plotkin [Plotkin, 1977]: a call-by-name simply-typed λ-calculus with booleans, natural numbers and their combinators, plus recursion. We consider two extensions of PCF: firstly, *Idealized Algol* (IA) is PCF extended with *shared state*, under the form of (mainly) ground type local references. Secondly, *Parallel* PCF ($\mathsf{PCF}_{/\!/}$) extends PCF with a primitive for pure parallel evaluation, without communication. $\mathsf{IA}_{/\!/}$, obtained by putting all these together, is rather complex: in particular it is a *concurrent* language, as the combination of parallelism and shared memory induce *racy behaviour*.

In the rest of this section, we introduce formally the syntax of $\mathtt{IA}_{/\!/}$ and its fragments. A disclaimer: $\mathtt{IA}_{/\!/}$ and its fragments are not claimed to be realistic or in any way faithful to actual programming languages; nor do we think that realistic programming languages should be inspired by $\mathtt{IA}_{/\!/}$. It is a toy language, whose only purpose is to put together certain programming aspects in a controlled environment where they can be studied in isolation; like a physicist studying properties of matter in a vacuum. Likewise, the reader should not see this work as a "study of the semantics of $\mathtt{IA}_{/\!/}$" or its fragments. Again, our contribution is semantic and methodological, $\mathtt{IA}_{/\!/}$ being merely a way to illustrate it and provide a target that is challenging but achievable.

2.2.1 Syntax of $\mathtt{IA}_{/\!/}$

Types. The **types** of $\mathtt{IA}_{/\!/}$ are the following, highlighting types relative to state.

$$\begin{array}{rcll} A, B & ::= & \mathbb{U} \mid \mathbb{B} \mid \mathbb{N} \mid A \to B & \mathsf{PCF} \\ & & \mid \mathbb{V} \mid \mathbb{S} & +\text{state} \end{array}$$

Above, $\mathbb{U}$ is a *unit* type with only one value, and $\mathbb{B}$ and $\mathbb{N}$ are types for *booleans* and *natural numbers*. In the presence of state, $\mathbb{V}$ is a type for *references* storing natural numbers, while $\mathbb{S}$ is for *semaphores*. We refer to $\mathbb{U}, \mathbb{B}$ and $\mathbb{N}$ as *ground types*, and use $\mathbb{X}, \mathbb{Y}$ to range over those. Let us now give the term constructions and typing rules.

Terms and typing. We define the terms of the language directly via typing rules.

Contexts are lists $x_1 : A_1, \ldots, x_n : A_n$. **Typing judgments** have form $\Gamma \vdash M : A$ with Γ a context and A a type. In addition to Figure 2.1, we consider present an exchange rule letting us permute the order of variable declarations in contexts. The eliminator rules for basic datatypes are restricted to eliminate only to ground types – general eliminators are defined as syntactic sugar: *e.g.* a conditional to $\mathbb{V}$ may be obtained as

$$\frac{\Gamma \vdash M : \mathbb{B} \qquad \Gamma \vdash N_1 : \mathbb{V} \qquad \Gamma \vdash N_2 : \mathbb{V}}{\Gamma \vdash \mathbf{mkvar}\,(\lambda x.\,\mathbf{if}\ M\ (N_1 := x)\,(N_2 := x))\,(\mathbf{if}\ M\ !N_1\ !N_2) : \mathbb{V}}$$

The **bad variable** and **bad semaphore** constructs **mkvar** and **mksem** are a common occurrence in the game semantical literature. While a "*good*" reference is tied to a memory location, many game models also comprise so-called "*bad variables*" inhabiting $\mathbb{V}$ but not behaving as actual variables. Full abstraction results in the concerned games models [Abramsky and McCusker, 1996, Ghica and Murawski, 2008] require a corresponding syntactic construct **mkvar** allowing one to *form* bad variables by appending arbitrary read and write methods[1]. The same holds for semaphores.

[1] A very nice result by McCusker is that in fact, the fully abstract games model for IA is also fully abstract without bad variables [McCusker, 2003]!

PCF

$$\frac{}{\Gamma \vdash \mathbf{skip} : \mathbb{U}} \qquad \frac{}{\Gamma \vdash \mathbf{tt} : \mathbb{B}} \qquad \frac{}{\Gamma \vdash \mathbf{ff} : \mathbb{B}} \qquad \frac{}{\Gamma \vdash n : \mathbb{N}} \qquad \frac{}{\Gamma, x : A \vdash x : A}$$

$$\frac{\Gamma, x : A \vdash M : B}{\Gamma \vdash \lambda x^A.\, M : A \to B} \qquad \frac{\Gamma \vdash M : A \to B \qquad \Gamma \vdash N : A}{\Gamma \vdash M\, N : B}$$

$$\frac{\Gamma \vdash M : \mathbb{U} \qquad \Gamma \vdash N : \mathbb{X}}{\Gamma \vdash M;\, N : \mathbb{X}} \qquad \frac{\Gamma \vdash M : \mathbb{B} \qquad \Gamma \vdash N_1 : \mathbb{X} \qquad \Gamma \vdash N_2 : \mathbb{X}}{\Gamma \vdash \mathbf{if}\ M\ N_1\ N_2 : \mathbb{X}}$$

$$\frac{\Gamma \vdash M : \mathbb{N}}{\Gamma \vdash \mathbf{succ}\, M : \mathbb{N}} \qquad \frac{\Gamma \vdash M : \mathbb{N}}{\Gamma \vdash \mathbf{pred}\, M : \mathbb{N}} \qquad \frac{\Gamma \vdash M : \mathbb{N}}{\Gamma \vdash \mathbf{iszero}\, M : \mathbb{B}}$$

$$\frac{\Gamma, x : \mathbb{X} \vdash M : \mathbb{Y} \qquad \Gamma \vdash N : \mathbb{X}}{\Gamma \vdash \mathbf{let}\ x = N\ \mathbf{in}\ M : \mathbb{Y}} \qquad \frac{\Gamma \vdash M : A \to A}{\Gamma \vdash \mathcal{Y}\, M : A}$$

+state

$$\frac{\Gamma, x : \mathbb{V} \vdash M : \mathbb{X}}{\Gamma \vdash \mathbf{newref}\ x{:=}n\ \mathbf{in}\ M : \mathbb{X}} \qquad \frac{\Gamma \vdash M : \mathbb{V} \qquad \Gamma \vdash N : \mathbb{N}}{\Gamma \vdash M{:=}N : \mathbb{U}} \qquad \frac{\Gamma \vdash M : \mathbb{V}}{\Gamma \vdash !M : \mathbb{N}}$$

$$\frac{\Gamma, x : \mathbb{S} \vdash M : \mathbb{X}}{\Gamma \vdash \mathbf{newsem}\ x{:=}n\ \mathbf{in}\ M : \mathbb{X}} \qquad \frac{\Gamma \vdash M : \mathbb{S}}{\Gamma \vdash \mathbf{grab}\, M : \mathbb{U}} \qquad \frac{\Gamma \vdash N : \mathbb{S}}{\Gamma \vdash \mathbf{release}\, N : \mathbb{U}}$$

$$\frac{\Gamma \vdash M : \mathbb{N} \to \mathbb{U} \qquad \Gamma \vdash N : \mathbb{N}}{\Gamma \vdash \mathbf{mkvar}\, M\, N : \mathbb{V}} \qquad \frac{\Gamma \vdash M : \mathbb{U} \qquad \Gamma \vdash N : \mathbb{U}}{\Gamma \vdash \mathbf{mksem}\, M\, N : \mathbb{S}}$$

+parallelism

$$\frac{\Gamma, x_1 : \mathbb{X}, x_2 : \mathbb{X} \vdash M : \mathbb{Y} \qquad \Gamma \vdash N_1 : \mathbb{X} \qquad \Gamma \vdash N_2 : \mathbb{X}}{\Gamma \vdash \mathbf{let} \begin{pmatrix} x_1 & = & N_1 \\ x_2 & = & N_2 \end{pmatrix} \mathbf{in}\ M : \mathbb{Y}}$$

Figure 2.1: Typing rules for $\mathtt{IA}_{/\!/}$

2.2.2 Operational Semantics of $\mathtt{IA}_{/\!/}$

In this monograph, we shall not refer explicitly to the operational semantics for $\mathtt{IA}_{/\!/}$, as we will only interact with those through already established denotational models. However, for the sake of completeness, we include here the definition.

We refer to constants of ground type as **values**; we use v to range over those, and n, b or c to range over values of respective types $\mathbb{N}, \mathbb{B}$ or $\mathbb{U}$. We give a small-step operational semantics, following [Ghica and Murawski, 2008]. We fix a countable set $\mathcal{L}$ of **memory locations**. A **store** is a partial map $s : \mathcal{L} \rightharpoonup \mathbb{N}$ with finite domain where $\mathbb{N}$ stands, overloading notations, for natural numbers. A **configuration** is a term together with a store, formally a pair $\langle M, s\rangle$ where s is a store with $\mathsf{dom}(s) = \{\ell_1, \ldots, \ell_n\}$ and $\Sigma \vdash M : A$ with $\Sigma = \ell_1 : \mathbb{V}, \ldots, \ell_i : \mathbb{V}, \ell_{i+1} : \mathbb{S}, \ldots, \ell_n : \mathbb{S}$.

Execution is formalized via reductions between configurations, with shape

$$\langle M, s\rangle \rightsquigarrow \langle M', s'\rangle$$

where $\mathsf{dom}(s) = \mathsf{dom}(s')$; we write $\rightsquigarrow^*$ for the reflexive transitive closure. If $\vdash M : \mathbb{X}$, we write $M \Downarrow v$ if $\langle M, \emptyset\rangle \rightsquigarrow^* \langle v, \emptyset\rangle$ for some value v or simply $M \Downarrow$ leaving v implicit. We give in Figure 2.2 the reduction rules; separated in *basic rules* listing the basic cases, and the *context rules* specifying the next available redexes. For rules not interacting with the state, we omit the state component – it is simply left unchanged by stateless basic reductions, and propagated upwards by stateless context rules.

2.2.3 Expressiveness of $\mathtt{IA}_{/\!/}$

We shall illustrate the expressiveness of the language via its fragments.

The core language: PCF. Introduced in [Plotkin, 1977], PCF is the paradigmatic (call-by-name) language for higher-order recursive computation.

PCF is of course Turing-complete, and may be used to program any (higher-order) computable function. For instance, the Ackermann function may be defined with

$$\mathcal{Y}\,(\lambda A^{\mathbb{N}\to\mathbb{N}\to\mathbb{N}} n^{\mathbb{N}} m^{\mathbb{N}}.\ \mathbf{if}\ (\mathbf{iszero}\ n)\ (\mathbf{succ}\ m)\ (\mathbf{iter}\ (A\ (\mathbf{pred}\ n))\ m))$$

with $\mathbf{iter} = \lambda f^{\mathbb{N}\to\mathbb{N}} x^{\mathbb{N}}.\ \mathcal{Y}\,(\lambda I^{\mathbb{N}\to\mathbb{N}} n^{\mathbb{N}}.\ \mathbf{if}\ (\mathbf{iszero}\ n)\ x\ (f\ (I\ (\mathbf{pred}\ n))))$; showing how we can use recursion on higher-order functions in PCF. We define additional combinators in PCF that will eventually be used in the monograph. First, there is a **divergence**

$$\frac{}{\Gamma \vdash \bot_A : A}$$

for any type A, defined as any looping program, for instance via $\bot_A = \mathcal{Y}_A\,(\lambda x^A.\ x)$. We also add syntactic sugar for an **equality test** for all ground types, with rule

$$\frac{\Gamma \vdash M : \mathbb{X} \qquad \Gamma \vdash N : \mathbb{X}}{\Gamma \vdash M =_{\mathbb{X}} N : \mathbb{B}}$$

Basic red. for PCF

$$
\begin{array}{rcl}
(\lambda x^A.\, M)\, N & \rightsquigarrow & M[N/x] \\
\mathbf{skip};\ N & \rightsquigarrow & N \\
\mathbf{if\ tt}\ M\ N & \rightsquigarrow & M \\
\mathbf{if\ ff}\ M\ N & \rightsquigarrow & N \\
\mathbf{succ}\, n & \rightsquigarrow & n+1 \\
\mathsf{pred}\, 0 & \rightsquigarrow & 0 \\
\mathsf{pred}\,(n+1) & \rightsquigarrow & n \\
\mathbf{iszero}\, 0 & \rightsquigarrow & \mathbf{tt} \\
\mathbf{iszero}\,(n+1) & \rightsquigarrow & \mathbf{ff} \\
\mathcal{Y}\, M & \rightsquigarrow & M\,(\mathcal{Y}\, M) \\
\mathbf{let}\ x = v\ \mathbf{in}\ M & \rightsquigarrow & M[v/x]
\end{array}
$$

Basic reductions for state

$$
\begin{array}{rcl}
\mathbf{newref}\ x\ \mathbf{in}\ v & \rightsquigarrow & v \\
\mathbf{newsem}\ x\ \mathbf{in}\ v & \rightsquigarrow & v \\
(\mathbf{mkvar}\ M\ N) := n & \rightsquigarrow & M\, n \\
!(\mathbf{mkvar}\ M\ N) & \rightsquigarrow & N \\
\mathbf{grab}(\mathbf{mksem}\ M\ N) & \rightsquigarrow & M \\
\mathbf{release}(\mathbf{mksem}\ M\ N) & \rightsquigarrow & N
\end{array}
$$

Interfering reductions

$$
\begin{array}{rcll}
\langle !\ell, s \uplus \{\ell \mapsto n\}\rangle & \rightsquigarrow & \langle n, s \uplus \{\ell \mapsto n\}\rangle & \\
\langle \ell := n, s \uplus \{\ell \mapsto \_\}\rangle & \rightsquigarrow & \langle \mathbf{skip}, s \uplus \{\ell \mapsto n\}\rangle & \\
\langle \mathbf{grab}(\ell), s \uplus \{\ell \mapsto 0\}\rangle & \rightsquigarrow & \langle \mathbf{skip}, s \uplus \{\ell \mapsto 1\}\rangle & \\
\langle \mathbf{release}(\ell), s \uplus \{\ell \mapsto n\}\rangle & \rightsquigarrow & \langle \mathbf{skip}, s \uplus \{\ell \mapsto 0\}\rangle & (n > 0)
\end{array}
$$

Basic reduction for parallelism

$$
\mathbf{let} \begin{pmatrix} x_1 & = & v_1 \\ x_2 & = & v_2 \end{pmatrix} \mathbf{in}\ M \quad \rightsquigarrow \quad M[v_1/x_1, v_2/x_2]
$$

Stateless context rules

$$
\frac{M \rightsquigarrow M'}{M\ N \rightsquigarrow M'\ N} \qquad \frac{M \rightsquigarrow M'}{\mathbf{if}\ M\ N_1\ N_2 \rightsquigarrow \mathbf{if}\ M'\ N_1\ N_2} \qquad \frac{M \rightsquigarrow M'}{\mathbf{succ}\ M \rightsquigarrow \mathbf{succ}\ M'} \qquad \frac{M \rightsquigarrow M'}{!M \rightsquigarrow !M'}
$$

$$
\frac{M \rightsquigarrow M'}{\mathbf{iszero}\ M \rightsquigarrow \mathbf{iszero}\ M'} \qquad \frac{N \rightsquigarrow N'}{M := N \rightsquigarrow M := N'} \qquad \frac{M \rightsquigarrow M'}{\mathbf{grab}(M) \rightsquigarrow \mathbf{grab}(M')}
$$

$$
\frac{M \rightsquigarrow M'}{\mathbf{release}(M) \rightsquigarrow \mathbf{release}(M')} \qquad \frac{M \rightsquigarrow M'}{M := v \rightsquigarrow M' := v} \qquad \frac{N \rightsquigarrow N'}{\mathbf{let}\ x = N\ \mathbf{in}\ M \rightsquigarrow \mathbf{let}\ x = N'\ \mathbf{in}\ M}
$$

$$
\frac{N_1 \rightsquigarrow N_1'}{\mathbf{let} \begin{pmatrix} x_1 & = & N_1 \\ x_2 & = & N_2 \end{pmatrix} \mathbf{in}\ M \rightsquigarrow \mathbf{let} \begin{pmatrix} x_1 & = & N_1' \\ x_2 & = & N_2 \end{pmatrix} \mathbf{in}\ M}
$$

$$
\frac{N_2 \rightsquigarrow N_2'}{\mathbf{let} \begin{pmatrix} x_1 & = & N_1 \\ x_2 & = & N_2 \end{pmatrix} \mathbf{in}\ M \rightsquigarrow \mathbf{let} \begin{pmatrix} x_1 & = & N_1 \\ x_2 & = & N_2' \end{pmatrix} \mathbf{in}\ M}
$$

Stateful context rules

$$
\frac{\langle M[\ell/x], s \uplus \{\ell \mapsto n\}\rangle \rightsquigarrow \langle M'[\ell/x], s' \uplus \{\ell \mapsto n'\}\rangle}{\langle \mathbf{newref}\ x := n\ \mathbf{in}\ M, s\rangle \rightsquigarrow \langle \mathbf{newref}\ x := n'\ \mathbf{in}\ M', s'\rangle}\ (\ell \in \mathcal{L}\ \text{fresh})
$$

$$
\frac{\langle M[\ell/x], s \uplus \{\ell \mapsto n\}\rangle \rightsquigarrow \langle M'[\ell/x], s' \uplus \{\ell \mapsto n'\}\rangle}{\langle \mathbf{newsem}\ x := n\ \mathbf{in}\ M, s\rangle \rightsquigarrow \langle \mathbf{newsem}\ x := n'\ \mathbf{in}\ M', s'\rangle}\ (\ell \in \mathcal{L}\ \text{fresh})
$$

Figure 2.2: Operational semantics of $\mathsf{IA}_{/\!/}$

returning **tt** iff the two terms yield the same value. For $\mathbb{X} = \mathbb{U}$ we may define $M =_{\mathbb{U}} N$ simply as $M;\ N;\ \mathbf{tt}$. Likewise, we set $M =_{\mathbb{B}} N$ as **if** M N (**if** N **ff tt**), and $\Gamma \vdash M =_{\mathbb{N}} N : \mathbb{B}$ similarly, with the obvious recursive program.

Next we introduce a n-ary case construct branching on all values of ground types. If $V = \{v_1, \ldots, v_n\}$ is a finite set of values of ground type $\mathbb{X}$, we set

$$\begin{array}{l} \mathbf{case}\ M\ \mathbf{of} \\ \quad v_1 \mapsto N_1 \\ \quad v_2 \mapsto N_2 \\ \quad \cdots \\ \quad v_n \mapsto N_n \end{array} \quad \stackrel{\text{def}}{=} \quad \begin{array}{l} \mathbf{let}\ x\ =\ M\ \mathbf{in} \\ \qquad \mathbf{if}\ x =_{\mathbb{X}} v_1\ \mathbf{then}\ N_1 \\ \quad \mathbf{else\,if}\ x =_{\mathbb{X}} v_2\ \mathbf{then}\ N_2 \\ \qquad \cdots \\ \quad \mathbf{else\,if}\ x =_{\mathbb{X}} v_n\ \mathbf{then}\ N_n \\ \quad \mathbf{else}\ \bot \end{array}$$

of type $\mathbb{Y}$ in context Γ if $\Gamma \vdash M : \mathbb{X}$ and $\Gamma \vdash N_i : \mathbb{Y}$ for all $1 \leq i \leq n$.

This makes crucial use of the **let** construct, ensuring that M is evaluated only once. Note that the historical presentation of PCF has no **let** construct; however including it makes for a more complete language of study with some control over the evaluation order, and also allows for a neater correspondence with the game semantics.

Idealized Algol. *Idealized Algol*, or IA for short, corresponds in Figure 2.1 to PCF + state. It is used in the game semantics community (starting with the seminal paper [Abramsky and McCusker, 1996]) as the (call-by-name) paradigmatic programming language with ground type references: one can store an integer (though nothing much would change if one could store other finite datatypes), but one cannot store a function, or a reference. IA subsumes PCF, so all the programs and syntactic sugar mentioned above may still be defined here. But IA also supports imperative programming, *e.g.*

$$\begin{array}{l} \lambda n^{\mathbb{N}}.\ \mathbf{newref}\ r\ \mathbf{in} \\ \quad \mathbf{newref}\ i\ \mathbf{in} \\ \qquad r := 1; \\ \qquad i := n; \\ \qquad \mathbf{while}\ (\mathbf{not}\ (\mathbf{iszero}\ !i))\ \mathbf{do} \\ \qquad\quad r := (!i) * (!r); \\ \qquad\quad i := \mathbf{pred}\ (!i) \\ \qquad \mathbf{done}; \\ \qquad !r \end{array}$$

where **while** M **do** N **done** is syntactic sugar for $\mathcal{Y}\,(\lambda x.\ \mathbf{if}\ M\ (N;\ x)\ \mathbf{skip})$; and **not** and the arithmetic operation $*$ are defined in the obvious way.

Beyond first-order imperative programming, IA supports a combination of higher-order and state letting us write programs such as the following *strictness test*

$$\begin{array}{l} \lambda f^{\mathbb{U}\to\mathbb{U}}.\ \mathbf{newref}\ r\ \mathbf{in}\ f\ (r := 1); \\ \qquad\qquad \mathbf{if}\ (\mathbf{iszero}\ (!r))\ \mathbf{ff}\ \mathbf{tt} \\ : (\mathbb{U} \to \mathbb{U}) \to \mathbb{B} \end{array}$$

replying **ff** when applied to a constant function, and **tt** if the function calls its argument.

Our presentation of IA differs from the historical one in two inessential ways. First, as for PCF the **let** construct is usually missing from IA. But our **let** evaluates only terms of ground type, which is definable[2] in IA by letting **let** $x = M$ **in** N be sugar for

$$\mathbf{newref}\, r\, \mathbf{in}\, r := M;\ (\lambda x.\, N)\,(!r)\,,$$

illustrating an important aspect of IA references: they store *values* and not *computations*, so an expression $r := M$ must evaluate M fully before storing the value.

The second inessential difference is that our version of IA includes semaphores – which we need, because they are inavoidable to get full abstraction in the presence of parallelism also. For sequential IA, semaphores of course are definable via:

$$\begin{array}{rcl} \mathbf{newsem}\, s\, \mathbf{in}\, M & = & \mathbf{newref}\, s\, \mathbf{in}\, M \\ \mathbf{grab}\, M & = & \mathbf{if}\ (\mathbf{iszero}\, !M)\,(M := 1)\,\bot \\ \mathbf{release}\, M & = & \mathbf{if}\ (\mathbf{iszero}\, !M)\,\bot\,(M := 0)\,, \end{array}$$

which is fine only because IA is sequential: if a program attemps to grab a semaphore which is not free (or dually, to release a semaphore which is free), then computation will hang as there is no other thread to free the semaphore while the program waits.

Parallel PCF. *Parallel* PCF, or $\mathsf{PCF}_{/\!/}$ for short, corresponds in Figure 2.1 to PCF + parallelism. To PCF it adds mere parallel computation. This is made official by the operational semantics of Figure 2.2, which will indeed evaluate both branches of a parallel let *in parallel*. But keep in mind that this parallelism is not observable as there is no primitive in $\mathsf{PCF}_{/\!/}$ allowing one to observe the evaluation order. Because of that, giving denotational semantics to $\mathsf{PCF}_{/\!/}$ in isolation is a fairly trivial endeavour: any denotational semantics for standard PCF will do, interpreting the parallel let as

$$\mathbf{let}\ x_1 = M_1\, \mathbf{in}\, (\mathbf{let}\ x_2 = M_2\, \mathbf{in}\ N)\,,$$

but in this monograph we do not wish one denotational semantics for each fragment of $\mathsf{IA}_{/\!/}$, we want all fragments to be interpreted in the *same* denotational universe!

Idealized Parallel Algol. *Idealized Parallel Algol*, or $\mathsf{IA}_{/\!/}$ for short, corresponds in Figure 2.1 to PCF + parallelism + state. It is used in the game semantics community [Ghica and Murawski, 2008] as the (call-by-name) paradigmatic higher-order concurrent programming language with shared memory and semaphores.

To IA, our version of $\mathsf{IA}_{/\!/}$ adds parallelism, under the form of

$$\frac{\Gamma, x_1 : \mathbb{X}, x_2 : \mathbb{X} \vdash M : \mathbb{Y} \qquad \Gamma \vdash N_1 : \mathbb{X} \qquad \Gamma \vdash N_2 : \mathbb{X}}{\Gamma \vdash \mathbf{let} \begin{pmatrix} x_1 & = & N_1 \\ x_2 & = & N_2 \end{pmatrix} \mathbf{in}\ M : \mathbb{Y}}$$

[2] Up to observational equivalent; interestingly the two do not have the same causal behaviour!

a parallel let operation which evaluates N_1 and N_2 independently. We refer to this as *parallelism* and not *concurrency* because this **let** only adds parallel evaluation; in our view concurrency emerges from parallel computation plus shared resources.

The usual presentation of $\mathsf{IA}_{/\!/}$ instead has a parallel composition operator

$$\frac{\Gamma \vdash M : \mathbb{U} \qquad \Gamma \vdash N : \mathbb{U}}{\Gamma \vdash M \parallel N : \mathbb{U}}$$

which may be defined as syntactic sugar, via $M \parallel N = \mathbf{let} \begin{pmatrix} x & = & M \\ y & = & N \end{pmatrix} \mathbf{in\ skip}$.

In this work, we adopt the parallel let rather than merely parallel composition because we also consider parallelism without shared state; but without state $M \parallel N$ is not a relevant construction as M, N produce no value and can only communicate with the ambiant program via the shared memory. Again, this is not a significant change with respect to the usual formulation of $\mathsf{IA}_{/\!/}$, as the same behaviour may be obtained with

$$\mathbf{let} \begin{pmatrix} x_1 & = & N_1 \\ x_2 & = & N_2 \end{pmatrix} \mathbf{in}\ M \quad = \quad \mathbf{newref}\ r_1, r_2\ \mathbf{in}\ (r_1 := N_1) \parallel (r_2 := N_2);\ (\lambda x_1 x_2.\ M)\,(!r_1)\,(!r_2)$$

if $\parallel$ is primitive rather than the parallel let.

$\mathsf{IA}_{/\!/}$ is a very expressive language. The most impactful thing to notice is that the combination of parallelism and shared state make it non-deterministic: one can define

$$\vdash \mathbf{choice} = \mathbf{newref}\ r\ \mathbf{in\ let} \begin{pmatrix} x & = & (r := 1) \\ y & = & !r \end{pmatrix} \mathbf{in\ iszero}\ y : \mathbb{B}$$

exploiting a race in the memory to generate non-deterministic behaviour.

2.3 Generalities on Semantics

Finally, we include some reminders and settle on some terminology and notations regarding observational equivalence, and denotational semantics. In this section, we use $\mathcal{L}$ as the *ambiant programming language* – it can refer to PCF, IA, $\mathsf{PCF}_{/\!/}$, or $\mathsf{IA}_{/\!/}$.

Observational equivalence. A **$\mathcal{L}$-context** for the judgment $\Gamma \vdash A$ is a term $C[]$ of $\mathcal{L}$ with a hole written [], such that for any $\Gamma \vdash M : A$ in $\mathcal{L}$, we have $\vdash C[M] : \mathbb{U}$ obtained by the (non capture-avoiding) substitution of [] with M.

Two terms $\Gamma \vdash M, N : A$ of $\mathcal{L}$ are **$\mathcal{L}$-observationally equivalent** iff

$$M \simeq_{\mathcal{L}} N \quad \Leftrightarrow \quad \text{for all } C[] \text{ a } \mathcal{L}\text{-context for } \Gamma \vdash A, \quad (C[M] \Downarrow \quad \Leftrightarrow \quad C[N] \Downarrow)$$

We omit $\mathcal{L}$ when it is clear from the context. Observational equivalence is usually regarded as the canonical equivalence on programs: $\mathcal{L}$-observationally equivalent programs are intercheangeable as long as the evaluation context is in $\mathcal{L}$.

Denotational semantics. Observational equivalence is ultimately an *operational* notion of program identity; it rests on the evaluation relation $M \Downarrow$, itself defined via the operational semantics. An alternative to study program identity is given by *denotational semantics*, embedding programs in a syntax-independent mathematical universe.

Denotational semantics is usually formulated *categorically*: the language $\mathcal{L}$ is interpreted in a category $\mathcal{C}$ equipped with sufficient structure. More precisely, a denotational semantics in $\mathcal{C}$ consists in several components. First, there is an interpretation function

$$[\![-]\!] : \mathbf{Types} \to \mathcal{C}_0$$

for *types* – this usually exploits the categorical structure, with *e.g.* $[\![A \to B]\!] = [\![A]\!] \Rightarrow [\![B]\!]$ resting on a cartesian closed structure. This interpretation is extended to

$$[\![-]\!] : \mathbf{Contexts} \to \mathcal{C}_0$$

an interpretation of contexts, defined simply via $[\![\Gamma]\!] = \&_{(x_i : A_i) \in \Gamma} [\![A_i]\!]$. Now, we have

$$[\![-]\!] : \mathbf{Terms}(\Gamma \vdash A) \to \mathcal{C}([\![\Gamma]\!], [\![A]\!])$$

the interpretation of terms, defined by structural induction on the typing rules of $\mathcal{L}$ – for the simply-typed λ-calculus this interpretation typically follows a cartesian closed structure on $\mathcal{C}$, while the primitives of $\mathcal{L}$ (*e.g.* natural numbers, booleans, manipulations of state, *etc*) are interpreted by well-chosen morphisms in $\mathcal{C}$. By a **denotational model** of $\mathcal{L}$, we mean a category $\mathcal{C}$ along with the interpretation function $[\![-]\!]$ – sometimes focusing on $\mathcal{C}$ and leaving the latter implicit.

In this monograph, we assume the reader is familiar with basic categorical logic, including the interpretation of the simply-typed λ-calculus in a cartesian closed category. For details, the reader is referred to [Lambek and Scott, 1988].

Adequacy and full abstraction. A denotational model should reflect the operational evaluation. We say that an interpretation in a model $\mathcal{C}$ is **computationally adequate**, or simply *adequate* for short, if for all $\vdash M : \mathbb{X}$ a closed term of ground type,

$$M \Downarrow v \quad \Leftrightarrow \quad [\![M]\!] = [\![v]\!] ,$$

i.e. the denotational model faithfully predicts the evaluation of a program to a value. For general reasons, an adequate model is automatically *sound* with respect to observational equivalence, *i.e.* if $[\![M]\!] = [\![N]\!]$ then $M \simeq_{\mathcal{L}} N$; it is **fully abstract** if the converse also holds: if $M \simeq_{\mathcal{L}} N$ then we actually have $[\![M]\!] = [\![N]\!]$.

Intensional full abstraction. Given a denotational model $\mathcal{C}$, we may replay in $\mathcal{C}$ the syntactic definition of observational equivalence: we fix an object for observations, typically $[\![\mathbb{U}]\!]$ (still written $\mathbb{U}$ to alleviate notations). An *observation* on A is then any

$$\alpha \in \mathcal{C}(A, \mathbb{U})$$

playing the role of $\mathcal{L}$-contexts: $f, g \in \mathcal{C}(A, B)$ are **observationally equivalent** iff[3]

$$f \simeq_{\mathcal{C}} g \quad \Leftrightarrow \quad \text{for all } \alpha \in \mathcal{C}(A \Rightarrow B, \mathbb{U}),\ \alpha \circ f = \alpha \circ g$$

like syntactic observational equivalence. Quotienting $\mathcal{C}$ by this yields a new model, and $\mathcal{C}$ is **intensionally fully abstract** when this quotiented model is fully abstract.

The term "intensional full abstraction" was originally proposed by Abramsky, Jagadeesan and Malacaria [Abramsky et al., 2000] to better describe the "full abstraction" results for PCF. Intensional full abstraction is *not* full abstraction: it is full abstraction of the quotiented model, but this quotiented model is usually not very useful as it is obtained non-constructively – in particular, a model constructed in this way rarely helps directly in reasoning on observation equivalence.

In contrast, *intensional full abstraction* keeps the focus on the model pre-quotient (which, at least in the case of game semantics, turned out to be more important). There we understand it as meaning that the "intensional behaviour" of morphisms of $\mathcal{C}$ captures that of $\mathcal{L}$, or at least is sufficiently close that the morphisms of $\mathcal{C}$ have the *same distinguishing power* as contexts of $\mathcal{L}$ through the interpretation – there is no "abstraction leaks". Interestingly, it is in that sense that the term "full abstraction" is often understood in compilation – see *e.g.* [New et al., 2016].

In this monograph, we try to be careful in distinguishing *full abstraction* from *intensional full abstraction*, though we might occasionally slip and omit *intensional*.

[3] This definition is adequate when $\mathcal{C}(\top, \mathbb{U})$ has only two elements, for convergence and divergence. In some settings such as in concurrent games, this must be adapted to match syntactic observational equivalence.

Part I

Pointer Game Semantics

Introduction to Part I

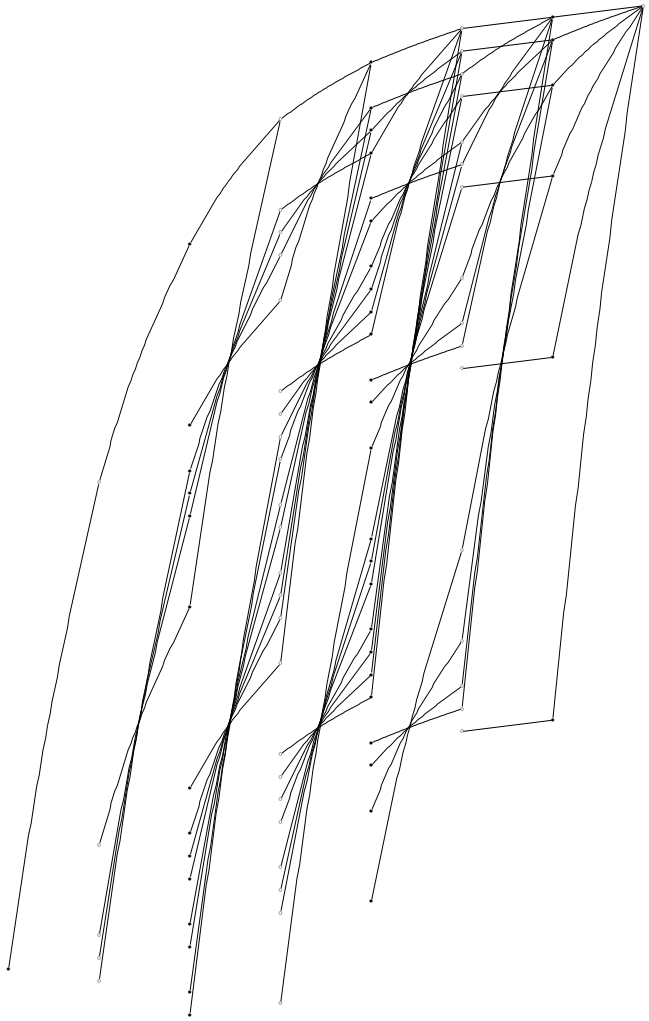


Figure 2.3: As a low-pressure system moves in from the west, we can expect widespread pointer showers to develop by Chapter 3, and these showers are likely to continue on and off throughout Part I.

The first part of this monograph is an introduction to "pointer game semantics", a phrasing intended to include both the traditional (sequential deterministic) Hyland-Ong games [Hyland and Ong, 2000], and the non-alternating games by Ghica and Muraski [Ghica and Murawski, 2008], which of course have much in common with the former.

There are two pedagogical difficulties when learning game semantics. The first is coming up with an intuition as to what *plays* and *strategies* represent in relation with the interactive behaviour of the program – the operational meaning of well-bracketing, innocence, *etc*. Understanding this is not enough to *work* with game semantics, but it suffices to see the scientific value and role of game semantics. The second difficulty is understanding the denotational machinery: the composition of strategies, the cartesian closed structure, *etc*. We organized our introduction to pointer games addressing these two difficulties separately, so that a reader wanting to get the *point* of game semantics without mastering the combinatorics of the categorical structure can do so.

Accordingly, Chapter 3 introduces pointer games and its main notions and achievements, independently of the mechanics of the interpretation of programs. The content *is* technical and mathematically precise, but omits the definition of the interpretation and the corresponding categorical structure. The chapter starts by introducing the basic intuitions behind game semantics, at first without committing to any particular technical setting, then motivates and introduces the main concepts of pointer games: *arenas*, *plays with pointers*, *strategies*, *etc*; resting on operational intuitions. Still relying on operational intuitions, we show how certain patterns in plays witness the use of computational effects such as state or control, and use this to motivate the notions of well-bracketing and innocence, and showcase the associated full abstraction results. Finally, we mention an alternative to pointers, *copy indices*, as in AJM games [Abramsky et al., 2000].

Then, Chapter 4 addresses the denotational interpretation of programs. We define the composition of strategies, constructing a cartesian closed category $\downarrow\uparrow$-**Strat**. We define the interpretation of IA in $\downarrow\uparrow$-**Strat**, and wrap up with a few complements.

Finally, Chapter 5 introduces *non-alternating pointer games* in the sense of Ghica and Murawski [Ghica and Murawski, 2008]. The exposition roughly following the pattern of the previous two chapters: first we define non-alternating strategies and illustrate how they capture the interactive behaviour of programs of $\mathsf{IA}_{/\!/}$; then we build the corresponding cartesian closed category $\circlearrowleft$-**Strat**, and define the interpretation of $\mathsf{IA}_{/\!/}$.

Chapter 3

First Steps in Game Semantics

In this first chapter, we give an introduction to game semantics – focusing in particular on so-called HO or *Hyland-Ong* games[1] [Hyland and Ong, 2000] presenting a selection of developments and landmark historical results. The presentation is technical, in that we give mathematically precise definitions and statements. However, we focus on operational intuitions, (temporarilly) remaining elusive on the main source of complexity in game semantics: the categorical structure and denotational definition of the interpretation of programs. This, in turn, shall be the topic of Chapter 4 – so altogether, Chapters 3 and 4 give an introduction to sequential alternating game semantics.

Of course, as an introduction to game semantics, this presentation is far from complete. Many developments in game semantics do not fit in the framework presented here; we also focus on semantics of programming languages and essentially omit the works on logics and proof systems, on the other side of the Curry-Howard correspondence.

3.1 Executions, Plays, and Strategies

This section aims to convey some intuitions, and to motivate and gradually put into place our choices of mathematical formalizations for the basic ingredients of game semantics.

3.1.1 Dialogues, Plays, and Simple Games

Game Semantics is designed as a way to give a clean mathematical description of the behaviour of *open* programs, as a back-and-forth interaction with an unspecified execution environment. More precisely, a program is interpreted as an aggregate of all

[1] Those are often called Hyland-Ong/Nickau or *HON* games, also acknowledging Nickau, who came up independently with a model of PCF [Nickau, 1994] that is essentially the same as Hyland and Ong's. That is fair, but then one should also acknowledge the work of Coquand, who also proposed independently definitions having much in common with those [Coquand, 1995] (though not for PCF). Rather than the cumbersome *HONC* games, we prefer to stick with *HO* games, which is well-recognized in the community and refers to the specific set of technical definitions introduced by Hyland and Ong.

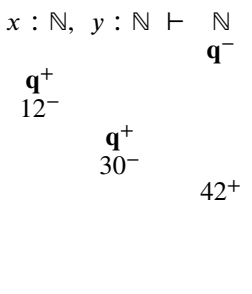


Figure 3.1: First-order

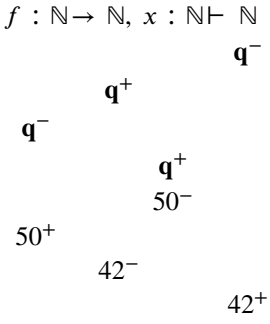


Figure 3.2: Higher-order

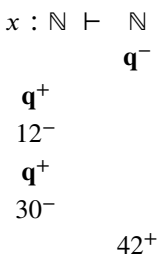


Figure 3.3: Repetition

its possible interactive behaviour with an execution environment. This is formalized as a two-player game between **Player** – which plays for the program under scrutiny; and **Opponent** – which plays the execution environment. Execution unfolds as those two players exchange *moves*, each of which represents an observable computational event.

First-order. As a first example, let us consider the following open PCF program:

$$x : \mathbb{N}, y : \mathbb{N} \vdash x + y : \mathbb{N}$$

For the sake of this discussion, we assume that the addition is computed on a left-to-right basis. From the point of view of standard operational semantics, there is no computation to perform here: the program is in *normal form*. *Game semantics* start where usual operational semantics stop, by *asking the environment the value of* x. One may then imagine a dialogue between the program and its environment, as follows:

O : "What is the value of $x + y$?"
P : "Evaluation is stuck: what is the value of x?"
O : "The value of x is 12."
P : "Evaluation is stuck: what is the value of y?"
O : "The value of y is 30."
P : "The value of $x + y$ is 42."

In game semantics, such a sequence is called a **play**, and is typically drawn in a diagram as in Figure 3.1. By convention, we assign to Player the positive polarity $+$, and to Opponent the negative polarity $-$. In this diagram and all game semantics diagrams to come, we label moves with their polarity. The flow of time is from top to bottom, and each column corresponds to a component of the interface of the program with its environment. Variable calls are represented with the symbol **q** which stands for "Questions". All other moves correspond to returns, and are called "Answers".

This dialogue is one possible interaction between our program of interest and an execution environment; there are countably many of those, for each values for x and y. Such interactions only display the events observable by the environment: the program $(x + 0) + y$ has exactly the same plays as $x + y$, the internal computation is ignored.

Higher-order. Let us now consider another example:

$$f : \mathbb{N} \to \mathbb{N}, x : \mathbb{N} \vdash f(x) : \mathbb{N}$$

One could expect Player to first prompt the environment for an actual function f from integers to integers. It is a central feature of game semantics that we instead have:

O : "What is the value of $f(x)$?"
P : "Evaluation is stuck: what is the output value of f?"
O : "What is the input value of f?"
P : "Evaluation is stuck: what is the value of x?"
O : "The value of x is 50."
P : "The input value of f is 50."
O : "The output value of f is 42."
P : "The value of $f(x)$ is 42."

which, in the usual style of game semantics, would appear as in Figure 3.2: we interact with f in a lazy fashion, never fully grasping the infinite object behind. In this way, game semantics reduces higher-order evaluation to an exchange of first-order tokens.

Simple games. But what are, formally, the plays of Figures 3.1 and 3.2?

In traditional game semantics, each type (or typing judgment) yields by interpretation a **game** whose rules describe the valid open executions on the interface given by that type. Exact formalizations vary wildly in the literature and we will see several of those in this monograph; as a stepping stone we first present what must be the simplest notion of game considered in the game semantics literature, **simple games** [Hyland, 1997]:

Definition 3.1.1. *A **simple game** comprises $A = (|A|, \mathrm{pol}_A, \mathrm{P}_A)$ with $|A|$ a set of **moves**, $\mathrm{pol}_A : |A| \to \{-, +\}$ a **polarity function**, and $\mathrm{P}_A \subseteq |A|^*$ a set of **plays** such that:*

non-empty: $\varepsilon \in \mathrm{P}_A$,
prefix-closed: *for all* $s \in \mathrm{P}_A$, *for all* $s' \sqsubseteq s$, *we have* $s' \in \mathrm{P}_A$,
alternating: *for all* $s = s_1 \dots s_n$, *for all* $1 \leq i \leq n-1$, $\mathrm{pol}_A(s_i) \neq \mathrm{pol}_A(s_{i+1})$,
negative: *for all* $s = s_1 \dots s_n$, *if* $n \geq 1$ *then* $\mathrm{pol}_A(s_1) = -$.

For instance, a game $\mathbb{B}$ for the booleans would have $|\mathbb{B}| = \{\mathbf{q}, \mathbf{tt}, \mathbf{ff}\}$, with $\mathrm{pol}_{\mathbb{B}}(\mathbf{q}) = -$, $\mathrm{pol}_{\mathbb{B}}(\mathbf{tt}) = \mathrm{pol}_{\mathbb{B}}(\mathbf{ff}) = +$, and $\mathrm{P}_{\mathbb{B}} = \{\varepsilon, \mathbf{q}, \mathbf{qtt}, \mathbf{qff}\}$. As above, we shall often write moves with their polarities, as in $|\mathbb{B}| = \{\mathbf{q}^-, \mathbf{tt}^+, \mathbf{ff}^+\}$ and $\mathrm{P}_{\mathbb{B}} = \{\varepsilon, \mathbf{q}^-, \mathbf{q}^-\mathbf{tt}^+, \mathbf{q}^-\mathbf{ff}^+\}$. However do keep in mind that this is merely for convenience: polarities are given by pol_A, and not intrinsic to moves, *i.e.* to the elements of $|A|$.

A program $M : A$ is then interpreted as a *strategy for Player* on the game for A:

Definition 3.1.2. *A **strategy** $\sigma : A$ on simple game A is a subset $\sigma \subseteq \mathrm{P}_A$ satisfying:*

non-empty:	$\varepsilon \in \sigma$,
prefix-closed:	*for all* $s \in \sigma$, *if* $s' \sqsubseteq s$, *then* $s' \in \sigma$,
receptive:	*for all* $s \in \sigma$, *if* $sa^- \in \mathrm{P}_A$, *then* $sa^- \in \sigma$,
deterministic:	*for all* $sa_1^+, sa_2^+ \in \sigma$, *we have* $a_1 = a_2$.

A strategy aggregates all executions that the program is prepared to make against any execution environment; prescribing, in each state, the next observable action. When presented an Opponent-ending play $sa^- \in P_A$, a strategy $\sigma : A$ checks whether sa^- appears in σ; and if so, if there is some extension sa^-b^+. If that is the case, σ plays b^+. For instance, the strategy for the constant $\mathbf{tt}$ would be given by the set:

$$\{\varepsilon, \mathbf{q}^-, \mathbf{q}^-\mathbf{tt}^+\} : \mathbb{B}\,.$$

Simple games are appealing; they seem to capture exactly the intuitions so far, without useless mathematical arabesque. But their expressiveness is limited in various ways, making them seldom used in game semantics. We review next a major constraint, playing a central role in the design of essentially all game semantics formalisms.

3.1.2 Replication and Thread Indexing

We consider the term $x : \mathbb{N} \vdash x + x : \mathbb{N}$; whose game semantics admits:

O	:	"What is the value of $x + x$?"
P	:	"Evaluation is stuck: what is the value of x?"
O	:	"The value of x is 12."
P	:	"Evaluation is stuck: what is the value of x?"
O	:	"The value of x is 30."
P	:	"The value of $x + x$ is 42."

drawn in game semantics style in Figure 3.3. This illustrates an important aspect of game semantics: it is *quantitative*, in that it replays separately each variable call. This may seem redundant but it is not: if the execution environment is non-deterministic or has access to state, then the two calls to x may obtain different values.

This does, however, ask the question of how this repetition is managed. Of course, the definition of simple games does not forbid simply repeating the same move. For instance, one could redefine the simple game for booleans as having the same set of moves and polarities as above, but plays given by the recursive definition:

$$\mathrm{P}_{\mathbb{B}} ::= \varepsilon \mid \mathbf{q}^- \mid \mathbf{q}^-\mathbf{tt}^+\mathrm{P}_{\mathbb{B}} \mid \mathbf{q}^-\mathbf{ff}^+\mathrm{P}_{\mathbb{B}}\,,$$

but this idea quickly fails as we climb the type-theoretic ladder. Consider the term:

$$f : \mathbb{B} \to \mathbb{B} \to \mathbb{B} \vdash f\,(f\,\mathbf{tt}\,\mathbf{ff})\,\mathbf{tt} : \mathbb{B}$$

and the play in Figure 3.4. On the right, we illustrate each position with an indicative, informal description of the corresponding operational state. Intuitively, Opponent

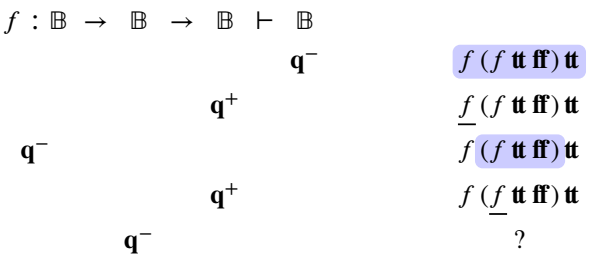


Figure 3.4: Ambiguity without thread-indexing

moves are requests to evaluate a certain subterm; so at Opponent moves we highlight the subterm currently being evaluated. Likewise, Player questions correspond to variable calls; so at Player moves we underline the corresponding variable occurrence.

On the first line, the initial Opponent move $\mathbf{q}^-$ prompts the evaluation of the whole term – so the whole term $f\,(f\,\mathbf{tt}\,\mathbf{ff})\,\mathbf{tt}$ is highlighted. In call-by-name, this prompts a Player question $\mathbf{q}^+$ corresponding to requesting an output for the first occurrence of f, which is therefore underlined in the second line. In our example, Opponent then plays $\mathbf{q}^-$ in the column corresponding to the first argument of f – so the first occurrence of f requests an evaluation of its first argument, highlighted in the third line. Yet again, the highlighted sub-term starts with an occurrence of f, so the next Player move, on the fourth line, is a move $\mathbf{q}^+$ asking an output for f, seemingly identical to the move on the second line – but the variable occurrence underlined is not the same.

Now, in our example, the next move is an Opponent move corresponding to f calling its *second* argument. But which occurrence of f? We have seen two of those, on the second and fourth lines! As this last $\mathbf{q}^-$ reacts to the $\mathbf{q}^+$ in the fourth line, it is natural to expect that it refers to it so that the operational state is $f\,(f\,\mathbf{tt}\,\mathbf{ff})\,\mathbf{tt}$. But in a sufficiently expressive language, it is also possible to find an environment in which the Opponent asks to evaluate the *first* occurrence of f, *i.e.* in which case the operational state is $f\,(f\,\mathbf{tt}\,\mathbf{ff})\,\mathbf{tt}$! Because there are two copies of the move $\mathbf{q}^+$ calling to f, Opponent has, intuitively, two available copies of the move $\mathbf{q}^-$ requesting the evaluation of the second argument of f; and without further structure, we cannot disambiguate[2].

The literature proposes two ways to solve this: *pointers*, or *copy indices*.

3.1.3 Pointer Games

The most widely used solution to this problem, used in particular in Hyland-Ong games [Hyland and Ong, 2000], is to enrich plays with *pointers* providing exactly the missing information. For instance, we show in Figure 3.5 two *plays with pointers* disambiguat-

[2]To be precise, the usual arrow construction of simple games [Hyland, 1997] enforces a property called *local alternation* that only allows one of these options. But the phenomenon does occur at third-order, or in game semantics not enforcing local alternation such as that for higher-order state [Abramsky et al., 1998].

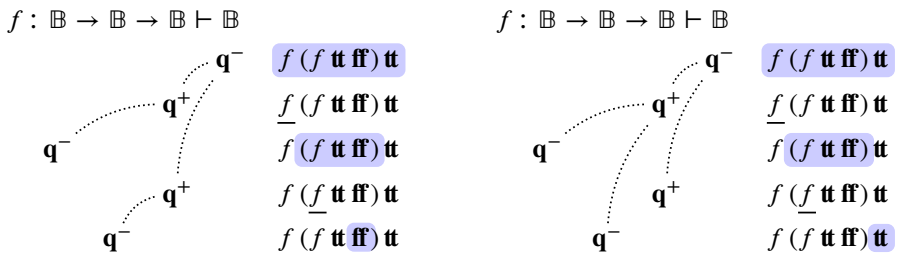


Figure 3.5: Two plays with pointers disambiguating Figure 3.4.

ing Figure 3.4. On the left hand side, Player is next to play $\mathbf{ff}^+$ while on the right hand side, Player is to play $\mathbf{tt}^+$. Pointers provide the missing *thread indexing*.

Adding pointers to plays require reworking the notion of game, which must now specify which pointers are allowed. This is achieved by switching to *arenas*:

Definition 3.1.3. *An **arena** comprises $A = (|A|, \mathrm{pol}_A, \leq_A)$ with $|A|$ a countable set of **moves**, $\mathrm{pol}_A : |A| \to \{-,+\}$ a **polarity function**, $\leq_A$ a **causality** partial order, s.t.:*

forestial:	*for all $a_1, a_2, a \in |A|$, if $a_1, a_2 \leq_A a$, then $a_1 \leq_A a_2$ or $a_2 \leq_A a_1$,*
well-founded:	*there is no infinite descending $a_1 >_A a_2 >_A a_3 >_A \cdots$,*
negative:	*if $a \in |A|$ is minimal for $\leq_A$, then $\mathrm{pol}_A(a) = -$,*
alternating:	*for all $a_1, a_2 \in |A|$, if $a_1 \rightarrowtriangle_A a_2$, then $\mathrm{pol}_A(a_1) \neq \mathrm{pol}_A(a_2)$,*

where $a_1 \rightarrowtriangle_A a_2$ means $a_1 <_A a_2$ and $a_1 \leq_A a \leq_A a_2$ implies $a = a_1$ or $a = a_2$.
This shall be completed (with Question/Answer labelling) in Definition 3.2.4.

We call $\rightarrowtriangle_A$ the **immediate causality** relation on A. Historically it has been written $\vdash_A$ in Hyland-Ong games, and called the *enabling relation* [Hyland and Ong, 2000]; but in this monograph we prefer to consider the partial order $\leq_A$ as primitive.

The arena for the typing judgment $f : \mathbb{B} \to \mathbb{B} \to \mathbb{B} \vdash \mathbb{B}$ appears in Figure 3.6. We have annotated the type to highlight the correspondence between components of the arena and ground type occurrences. Usually these annotations are not present in diagrams for arenas or plays, but the same information is conveyed by displaying the type, and placing each move under the corresponding type component.

Both simple games and arenas are forests; but should be read differently: while in simple games an execution amounts to exploring a branch of the forest, an arena simply presents the set of observable computational events along with their hierarchical dependencies. For instance, in Figure 3.6, Opponent may start computation with $\mathbf{q}^-_4$. This enables three moves: $\mathbf{tt}^+_4$ and $\mathbf{ff}^+_4$, as Player may implement a constant function answering without evaluating its arguments. And, $\mathbf{q}^+_3$ if Player wishes to evaluate f. In turn, this lets Opponent return $\mathbf{tt}^-_3$ or $\mathbf{ff}^-_3$, or evaluate one of its arguments – and so on.

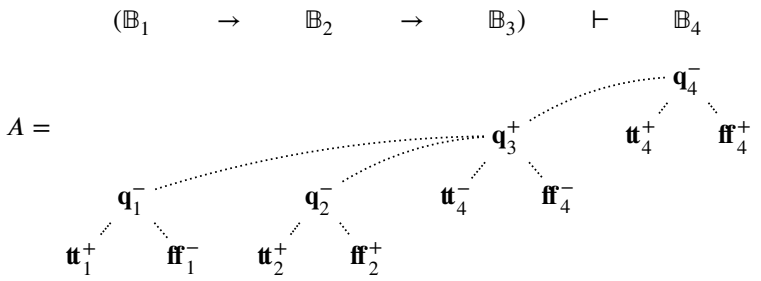


Figure 3.6: The arena for the sequent $f : \mathbb{B} \to \mathbb{B} \to \mathbb{B} \vdash \mathbb{B}$

Clearly, executions will not be merely walks on this tree: typically a function evaluates some of its arguments, possibly several times, *then* returns a value. Accordingly, executions may hop around branches of the arena, as captured formally by:

Definition 3.1.4. *A **pointing sequence** on arena A is $s = s_1 \dots s_n \in |A|^*$, where each s_i may come with a **pointer** to some s_j with $j < i$. It is **legal** if it satisfies:*

justified: *if s_i points to s_j, then $s_j \rightarrowtriangle_A s_i$,*
initialized: *if s_i has no pointer, then it is minimal in A.*

*A **(alternating) legal play** on A is a legal pointing sequence which is:*

alternating: *for all $1 \leq i \leq n-1$, $\mathrm{pol}_A(s_i) \neq \mathrm{pol}_A(s_{i+1})$,*

We write $\downuparrows$-Pla$\widehat{\text{ys}}(A)$ for the set of alternating legal plays on A.

For instance, the following is an alternating legal play in the arena of Figure 3.6:

$$\mathbf{q}_4^- \leftarrow \mathbf{q}_3^+ \leftarrow \mathbf{q}_1^- \quad \mathbf{q}_3^+ \quad \mathbf{q}_2^- \tag{3.1}$$

but it is not usually drawn like that: instead it is drawn from top to bottom instead of left to right, pointers are represented via dotted lines and without arrow heads (as they must always go back in time), and the indices $1, 2, 3, 4$ are omitted, the same information being conveyed by placing moves under the corresponding type component. Following this convention, the alternating legal play above is drawn as in Figure 3.5 on the rhs.

For the rest of this chapter, unless specified otherwise, by *play* we always mean *alternating legal play*. The arguably heavy notation $\downuparrows$-Pla$\widehat{\text{ys}}(A)$ invites an explanation: say that a play $s \in \downuparrows$-Pla$\widehat{\text{ys}}(A)$ is **in state** O if it has even length, *i.e.* it is Opponent's turn to play. Likewise, an odd-length play is **in state** P. The two arrows $\downuparrows$ suggest alternation as the control alternates between two states O and P, while the arrow $\curvearrowleft$ evokes pointers as in (3.1). This is the first instance of a scheme followed throughout this monograph for different kinds of plays – alternating or not; pointing or not.

We introduce some notations on plays. We use s, t, u, v to range over plays, and ε for the empty play. We write $\sqsubseteq$ for the prefix ordering on plays, which includes compatibility with pointers. We often extend plays by juxtaposing moves as in sab where it is

understood that s is a play and a, b are moves, leaving their possible pointer implicit. When writing equalities such as $sab = t$, it is understood that pointers are required to coincide as well. If s is a play, we write it $s_1 \dots s_n$ and refer to s_i for individual moves. If s_i points to s_j (with $j < i$), we call s_j the **justifier** of s_i.

Plays with pointers are a powerful formalism, but one drawback is that unlike in simple games, they are no longer simply strings of moves. The reader may notice that they are referred to in english, as in "s_j points to s_i" rather than via a specific rigorous representation, *e.g.* conveyed by a partial function from indices to indices. Debatable as it is, this is the standard practice in Hyland-Ong games – manipulations on plays with pointers are always cumbersome, but can quickly become unwieldy especially if done formally, as pointers must be constantly reindexed or reassigned[3].

We can define strategies on an arena, similarly to Definition 3.1.2:

Definition 3.1.5. *For A an arena, an **alternating strategy** $\sigma : A$, or **$\downarrow\uparrow$-strategy** for short, is a set $\sigma \subseteq \downarrow\uparrow\text{-}\widehat{\mathsf{Plays}}(A)$ satisfying the following conditions:*

$$\begin{array}{rl} \text{non-empty:} & \epsilon \in \sigma\,, \\ \text{prefix-closed:} & \forall s \sqsubseteq t \in \sigma,\ s \in \sigma\,, \\ \text{receptive:} & \forall s \in \sigma,\ sa^- \in \downarrow\uparrow\text{-}\widehat{\mathsf{Plays}}(A) \implies sa^- \in \sigma\,, \\ \text{deterministic:} & \forall sa^-b_1^+, sa^-b_2^+ \in \sigma,\ sab_1 = sab_2 \end{array}$$

We refer to an alternating strategy simply as a *strategy* when it causes no confusion.

Copy indices. We are due for a discussion on the other main solution to the thread indexing problem, *copy indices*. But we prefer to postpone this to Section 3.4, and instead we carry on with the construction of pointer game semantics.

3.2 Carving Out Innocent Strategies

Now that we know how to play, we may explore the model and examine its computational meaning – starting from PCF, the core of all languages used in this monograph.

3.2.1 The Strategies of PCF

Interpretation of types. Following Section 2.2, we consider in this monograph a version of PCF with one constructor $\Rightarrow$ for the function space, and three base types ($\mathbb{U}$, $\mathbb{B}$ and $\mathbb{N}$). Accordingly, there are three *basic arenas* $\mathbf{U}$, $\mathbf{B}$ and $\mathbf{N}$, described in Figures 3.7, 3.8 and 3.9 respectively – ignoring for now the annotation of moves with $\mathcal{Q}$ and $\mathcal{A}$.

In light of the discussion above, these arenas should be rather self-explanatory: Opponent prompts computation with the initial move $\mathbf{q}^-$, and Player answers with a value. One important common point is that all three are *well-opened*:

[3] There have been proposals for a more rigorous handling of pointers, notably resting on nominal sets in [Gabbay and Ghica, 2012]. The issue does not appear in AJM games, where pointers are absent altogether – though this absence limits the expressiveness of AJM games. In the model proposed later in this monograph, *concurrent games*, the issue is avoided as pointers are present, but derived rather primitive.

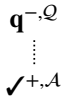


Figure 3.7: **U**

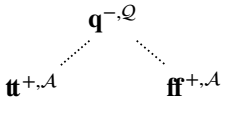


Figure 3.8: **B**

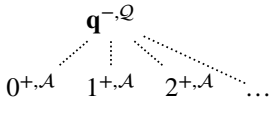


Figure 3.9: **N**

Definition 3.2.1. *An arena A is **well-opened** if and only if it has exactly one minimal move, called the **initial move** of A, and written* $\mathrm{init}(A)$.

In fact, arenas arising as the interpretation of types of PCF will always be well-opened. This lets us describe an easy construction for the arrow of two arenas.

Definition 3.2.2. *Consider A_1, A_2 arenas with A_2 well-opened.*
*Then, the (well-opened) **arrow** arena $A_1 \Rightarrow A_2$ is defined via components:*

$$\begin{array}{rrcl}
\text{moves:} & |A_1 \Rightarrow A_2| & = & |A_1| \ + \ |A_2| \\
\text{causality:} & (i,a) \leq_{A_1 \Rightarrow A_2} (j,a') & \Leftrightarrow & (i = j \ \&\ a \leq_{A_i} a') \ \vee \ (i = 2 \ \&\ a = \mathrm{init}(A_2)) \\
\text{polarities:} & \mathrm{pol}_{A_1 \Rightarrow A_2}(i,a) & = & (-1)^i \mathrm{pol}_{A_i}(a)\,.
\end{array}$$

In $A \Rightarrow B$, the two arenas are first put side by side, with polarities reversed in A (so that Player and Opponent exchange roles). Then, A is set to depend on the initial move of B. This means that once the initial move is played in $A \Rightarrow B$, Player has two choices: either to play in B (ignoring A), or to play some minimal move in A.

Altogether, this yields an interpretation of types of PCF as arenas, via

$$[\![A \to B]\!] = [\![A]\!] \Rightarrow [\![B]\!]$$

with $[\![\mathbb{U}]\!] = \mathbf{U}$, $[\![\mathbb{B}]\!] = \mathbf{B}$ and $[\![\mathbb{N}]\!] = \mathbf{N}$. The reader is invited to check that for instance, $[\![(\mathbb{B} \to \mathbb{B} \to \mathbb{B}) \to \mathbb{B}]\!]$ is indeed the arena shown in Figure 3.6.

Interpretation of terms. For each term $\vdash M : A$, there is a strategy

$$[\![M]\!] : [\![A]\!]\,,$$

its *interpretation* – it is computed by induction on M following the methodology of denotational semantics, via a cartesian closed category of arenas and strategies. This will appear in due course but we postpone it for now: it takes a while to set up this machinery, which is not needed to build up an intuition for what game semantics *do*.

Though the strategy of a term is computed denotationally, an experienced game semanticist will be able to directly list its plays, without going through the intricate definition of the interpretation. This is because plays are executions: rather than denotationally, they can be obtained directly from the term by operational means[4]. This is

[4]Though making this formal is not easy, and has been the topic of an active line of work around *operational game semantics* [Danos et al., 1996, Jaber, 2015, Ghica and Tzevelekos, 2012, Levy and Staton, 2014].

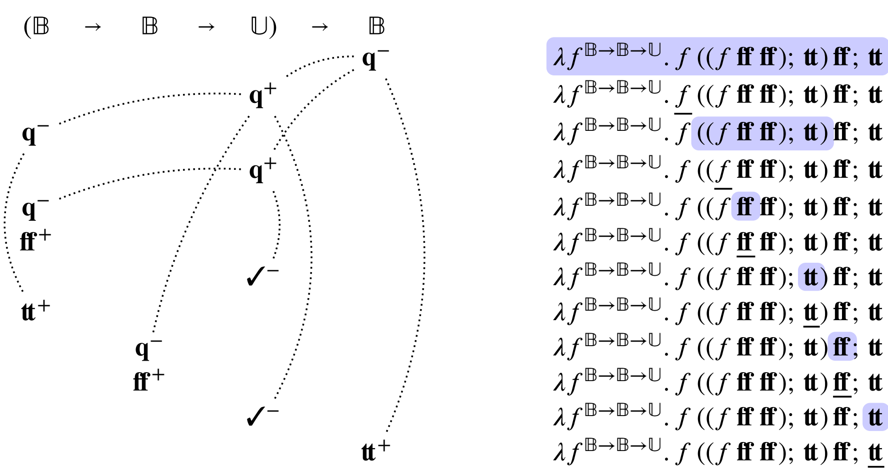


Figure 3.10: Illustration of the operational content of game semantics

illustrated in Figure 3.10. As before, Opponent moves trigger the evaluation of a subterm, which is highlighted. The following Player move then corresponds to the head (*i.e.* leftmost) variable occurrence (or constant) of the subterm being evaluated. The pointers from Player moves correspond to the stage where the variable in head position was abstracted, or to the function call being returned by the value in head position.

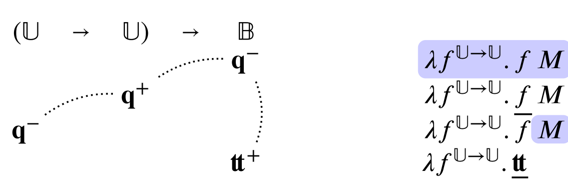
The standard way of stating that the model reflects computation is:

Proposition 3.2.3. *The interpretation is computationally adequate for* PCF*:*
For each $\vdash M : \mathbb{X}$*, we have* $M \Downarrow$ *iff* $[\![M]\!]$ *has an answer to the initial move.*

Plays give a mathematically, syntax-independent notion of interactive execution. In principle, a play may be realized by many different programs, but are all plays realizable by some program? By a program from PCF, or from a more expressive language?

3.2.2 Well-Bracketing

For instance, is the following play a possible execution of a term?



We argue informally why this *cannot* be an execution in PCF. The first action of the term is to ask its argument, so within PCF it must have the form $\lambda f^{\mathbb{U}\to\mathbb{U}}.\, f\ M$; we

annotate the figure with the corresponding operational state as in Figure 3.10. In the last line, **tt** at toplevel indicates the overall computation has terminated to **tt**. This is confusing, since in the third line, only the argument of f was under evaluation. How can evaluating the argument of f cause the whole computation to terminate?

Nevertheless, this play is indeed a realistic execution, for the term

$$\lambda f^{\mathbb{U}\to\mathbb{U}}.\,\mathbf{callcc}\,(\lambda k^{\mathbb{B}\to\mathbb{U}}.\,f\,(k\,\mathbf{tt});\,\bot) : (\mathbb{U}\to\mathbb{U})\to\mathbb{B}$$

where **callcc** is the *call-with-current-continuation* primitive originating in *Scheme*, and which famously may be typed with Peirce's law [Griffin, 1990]. We shall not need the precise operational semantics of **callcc**, but informally **callcc** M immediately calls M, feeding it a special function k, the "continuation". When the continuation is called with value v, **callcc** interrupts M and returns v at toplevel, breaking the call stack discipline.

Can the play above be realised without **callcc** (or some other analogous *control operator*)? We can show that the answer is no, by capturing those plays that "respect the call stack discipline", and refining the whole interpretation to show that the interpretation of terms only yields such plays. This is the goal of the notion of *well-bracketing*.

First we enrich arenas:

Definition 3.2.4. *A **Question/Answer labeling** on arena A is a function*

$$\lambda_A : |A| \to \{\mathcal{Q}, \mathcal{A}\}$$

which satisfies the following conditions:

question-opening:	*if $a \in |A|$ is minimal, then $\lambda_A(a) = \mathcal{Q}$,*
answer-closing:	*if $\lambda_A(a) = \mathcal{A}$, then a is maximal for $\leq_A$,*

Questions intuitively correspond to variable calls, while *Answers* correspond to returns. From now on, we assume that all **arenas** have a Question/Answer labeling. For basic arenas $\mathbf{U}, \mathbf{B}$ and $\mathbf{N}$, the labelling is as indicated in Figures 3.7, 3.8 and 3.9: the initial Opponent move is a Question, and the different values are Answers. This is extended to the arrow arena with $\lambda_{A\Rightarrow B}(1, a) = \lambda_A(a)$ and $\lambda_{A\Rightarrow B}(2, b) = \lambda_B(b)$.

If $s \in \downarrow\uparrow\text{-}\widehat{\mathsf{Plays}}(A)$ and s_i is an answer, it cannot be minimal in A by *question-opening*. Its antecedent in A appears in s as some s_j with $j < i$, and is a question by *answer-closing*. We say that s_i **answers** s_j. If a question in s has an answer in s we say it is **answered** in s. The last unanswered question of s, if any, is the **pending question**.

We now capture executions respecting the call stack discipline:

Definition 3.2.5. *Consider A an arena, and $s \in \downarrow\uparrow\text{-}\widehat{\mathsf{Plays}}(A)$ an alternating legal play. It is **well-bracketed** if for all prefix $ta^{\mathcal{A}} \sqsubseteq s$, a answers the pending question of t.*

All plays encountered in the monograph until now are well-bracketed, with the exception of the example at the beginning of Section 3.2.2. We can then define:

Definition 3.2.6. *Let $\sigma : A$ be a strategy on A.*
*It is **well-bracketed** iff for all $sa^{+} \in \sigma$, if s is well-bracketed then so is sa.*

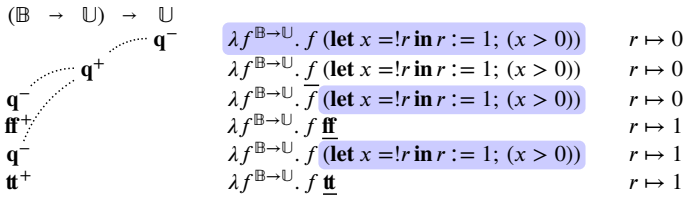


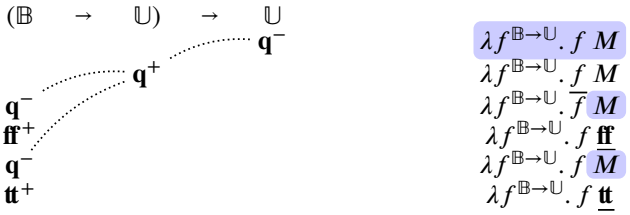
Figure 3.11: A play for a strategy with references

Thus, a well-bracketed strategy is never the first to break the call stack discipline.

We shall see that all the operations on strategies used in the interpretation of PCF (notably, composition) preserve well-bracketing. In the execution at the start of Section 3.2.2, Player is the first to break well-bracketing; so this execution cannot occur in a well-bracketed strategy, and is therefore – as expected – unrealizable in PCF.

3.2.3 Visibility and Innocence

Likewise, is this play a possible execution of a term?



This also seems unfeasible in PCF. Again, on the rhs we show, assuming a term realising this play, its corresponding operational states. At the third and fifth moves, the *same* subterm is being evaluated; yet we get two distinct answers. In an extension of PCF with a primitive $\vee$ for non-deterministic choice, this play would be realisable by $\lambda f^{\mathbb{B}\to\mathbb{U}}.\, f\,(\mathbf{tt} \vee \mathbf{ff})$. But does it make computational sense in a *deterministic* language?

Once more, the answer is yes: the play above describes a valid execution of the term

$$\lambda f^{\mathbb{B}\to\mathbb{U}}.\, \mathbf{newref}\ r\ \mathbf{in}\ f\ (\mathbf{let}\ x =! r\ \mathbf{in}\ r\ :=1;\ (x>0)) : (\mathbb{B}\to\mathbb{U})\to\mathbb{U}$$

in IA (see Section 2.2), *i.e.* PCF extended with references: $\mathbf{newref}\ r\ \mathbf{in}\ M$ allocates a reference r initialized to 0. We show in Figure 3.11 an operational description.

Yet, again, this play cannot be realised in PCF. To show this, we give a version of *innocence* [Hyland and Ong, 2000], formalizing that without state, independent evaluations of the same subterm must yield the same response. The first step is a mathematical way to state that two Opponent-ending plays "evaluate the same subterm", like the two prefixes of the play of Figure 3.11 terminating with a $\mathbf{q}^-$ on the left.

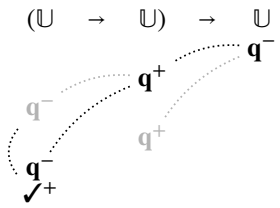


Figure 3.12: A non P-visible play

The operation computing (a mathematical notion of) "current subterm" is:

Definition 3.2.7. *Let $s \in \downarrow\uparrow\text{-}\widehat{\mathsf{Plays}}(A)$. Its **P-view** is the subsequence defined by:*

$$\begin{array}{rcll} \ulcorner \varepsilon \urcorner & = & \varepsilon & \\ \ulcorner s a^+ \urcorner & = & \ulcorner s \urcorner a^+ & \\ \ulcorner s a_1^+ s' a_2^- \urcorner & = & \ulcorner s \urcorner a_1^+ a_2^- & \text{if } a_2 \text{ points to } a_1, \\ \ulcorner s a_2^- \urcorner & = & a_2^- & \text{if } a_2 \text{ is minimal,} \end{array}$$

where pointers are preserved if their target remains in the subsequence.

We take the immediate prefix for P-ending plays and follow the pointer for O-ending plays. For instance, the prefixes of length 3 and 5 of the play on Figure 3.11 have the same P-view, capturing that they correspond to the same subterm. This is a fundamental definition – really, one of the cornerstones of HO games [Hyland and Ong, 2000].

But in order to use this definition it better be a well-defined operation on plays, which is not yet the case: for instance, in Figure 3.12 we gray out moves not selected in computing the P-view of $s \in \downarrow\uparrow\text{-}\widehat{\mathsf{Plays}}(A)$ for $A = [\![(\mathbb{U} \to \mathbb{U}) \to \mathbb{U}]\!]$. The subsequence of $\ulcorner s \urcorner$ in black is an alternating sequence of moves as needed, but it does not contain the justifier of $\checkmark^+$. So $\checkmark^+$ has no justifier in $\ulcorner s \urcorner$, which means that $\ulcorner s \urcorner \notin \downarrow\uparrow\text{-}\widehat{\mathsf{Plays}}(A)$ as $\checkmark^+$ is not minimal in A. Accordingly, we focus on those plays where $\ulcorner - \urcorner$ is defined:

Definition 3.2.8. *A play $s \in \downarrow\uparrow\text{-}\widehat{\mathsf{Plays}}(A)$ is **P-visible** if $\forall t \sqsubseteq s$, $\ulcorner t \urcorner \in \downarrow\uparrow\text{-}\widehat{\mathsf{Plays}}(A)$. Likewise, a strategy $\sigma : A$ is **P-visible** (or just **visible**) iff all its plays are P-visible.*

We write $\mathbf{Vis}(A)$ *for the set of P-visible strategies on arena A.*

So, "computing P-views never drops pointers", or "Player always points in the P-view". On P-visible $s \in \downarrow\uparrow\text{-}\widehat{\mathsf{Plays}}(A)$, the P-view always yields a (P-visible) play. We now define *innocent strategies* as those that behave the same in any situation where the same subterm is being evaluated, *i.e.* whose behaviour only depends on the *P-view*:

Definition 3.2.9. *A P-visible strategy $\sigma : A$ is **innocent** if it satisfies:*

innocence: *for all $sa^+ \in \sigma$ and $t \in \sigma$, if $\ulcorner s \urcorner = \ulcorner t \urcorner$ then $ta^+ \in \sigma$ with $\ulcorner sa \urcorner = \ulcorner ta \urcorner$.*

We write $\mathbf{Inn}(A)$ *for the set of innocent strategies on arena A.*

The clause $\ulcorner sa \urcorner = \ulcorner ta \urcorner$ serves to ensure that a has the "same" justifier in s and in t. Again we shall see that innocence is preserved by all operations involved in the interpretation of PCF so that any term of PCF is interpreted by an innocent strategy.

The play at the beginning of Figure 3.2.3 contains a direct failure of innocence – hence, it cannot appear in an innocent strategy and thus cannot be realised in PCF.

O-visibility. Whenever a condition constrains the behaviour of Player, one may consider the dual concerning Opponent. As a short digression we consider *O-visibility*, which will be technically helpful later on in this chapter. First we define:

Definition 3.2.10. *Let $s \in \Updownarrow\text{-}\widehat{\mathsf{Plays}}(A)$. Its **O-view** is the subsequence defined by:*

$$\begin{array}{rcll} \llcorner \varepsilon \lrcorner & = & \varepsilon & \\ \llcorner sa^- \lrcorner & = & \llcorner s \lrcorner a^- & \\ \llcorner sa_1^- s' a_2^+ \lrcorner & = & \llcorner s \lrcorner a_1^- a_2^+ & \text{if } a_2 \text{ points to } a_1, \end{array}$$

where pointers are preserved if their target remains in the subsequence.

The definition mirrors that of the P-view, with one missing clause for the initial move: this mirrors an assymmetry in the definition of arenas/plays, where initial moves must always be negative. As for P-views, this operation is well-defined only for:

Definition 3.2.11. *A play $s \in \Updownarrow\text{-}\widehat{\mathsf{Plays}}(A)$ is **O-visible** if $\forall t \sqsubseteq s$, $\llcorner t \lrcorner \in \Updownarrow\text{-}\widehat{\mathsf{Plays}}(A)$.*

O-visibility captures those plays where the external environment acts in a P-visible way, *i.e.* does not have access to effects that break P-visibility, see Section 3.3.3. We say that $s \in \Updownarrow\text{-}\widehat{\mathsf{Plays}}(A)$ is **visible** when it is both P-visible and O-visible.

One can also apply the same principle to innocence and define O-innocence capturing these plays where Opponent acts innocently, but this turns out less useful.

3.2.4 Finite Definability and Full Abstraction

At this point, via well-bracketing and innocence we have effectively eliminated all PCF non-definable behaviour. The theorem making this formal is called *finite definability*; we show the main steps of its proof as well as how it entails *intensional full abstraction*.

Meager innocent strategies. We have used the P-view operation as a mathematical attempt to capture the "current subterm" of a term reached in a play. Accordingly, it makes sense to attempt recovering syntax by examining those plays obtained by $\ulcorner - \urcorner$. Clearly, for any P-visible $s \in \Updownarrow\text{-}\widehat{\mathsf{Plays}}(A)$, we have $\ulcorner \ulcorner s \urcorner \urcorner = \ulcorner s \urcorner$, so the plays obtained by $\ulcorner - \urcorner$ are exactly those invariant under $\ulcorner - \urcorner$, easily characterized as those P-visible plays where *Opponent always points to the previous move* – call these the **P-views**.

Following this discussion, it should be no surprise that the P-views are the key to reconstructing a term from a strategy. If $\sigma : A$ is P-visible, we define

$$\urcorner \sigma \ulcorner = \{ \ulcorner s \urcorner \mid s \in \sigma \}$$

the set of **P-views** of σ. If σ is innocent, it is a simple fact that $\ulcorner\sigma\urcorner \subseteq \sigma$. Moreover, σ can then be recovered as the set of P-visible plays s such that for all $t \sqsubseteq s$, we have $\lceil t \rceil \in \ulcorner\sigma\urcorner$. Thus, sets of P-views are an equivalent representation of innocent strategies.

It is informative to examine those sets of P-views matching innocent strategies:

Definition 3.2.12. *A **meager innocent strategy** on A is a set V of P-views on A, s.t.:*

non-empty:	$\varepsilon \in V$,
prefix-closed:	*for all* $s \in V$ *and* $t \sqsubseteq s$, *then* $t \in V$
receptive:	*for all* $s \in V$, *if* sa^- *is a P-view then* $sa \in V$,
deterministic:	*for all* $sa_1^+, sa_2^+ \in V$ *we have* $sa_1 = sa_2$.

*Finally, V is **well-bracketed** if all its plays are well-bracketed.*
We write $\mathbf{Meager}(A)$ *for the set of meager strategies on arena A.*

From the above and routine verification, we obtain:

Proposition 3.2.13. *Consider A an arena. Then we obtain a bijection:*

$$\ulcorner - \urcorner \quad : \quad \mathbf{Inn}(A) \quad \simeq \quad \mathbf{Meager}(A)\,,$$

and moreover σ is well-bracketed iff $\ulcorner\sigma\urcorner$ is.

Proof. The only subtlety is that is $\ulcorner\sigma\urcorner$ is well-bracketed, then σ is well-bracketed. This follows from the preliminary observation, proved by induction on s, that if s is well-bracketed then its pending question (if any) always appears in the P-view $\lceil s \rceil$. □

In a certain sense, meager innocent strategies *are* syntax – or at least, each finite meager innocent strategy may be equivalently written in the syntax of PCF:

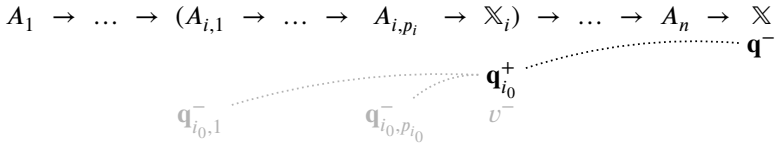
Finite definability. Let us say that an innocent strategy $\sigma : A$ is **finite** if the corresponding meager strategy $\ulcorner\sigma\urcorner$ has finitely many even-length plays. The **size** of finite σ is then the cardinal of the set of $s \in \ulcorner\sigma\urcorner$ of even length.

Theorem 3.2.14. *Let A be a* PCF *type, and $\sigma : [\![A]\!]$ be finite well-bracketed innocent. Then, there is a* PCF *term $\vdash M : A$ s.t. $[\![M]\!] = \sigma$.*

Sketch. For more details than this sketch, see [Hyland and Ong, 2000].
W.l.o.g., the type A has the form $A_1 \to \cdots \to A_n \to \mathbb{X}$ where for each $1 \leq i \leq n$,

$$A_i = A_{i,1} \to \cdots \to A_{i,p_i} \to \mathbb{X}_i\,.$$

We reason by induction on the size of σ. If σ has no reaction to the (unique) minimal $\mathbf{q}^-$ in $\mathbb{X}$ (*i.e.* $\sigma = \{\varepsilon, \mathbf{q}^-\}$), any diverging term will do. Otherwise, by determinism there is exactly one move a^+ s.t. $\mathbf{q}^- a^+ \in \sigma$. If a^+ is an answer v^+ on $\mathbb{X}$, then M is the matching constant. Otherwise, a^+ is the initial $\mathbf{q}_{i_0}^+$ in some A_{i_0}. The situation is



with, in grey, the possible extended P-views. For each, there is a residual substrategy.

We extract those – first, if $\mathbf{q}^+_{i_0}$ immediately returns. For each value v in $\mathbb{X}_i$, we form

$$\ulcorner\!\urcorner \sigma_v \urcorner\!\ulcorner = \{\mathbf{q}^- s \mid \mathbf{q}^- \mathbf{q}^+_{i_0} v^- s \in \ulcorner \sigma \urcorner\},$$

a meager innocent strategy on $[\![A]\!]$ of size strictly lesser than σ. By IH there is $\vdash M_v : A$ with $[\![M_v]\!] = \sigma_v$. As σ is finite, there are finite many v s.t. σ_v is non-diverging.

Alternatively, for all $1 \leq j \leq p_{i_0}$, we consider P-views $\mathbf{q}^- \mathbf{q}^+_{i_0} \mathbf{q}^-_{i_0,j} s \in \ulcorner\!\urcorner \sigma \urcorner\!\ulcorner$ where as a P-view, s answers neither $\mathbf{q}^+_{i_0}$, nor $\mathbf{q}^-$ by well-bracketing. Such a P-view yields

$$\mathbf{q}^-_{i_0,j} s \in \Updownarrow\text{-}\mathsf{Plays}([\![A_1 \to \cdots \to A_n \to A_{i_0,j}]\!])$$

a P-view where moves in s formerly depending on $\mathbf{q}^-$ in $[\![A]\!]$ are set to depend on $\mathbf{q}^-_{i_0,j}$. Considering all such P-views generates a meager innocent strategy of size strictly lesser than σ, hence by induction hypothesis there is $\vdash M_{i_0,j} : A_1 \to \cdots \to A_n \to A_{i_0,j}$ s.t. $[\![M_{i_0,j}]\!] = \sigma_{i_0,j}$. Finally, with all this data we may form $\vdash M : A$ as

$$\begin{aligned}\lambda x_1^{A_1} \ldots x_n^{A_n}.\, \mathbf{case}\, x_{i_0}\, (M_{i_0,1}\, x_1\ \ldots\ x_n) \ldots (M_{i_0,p_{i_0}}\, x_1\ \ldots\ x_n)\, \mathbf{of}\\ v_1 \mapsto M_{v_1}\\ \cdots\\ v_p \mapsto M_{v_p}\end{aligned}$$

where p is such that every σ_{v_i} with $i > p$ is diverging. We get, as needed $[\![M]\!] = \sigma$. □

This final statement is conceptually straightforward (if one has access to the definition of the interpretation, which we save for the next chapter), but technically challenging as reasoning concretely on the categorical definition of the interpretation and on the (forthcoming) composition of strategies is unwieldy. In the author's opinion, there is still room for improvement in finding a conceptually clean, fully detailed and human-readable proof of this that makes it as easy technically as it is conceptually.

Here, **case** is the syntax introduced in Section 2.2, involving the **let** construct. Without that, simply iterating **if** constructs would yield a strategy that re-computes

$$x_{i_0}\, (M_{i_0,1}\, x_1\ \ldots\ x_n) \ldots (M_{i_0,p_{i_0}}\, x_1\ \ldots\ x_n)$$

each time it matches it against a value. This is what is done in [Hyland and Ong, 2000] as the historical presentation of PCF does not include a **let** construct. This yields a term that is not quite σ, but is nonetheless observationally equivalent.

We conclude this discussion on definability with the statement:

Conjecture 3.2.15 (Intensional Universality)**.** *Consider a* PCF *type A. For any* computable *well-bracketed innocent* $\sigma : [\![A]\!]$*, there is* $\vdash M : A$ *s.t.* $[\![M]\!] = \sigma$.

Hyland and Ong [Hyland and Ong, 2000] prove universality up to *contextual equivalence* (with a similar independent result in AJM games [Abramsky et al., 2000]). Their

construction fails to achieve intensional universality because their version of PCF lacks a "**let**" construct, making them unable to avoid repeated computation. There is strong confidence that their construction yields intensional universality in the presence of **let** (intensional universality does hold in call-by-value [Murawski and Tzevelekos, 2013]), but as far as we know the details have not been properly checked.

Intensional Full Abstraction. Recall from Section 2.3 that a model is *intensionally fully abstract* for a language when its observational quotient is fully abstract.

The famous *full abstraction for* PCF follows by a fairly typical argument, once one has an adequate model with sufficient definability properties.

Theorem 3.2.16 (Intensional Full Abstraction for PCF)**.** *Innocent well-bracketed strategies form an intensionally fully abstract model for* PCF.

Sketch. Consider $\vdash M, N : A$ observationally equivalent, and assume $[\![M]\!]$ and $[\![N]\!]$ are not, *i.e.* there is a test α which distinguishes them. One may easily prove (see [Hyland and Ong, 2000]) that it suffices to consider α finite. By Theorem 3.2.14, α is defined by a PCF term, which we present as a context $C[-]$. By Proposition 3.2.3, it then follows that $C[-]$ distinguishes M and N, contradiction. □

We have seen that by imposing well-bracketing and innocence, we have successfully banned all non PCF-definable behaviour. But dually, are references and control operators *complete*, are they sufficient to express unrestricted strategies?

3.3 Non-Innocence and Effects

Though Hyland-Ong games were originally designed as a model of plain PCF, what is in hindsight one of their most striking feature is that a crisp description of a few computational effects already hide in the definitions seen so far.

3.3.1 Non-Local State

As before, we start the discusion by looking at an example:

$$\begin{array}{c} \mathbb{B} \\ \mathbf{q}^- \\ \vdots \\ \mathbf{tt}^+ \\ \\ \mathbf{q}^- \\ \vdots \\ \mathbf{ff}^+ \end{array}$$

Clearly, this play is not realizable in PCF since it breaks innocence. But unlike the play of Section 3.2.3, it is not realizable either in IA, the extension of PCF with local state. Indeed, if this is to be realized by a deterministic program, some state must be

shared – or some information must flow – between the two calls. In contrast, a closed term with access to local state may initialize variables only once computation has begun, and thus seperate calls cannot enjoy any shared state.

Concretely, an hypothetical $\vdash \mathbf{bad} : \mathbb{B}$ acting as above would break β-equivalence:

$$(\lambda x.\, \mathbf{if}\ x\ x\ x)\, \mathbf{bad}$$

should evaluate to **ff**, as all calls are routed to **bad**. Yet, **if bad bad bad** should evaluate to **tt**: we now have three separate instances of **bad**, so only the first call matters. In this monograph we deem this pathological: we are concerned with call-by-name languages, which must hence enjoy the call-by-name equational theory, including β-equivalence.

We ban **bad** via the notion of *single-threadedness*: a single-threaded strategy only depends on the *current thread*, so that it cannot pass information between distinct instances. For the next definition, in a play $s \in \downarrow\uparrow\text{-}\mathsf{Plays}(A)$, we say that s_i is **hereditarily justified** by s_j (with $j < i$) if following the pointers from s_i eventually hits s_j. If s_j is initial, we say that s_j is the (necessarily unique) **initial hereditary justifier** of s_i.

Definition 3.3.1. *Consider A an arena, and $s \in \downarrow\uparrow\text{-}\mathsf{Plays}(A)$.*

*The **current thread** of non-empty s, written $\lceil s \rceil$, is the subsequence of s comprising all moves with the same initial hereditary justifier as the last move of s; and $\lceil \varepsilon \rceil = \varepsilon$.*

The definition of single-threaded strategies then mimics that of innocent strategies:

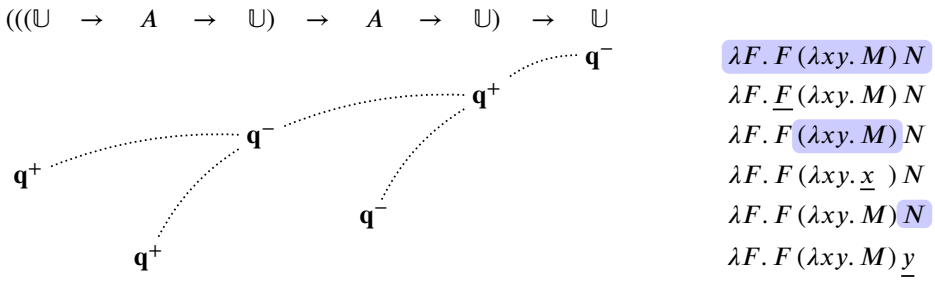
Definition 3.3.2. *Consider $\sigma : A$ any strategy. It is **single-threaded** if it satisfies:*

well-threaded: *for all $sa^+ \sqsubseteq t \in \sigma$, the justifier of a appears in $\lceil s \rceil$,*
single-threaded: *for all $sa^+, t \in \sigma$, if $\lceil s \rceil = \lceil t \rceil$ then $ta^+ \in \sigma$ with $\lceil sa \rceil = \lceil ta \rceil$.*

This bans pathological strategies such as **bad**. On the other hand, single-threadedness is preserved under all operations involved in the interpretation of IA so that the interpretation of any term of IA yields a single-threaded strategy.

3.3.2 Higher-Order State

Finally, for our last example, consider the following play:



As shown in the indicative term on the right hand side, and following the intuitions on the operational content of game semantics so far, the last Player move corresponds to calling variable y. But how can that make sense, since y is not even in the scope!

Yet, again, this play is computationally meaningful – it witnesses *higher-order state*:

$$\lambda F.\, \mathbf{newref}_A\, r\, \mathbf{in}\, F\, (\lambda x y.\, r := y;\; x)\, (!r)$$

Though we borrow the notation from IA, this must be read completely differently. Here A can be an arbitrary type, *i.e.* potentially a function. Whereas in IA, references store *values*, here they store *computations*; so that when the reference is read, the control flow *jumps* accordingly, giving control to arbitrary program phrases.

Higher-order state is a sensible programming feature, found in the wild in call-by-value languages such as ML. In call-by-value game semantics, a games model with morphisms set to comprise all well-bracketed single-threaded strategies is fully abstract model for a version of ML with higher-order store [Abramsky et al., 1998]. Semantic investigations have also been conducted in call-by-name [Laird, 2002, Goyet, 2013], but this is somewhat less natural from a programming language point of view. Either way, we leave higher-order state out of the scope of this monograph.

So how should one get rid of (the behaviours generated by) higher-order state? Without higher-order state, only information can flow through shared state, not control. So Player can only pass control (*i.e.* call variables) within the *scope*, *i.e.* the current syntactic branches. The same sentence, translated in game semantics terminology along the dictionary elaborated so far, reads: *Player can only point within the P-view*. But we have already seen this: this is *P-visibility* (Definition 3.2.8). So P-visibility, originally imposed as a preliminary construction just to make P-views well-defined, is actually also the condition we need to allow ground state but ban higher-order state!

P-visibility is preserved by all constructions used in the interpretation of IA (including ground type references), while the play above clearly is not P-visible.

3.3.3 The "Semantic Cube"

It is striking that almost, if not all, strategies in Hyland-Ong games correspond to existing programming features, for a framework originally designed for plain PCF only! From now on we assume that all strategies are single-threaded and P-visible, the canvas for the celebrated "semantic cube" that we shall present now.

Idealized Algol. Well-bracketed strategies support an adequate interpretation of IA:

Proposition 3.3.3. *There is an interpretation $[\![-]\!]$ sending types of* IA *to arenas, and terms $\vdash M : A$ to well-bracketed, single-threaded, P-visible $[\![M]\!] : A$.*
Moreover, for all $\vdash M : \cup$, $M \Downarrow$ iff in $[\![M]\!]$, the initial move has an answer.

The interpretation of IA will be detailed in Section 4.2.

It turns out that well-bracketed strategies support "only" the interpretation of IA: we have a finite definability property similar to Theorem 3.2.14. We state:

Theorem 3.3.4. *Consider A an* IA *type.*
*For any $\sigma : [\![A]\!]$ visible, well-bracketed, single-threaded, and finite (*i.e. *with finitely many even-length plays) there is $\vdash M : A$ a term of* IA *such that $[\![M]\!] = \sigma$.*

We shall not give the detailed proof of this as their statement relies on the compositional structure, but the key arguments appear in Section 4.3.2.

As for PCF, from those definability properties follow:

Theorem 3.3.5 (Intensional Full Abstraction for IA)**.** *P-visible well-bracketed single-threaded strategies form an intensionally fully abstract model for* IA.

The reasoning for that is the same as for Theorem 3.2.16.

Recall that intensional full abstraction means that the observational quotient of the model is fully abstract; in other words, visible well-bracketed single-threaded strategies have the same distinguishing power as terms of IA. As for PCF this makes the quotiented fully abstract model hard to reason with, but for IA something much more interesting happens: we can characterize concretely observational equivalence.

Complete plays. This is achievable by understanding which parts of a P-visible, well-bracketed strategy can be explored by a P-visible, well-bracketed Opponent – and among those, which lead to an observable result at toplevel. This is done via:

Definition 3.3.6. *Consider A an arena. A play $s \in \Downarrow\!\uparrow\text{-}\widehat{\mathsf{Plays}}(A)$ is* ***complete*** *iff it is well-bracketed, O-visible, P-visible, and every question has an answer.*

If $\sigma : A$, we write $\mathsf{comp}(\sigma)$ for the set of its complete plays. Complete plays indeed characterize observational equivalence for P-visible well-bracketed strategies:

Proposition 3.3.7. *Consider $\sigma, \tau : A$ two P-visible, well-bracketed strategies. Then,*

$$\sigma \simeq \tau \qquad \Leftrightarrow \qquad \mathsf{comp}(\sigma) = \mathsf{comp}(\tau)\,.$$

Again we omit the (rather simple) details which rely on the compositional structure introduced in the next chapter only, but the idea is the following. If a strategy $\alpha : A \Rightarrow \mathbb{U}$ distinguishes σ and τ, then it is straightforward to read from that a complete play in one but not the other. Reciprocally, if s is a complete play in σ but not in τ, then $\mathbf{q}^- s \checkmark^+$ (with adequate pointers) is a play in $A \Rightarrow \mathbb{U}$. The set of all prefixes of $\mathbf{q}^- s \checkmark^+$, closed under receptivity, is then a valid strategy that distinguishes σ and τ.

From Proposition 3.3.7 and Theorem 3.3.5, it shall be clear that we get:

Theorem 3.3.8. *Consider A a type, and $\vdash M, N : A$ two terms of* IA*. Then,*

$$M \simeq N \qquad \Leftrightarrow \qquad \mathsf{comp}([\![M]\!]) = \mathsf{comp}([\![N]\!])\,.$$

This entails that the fully abstract model is *effectively presentable*: it can be phrased as a computable function into an effective domain [Plotkin, 1981]; concretely this means that observational equality is decidable between finite strategies (those with finitely

many even-length plays; forming the compact elements of the domain of strategies). This is remarkable: in the theory of programming languages, situations like that where one has a concrete characterization of observational equivalence, are rare!

This comes from the ability to write a context $C[-]$ reproducing one complete play s and *only* s; so that $C[M] \Downarrow$ is exactly equivalent to $s \in [\![M]\!]$. This is impossible for an innocent context, because if an innocent strategy includes one play, it also includes all plays with the same P-views. In fact, we know that the observational quotient in the model of PCF cannot be characterized concretely, because of Loader's result that the (unique) fully abstract model of PCF is not effectively presentable [Loader, 2001].

Full abstraction with control. We now drop well-bracketing. We also consider temporarily the extensions CIA and CPCF respectively of IA and PCF with

$$\mathbf{callcc}_{A,B} : ((A \to B) \to A) \to A\,,$$

a *control operator* with behaviour described informally in Section 3.2.2. We shall not cover its behaviour or interpretation in detail as it plays no role in this monograph beyond painting the big picture. But we do mention two main results, first for CIA:

Theorem 3.3.9. *Visible single-threaded strategies form a fully abstract model for* CIA. *Moreover, if* $\vdash M, N : A$ *are two terms of* CIA*, then*

$$M \simeq N \qquad \Leftrightarrow \qquad \mathsf{O\text{-}vis}([\![M]\!]) = \mathsf{O\text{-}vis}([\![N]\!])\,,$$

with $\mathsf{O\text{-}vis}(\sigma)$ *the set of O-visible plays of a strategy* σ.

This is proved again via an adequate definability result. This theorem was surely known long before it was published[5] in [Murawski, 2007]. Here again, the fully abstract model is also effectively presentable. Note that contexts from CIA can observe more than contexts from IA: **callcc** lets us observe plays that cannot be completed, and hence would never have led to an observable behaviour within IA.

Finally, we mention one last fully abstract model:

Theorem 3.3.10. *Visible innocent strategies form a fully abstract model for* CPCF.

This is due to [Laird, 1997], proved again via an adequate definability property. In this case also, the fully abstract model is effectively presentable, but not as concretely as above: instead, one shows that equality between compact elements is decidable, by reducing equivalence checking to a finite number of tests. The fully abstract model can also be described concretely – and in fact *was* described concretely, before even the fully abstract games model for PCF were developed [Cartwright et al., 1994]!

The "semantic cube". We have, so far, seen the following full abstraction results:

[5] Murawski uses a different primitive for control, but the difference is superficial within IA.

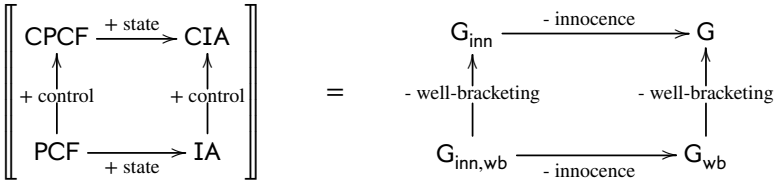


Figure 3.13: The Semantic Cube

Theorem 3.3.11 (Semantic Cube)**.** *We have four intensional full abstraction results:*

G	*is fully abstract for*	PCF + state + control
G + innocence	*is fully abstract for*	PCF + control
G + well-bracketing	*is fully abstract for*	PCF + state
G + innocence + well-bracketing	*is fully abstract for*	PCF

where G *denotes the model formed of visible, single-threaded strategies.*

This "Semantic Cube", drawn in Figure 14.2, expresses that the conditions on strategies capture the behaviour generated by certain computational effects; or rather the *absence* of certain effects. The achievement is noteworthy, as it is famously difficult to *combine* semantic accounts of effects. But independently of purely semantic purposes, this gives us a microscope to study behaviourally interactions between effects in programming languages. We demonstrate this with the following orthogonality property[6] between state and control which nicely illustrates the strength of game semantics. We only state it as a conjecture, because it relies on intensional universality for PCF.

Conjecture 3.3.12. *Let* $\vdash M : A$ *a term of* CIA *with* A *a* PCF *type. Assume that*

(1) $M \simeq N_1$ *where* N_1 *does not use* **callcc***,*
(2) $M \simeq N_2$ *where* N_2 *does not use* **newref***,*

then there is $\vdash N : A$ *in pure* PCF *such that* $M \simeq N$.

Proof. We have $N_1 \simeq N_2$. By Theorem 3.3.9, they have the same O-visible plays. The P-views of $[\![N_2]\!]$ are O-visible, so they must be in $[\![N_1]\!]$, so they must be well-bracketed. But $[\![N_2]\!]$ is innocent, so if its P-views are well-bracketed, then it must be well-bracketed (Proposition 3.2.13). Summing up, $[\![N_2]\!]$ is an innocent well-bracketed strategy. But it is also computable, since the interpretation is computable. Hence, by Conjecture 3.2.15, there is $\vdash N : A$ in pure PCF such that $[\![N]\!] = [\![N_2]\!]$. In particular, it has the same O-visible plays as M, hence $M \simeq N$ by Theorem 3.3.9 again. □

We finish this section with two concluding notes.

Firstly, game semantics has the reputation that its full abstraction results are unsatisfactory, because they consist in constraining strategies until they are essentially syntax,

[6] We learnt of it from a talk by Paul Levy in 2014 [Levy, 2014].

followed by the same quotient that one could have done already in the syntax. This criticism holds for PCF, and even then seems short-sighted: without the fully abstract model for PCF, we would not have the semantic cube. But furthermore, beyond PCF, *all fully abstract models in the semantic cube can be effectively presented!*

Secondly, unfortunately the semantic cube holds only for sequential deterministic languages. Beyond that we still have models (we shall see one in Chapter 5), but they do not fit in the same unified landscape as the cube above. What can be done about that? This is one of the leading questions behind this monograph.

3.4 An Alternative to Pointers: Copy Indices

As the final section of this chapter, we briefly present an alternative to the use of pointers for thread indexing: *copy indices*, used in particular in the other major family of traditional game semantics models, called AJM games for Abramsky, Jagadeesan and Malacaria [Abramsky et al., 2000]. We do this, on the one hand, to account for the large part of the game semantics literature that depends on it; and on the other hand, to help prepare the reader for concurrent games, which use copy indices.

The idea of copy indices is that rather than explicitly authorizing repetitions (and adding pointers to disambiguate the relationships between copies), we keep the invariant that each move must appear *at most once* in a play. So as to allow players to perform duplications, we create explicit copies of duplicable moves, which are kept apart by adjoining to each a unique identifier, a natural number called a *copy index*.

Copy indices must be added by an explicit construction on games. To that end, games with copy indices typically follow linear logic [Girard, 1987] and decompose the arrow construction in two steps $A \Rightarrow B = !A \multimap B$. Here, $\multimap$ is a *linear* or *affine* arrow that does not allow replication of the argument; and $!A$ is the *bang* construction from linear logic, which creates countably many copies of a game by adjoins copy indices:

Definition 3.4.1. *Consider A a simple game. Then, the simple game $!A$ has:*

$$\begin{array}{rcl} |!A| & = & \mathbb{N} \times |A| \\ \mathrm{pol}_{!A}(i,a) & = & \mathrm{pol}_A(a) \\ \mathrm{P}_{!A} & = & \{s \in |!A|^* \mid s \text{ alternating}, \forall i \in \mathbb{N},\ s \upharpoonright i \in \mathrm{P}_A\} \end{array}$$

where $s \upharpoonright i \in |A|^*$ is the obvious operation only keeping moves of the form (i, a).

From here and throughout this monograph, we adopt the convention that copy indices are typeset in grey: this helps in easily distinguishing them from other indices, and is in line with the intuition that though their presence is needed to give copies an identity, their precise value is in general irrelevant. Now we have, for instance:

$$\mathrm{P}^I_{!\mathbb{B}} ::= \varepsilon \mid \mathbf{q}^-_i \mid \mathbf{q}^-_i \mathbf{tt}^+_i \mathrm{P}^{I \uplus \{i\}}_{!\mathbb{B}} \mid \mathbf{q}^-_i \mathbf{ff}^+_i \mathrm{P}^{I \uplus \{i\}}_{!\mathbb{B}} \qquad i \notin I$$

and $\mathrm{P}_{!\mathbb{B}} = \mathrm{P}^{\emptyset}_{!\mathbb{B}}$; so plays in $!\mathbb{B}$ is any sequence of plays in $\mathbb{B}$, each with a copy index, in any order but such that each copy index is used at most once.

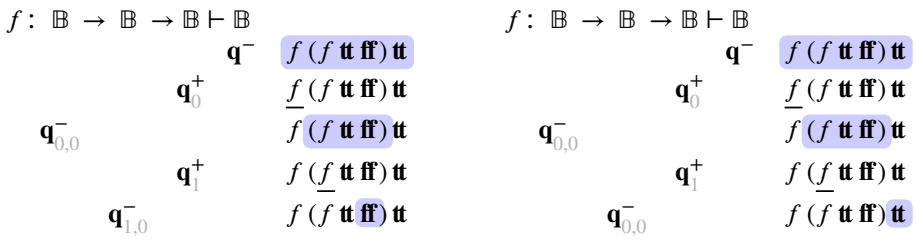


Figure 3.14: Two plays with copy indices disambiguating Figure 3.3.

As mentioned above, the usual call-by-name arrow is defined in linear logic by "Girard's translation". Accordingly, the interpretation of types include an occurrence of ! each time we cross the left of an arrow. Now, we saw that each move corresponds to a precise occurrence of a ground type in the type/typing judgment; so in the scope of a fixed number of !. Accordingly, each move carries a *sequence* of copy indices, once for each ambiant ! – by convention, we write first the copy index for the outermost !.

With this convention, the two plays corresponding to Figure 3.5 may be drawn[7] with copy indices in Figure 3.14. This shows how copy indices carry implicitely the hierarchical relationship between moves expressed by pointers. Yet, plays are still straightforward sequences of moves, without unwieldy additional structure.

But if the information expressed by pointers can be recovered on mere plays with such a small technical overhead, why bother with pointers at all? In fact, we have yet to see the main subtlety – and technical hurdle – arising from copy indices.

Equality up to copy indices. The issue is that there are now many more strategies, as when playing a positive move, Player may seemingly arbitrarily choose a copy index – which does not seem to carry any relevant computational information.

In fact, this has the concrete consequence that simple games with replication as in Definition 3.4.1 fail to be a model of the simply-typed λ-calculus. Indeed we have

$$x : \mathbb{B} \vdash (\lambda f.\, f\, x\, x)\,(\lambda yz.\, y) \quad \cong_\beta \quad (\lambda f.\, f\, x\, x)\,(\lambda yz.\, z) : \mathbb{B} \tag{3.2}$$

yet, following the usual definition of the interpretation (which we shall not present right now), the left hand side calls x with copy index 0 while the right hand side calls x with copy index 1. Copy indices trace back which occurrence of x was selected in $\lambda f.\, f\, x\, x$, but remembering this is incompatible with β-equivalence.

To address this, AJM games add to simple games an equivalence relation on plays:

[7] Again, the play on the right hand side is not accepted by the usual definitions of simple games or AJM games as it is not *locally alternating*. But it is nevertheless illustrative of the real phenomenon.

Definition 3.4.2. *An **AJM game** comprises a simple game $A = (|A|, \mathrm{pol}_A, \mathrm{P}_A)$, together with an equivalence relation $\cong_A$ on P_A satisfying the following conditions:*

(1) *if $s_1 \dots s_n \cong_A t_1 \dots t_p$, then $n = p$ and for all $1 \leq i \leq n$, $\mathrm{pol}_A(s_i) = \mathrm{pol}_A(t_i)$,*
(2) *if $s_1 \dots s_n \cong_A t_1 \dots t_n$, then for all $1 \leq i \leq n$, $s_1 \dots s_i \cong_A t_1 \dots t_i$,*
(3) *if $s \cong_A t$ and $sa \in \mathrm{P}_A$, then there is $tb \in \mathrm{P}_A$ such that $sa \cong_A tb$.*

We think of the equivalence relation $\cong_A$ as formalizing *reindexings*, *i.e.* we have $s \cong_A t$ iff s and t are the same play, but with a different choice of copy indices. For ground types such as $\mathbb{B}$ the equivalence relation is restricted to reflexive pairs, but:

Definition 3.4.3. *Consider A an AJM game. We set $!A$ following Definition 3.4.1, plus:*

$$s_1 \dots s_n \cong_{!A} t_1 \dots t_n \quad \Leftrightarrow \quad \text{there is a permutation } \pi : \mathbb{N} \simeq \mathbb{N} \text{ such that } \begin{cases} \text{for all } 1 \leq i \leq n, \text{ if } s_i = (k, a) \text{ then } t_i = (\pi(k), a'), \\ \text{and for all } k \in \mathbb{N},\ s \upharpoonright k \cong_A t \upharpoonright \pi(k). \end{cases}$$

In other words, an equivalence $s \cong_{!A} t$ follows a global reindexing on copy indices introduced by $!$, and local reindexings on each individual copy.

These reindexings on plays induce a partial equivalence relation on strategies:

Definition 3.4.4. *Consider A an AJM game.*
For $\sigma, \tau : A$ strategies on the underlying simple game, we write $\sigma \approx \tau$ iff:

$$\begin{array}{rl} \rightarrow\text{-simulation:} & \forall sa^+ \in \sigma,\ t \in \tau,\ s \cong_A t \ \Rightarrow\ \exists b^+,\ tb^+ \in \tau \ \&\ sa^+ \cong_A tb^+ \\ \leftarrow\text{-simulation:} & \forall s \in \sigma,\ tb^+ \in \tau,\ s \cong_A t \ \Rightarrow\ \exists a^+,\ sa^+ \in \sigma \ \&\ sa^+ \cong_A tb^+ \\ \rightarrow\text{-receptive:} & \forall sa^- \in \sigma,\ t \in \tau,\ sa^- \cong_A tb^- \ \Rightarrow\ tb^- \in \tau \\ \leftarrow\text{-receptive:} & \forall s \in \sigma,\ tb^- \in \tau,\ sa^- \cong_A tb^- \ \Rightarrow\ sa^- \in \sigma \end{array}$$

*We say that $\sigma : A$ is **uniform** iff $\sigma \approx \sigma$.*

This allows us to consider equal strategies behaving the same up to their choice of copy indices – for instance, it identifies the two strategies corresponding to the terms of (3.2). But it does more: it select the *uniform* strategies, which are unable to observe the copy indices chosen by Opponent; *i.e.* with respect to which $\approx$ is a congruence.

When setting up a game semantics based on copy indices, the idea is that one only considers strategies which are uniform, and those are considered up to $\approx$.

Pointers or Copy Indices? In designing a game semantics supporting replication, one must use one or the other of these two structures. Which one should be chosen?

Beyond thread indexing, pointers also import into plays relevant semantical structures. In Hyland-Ong games, pointers are key to defining *P-visibility* and *innocence* [Hyland and Ong, 2000], the conditions that eventually gave rise to the semantic cube. On the other hand, pointers are hard to manipulate in a rigorous yet unobtrusive way.

In contrast, the AJM model has simpler plays, and enjoys a clean linear decomposition in the sense of Linear Logic [Girard, 1987]. However, terms are interpreted as

$\approx$-equivalence classes rather than concrete strategies and all constructions come with a sometimes challenging proof obligation that it preserves $\approx$. Conditions which rely on pointers such as visibility and innocence are beyond reach without further structure.

Much of the literature in game semantics for programming languages has been developed based on pointers and Hyland-Ong games. However, in this monograph, we shall see that *concurrent games* are based on copy indices. This is for two reasons: firstly, their finer intensional structure seems to require it. Secondly, despite using copy indices and unlike in AJM games, the *pointers* of Hyland-Ong games which have proved so fruitful in studying program behaviour are retained as derived structures.

3.5 Conclusions and Historical Notes

Much of the basic definitions presented in this chapter originate (though with minor variations) in Hyland and Ong's original paper [Hyland and Ong, 2000], including the definitions of *visibility*, of *innocence* and of *well-bracketing* (on plays). The realization that lifting these constraints yielded fully abstract models for richer languages came very quickly: full abstraction for IA is due to [Abramsky and McCusker, 1996], full abstraction for CPCF is due to Laird in [Laird, 1997]. To our knowledge, the full abstraction result for the missing corner of the cube (CIA) was only developed in detail significantly later by Murawski [Murawski, 2007] – though it was certainly well-known.

It might seem confusing that apart from this latter paper, all the works cited here, which build on Hyland and Ong's framework, date from *before* the publication of the original Hyland and Ong paper. In fact, though the Hyland and Ong paper was published only in 2000, preliminary versions of it circulated from the earlier 90s[8], in parallel with versions of the AJM model. These developments, and announcements by the two competing groups, were closely followed by the broader denotational semantics community, feeling that the resolution of the problem of "full abstraction for PCF" was close.

In hindsight, rather than the full abstraction result for PCF *per se*, what had the most impact was definitely the nice surprised that many different effects could be captured in a unified framework; justifying the 2017 *Alonzo Church Award* awarded to the founders of game semantics. Game semantics has since then grown to a broad topic; and this monograph shall cover some of its most recent developments.

[8]The workshop celebrating the 25 years of game semantics was held in 2018 as part of the federated logic conference (FLoC), which places the birth of game semantics – somewhat arbitrarily – in 1993.

Chapter 4

The Category of Alternating Strategies

In this chapter, we continue our introduction to sequential game semantics, filling the gap left open in the previous chapter: the compositional structure. We show how to compose alternating strategies, forming a cartesian closed category $\downarrow\uparrow$-**Strat**. Then, we define the interpretation of IA into $\downarrow\uparrow$-**Strat**, and conclude with some complements.

4.1 The Ambient Cartesian Closed Category $\downarrow\uparrow$-Strat

We organize arenas and P-visible strategies into a cartesian closed category $\downarrow\uparrow$-**Strat**. Its objects will be arenas, already presented in Section 3.1.3. But the cartesian structure requires more arena constructions beyond the well-opened case shown in Chapter 3.

4.1.1 More Arena Constructions

We start with the *product arena*, used for interpreting contexts:

Definition 4.1.1. *Consider A_1, A_2 arenas. Their **product** $A_1 \& A_2$ has components:*

$$\begin{array}{rrcl}
\text{moves:} & |A_1 \& A_2| & = & |A_1| + |A_2| \\
\text{causality:} & (i,a) \leq_{A_1 \& A_2} (j,a') & \Leftrightarrow & i = j \;\&\; a \leq_{A_i} a' \\
\text{polarities:} & \mathrm{pol}_{A_1 \& A_2}(i,a) & = & \mathrm{pol}_{A_i}(a)\,, \\
\mathcal{Q}/\mathcal{A}\text{-labeling:} & \lambda_{A_1 \& A_2}(i,a) & = & \lambda_{A_i}(a)\,,
\end{array}$$

generalizing to an n-ary construction $\&_{i \in I} A_i$ in the obvious way (for I a finite set).

This construction puts arenas A_1 and A_2 side-by-side, without interaction – Opponent may start evaluation in either component. Like a forest is formed of trees, any arena is – up to isomorphism – a product of well-opened arenas:

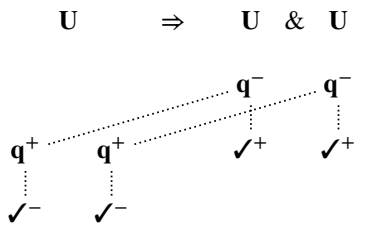


Figure 4.1: The arena $\mathbf{U} \Rightarrow (\mathbf{U} \& \mathbf{U})$

Lemma 4.1.2. *Consider A an arena.*
Then there is $(A_i)_{i \in I}$ of well-opened arenas, unique up to iso, s.t. $A \cong \&_{i \in I} A_i$.

Here, an **isomorphism** of arenas is any forest-isomorphism preserving polarities and $\mathcal{Q}/\mathcal{A}$-labeling – likewise, two families $(A_i)_{i \in I}$ and $(B_j)_{j \in J}$ are **isomorphic** if there is a bijection $\pi : I \simeq J$ and for each $i \in I$, an isomorphism $A_i \cong B_{\pi(i)}$.

Recall that in Definition 3.2.2, we defined the **arrow** $A \Rightarrow B$ of two arenas A and B with B well-opened. Building a cartesian closed category requires us to extend $A \Rightarrow B$ to the case where B may not be well-opened. However, $A \Rightarrow -$ shall be the right adjoint to $A \times -$, and so must preserve products; so we know that we must have:

$$A \Rightarrow (B \times C) \qquad \cong \qquad (A \Rightarrow B) \times (A \Rightarrow C)\,,$$

and via Lemma 4.1.2 we may take that as a definition:

Definition 4.1.3. *For A, $B = \&_{i \in I} B_i$ any two arenas, we set:*

$$A \Rightarrow B \;=\; \bigamp_{i \in I} A \Rightarrow B_i\,.$$

This has the effect of creating one copy of A for each minimal move in B[1] – in particular, if B is the empty arena, then so is $A \Rightarrow B$. In Figure 4.1 we show an example illustrating the arrow construction on non well-opened arenas.

4.1.2 Composition of Strategies

Restrictions. We first require notations for *restrictions* of plays of an arena to a subcomponent. We will actually define the restriction of a play along an *embedding*:

Definition 4.1.4. *Consider A, B arenas.*
*An **embedding** $f : A \hookrightarrow B$ is any injection $f : |A| \to |B|$.*

[1] This definition is equivalent to the original one [Hyland and Ong, 2000]. One may alternatively avoid the duplication of A by considering arenas to be directed acyclic graphs rather than forests, and having minimal moves of A depend on *all* the minimal moves in B: this is done for instance in [Harmer, 2004]. This is fine in the sense that the two constructions generate the same set of plays (see Definition 3.1.4), but arenas then no longer characterize types up to isomorphism, and the connection with thin concurrent games is blurred out.

We use embeddings to address specific components inside arenas: for instance, there is a canonical embedding $A \hookrightarrow A \times B$, etc. The embeddings we use in practice satisfy many additional properties, but it is not useful to require them explicitely.

We can *restrict* a play along an embedding:

Definition 4.1.5. *Consider A, B arenas, $f : A \hookrightarrow B$, and s a pointing sequence on B.*
*The **restriction of** s **along** f, denoted by $s \upharpoonright f$, has moves the sequence defined by*

$$\begin{array}{rcll} \varepsilon \upharpoonright f & = & \varepsilon & \\ sb \upharpoonright f & = & (s \upharpoonright f)a & \text{if } b = f(a), \\ sb \upharpoonright f & = & s \upharpoonright f & \text{otherwise.} \end{array}$$

Moreover, in $s \upharpoonright f$, s_i points to s_j iff there is a sequence of pointers

$$\ldots \quad s_j \overset{\cdots}{\cdot} \ldots \overset{\cdots}{\cdot} \ldots \cdot \ldots \overset{\cdots}{\cdot} \ldots \overset{\cdots}{\cdot} s_i$$

in s, with all intermediary points not selected in $s \upharpoonright j$.

The restriction $s \upharpoonright f$ is a pointing sequence on A, but there is no guarantee that it is a legal play on A, even if $s \in \downarrow\uparrow\text{-}\widehat{\mathsf{Plays}}(B)$. However, that will often be the case when using this notion. This overly general definition has two use cases in practice:

Firstly, situations like restrictions along the canonical embedding $A \hookrightarrow A \times B$, where the transitive clause for pointers is not used at all. For $s \in \downarrow\uparrow\text{-}\widehat{\mathsf{Plays}}(A \times B)$, we write $s \upharpoonright A \in \downarrow\uparrow\text{-}\widehat{\mathsf{Plays}}(A)$ for the restriction along the (implicit) canonical embedding, which the reader should always be able to disambiguate easily. Secondly, situations like restrictions along the canonical embedding $A \Rightarrow C \hookrightarrow (A \Rightarrow B) \Rightarrow C$. In that case we write $s \upharpoonright A, C$, which may not always be a valid play on $A \Rightarrow C$ (it may not be alternating). As here, in general we never explicitely specify the embedding used in the restriction, but it should hopefully always be clear from the context.

Interactions. Consider $\sigma : A \Rightarrow B$ and $\tau : B \Rightarrow C$ two strategies. Observe that σ and τ have dual perspectives on the polarities in B: when σ plays a Player move in B, it is an Opponent move for τ, and reciprocally. The first step in defining composition, is to *plug* σ and τ, letting them synchronize on B: this is called the *parallel interaction* of σ and τ. Then we *hide* the synchronized events, leaving a strategy on $A \Rightarrow C$.

We first focus on *parallel interaction*. An *interaction* between σ and τ, resulting of the synchronization of plays of σ and τ, is a pointing sequence that spans A, B and C:

Definition 4.1.6. *Consider A, B and C arenas.*
*An **interaction** on A, B, C is a legal pointing sequence u on $(A \Rightarrow B) \Rightarrow C$ s.t.:*

$$\begin{array}{rl} \text{left-legal:} & u \upharpoonright A, B \in \downarrow\uparrow\text{-}\widehat{\mathsf{Plays}}(A \Rightarrow B), \\ \text{right-legal:} & u \upharpoonright B, C \in \downarrow\uparrow\text{-}\widehat{\mathsf{Plays}}(B \Rightarrow C), \\ \text{outer-legal:} & u \upharpoonright A, C \in \downarrow\uparrow\text{-}\widehat{\mathsf{Plays}}(A \Rightarrow C). \end{array}$$

We write $\mathrm{I}(A, B, C)$ the set of all interactions on A, B, C.

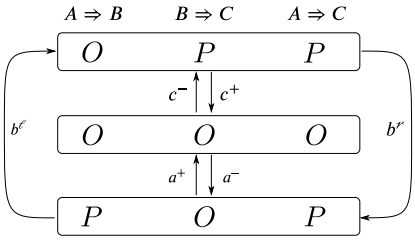


Figure 4.2: State diagram for sequential interactions

In interactions, moves in A, C are said to be **visible** while moves in B are **hidden** or **synchronized**. One may regard an interaction $u \in \mathtt{I}(A, B, C)$ as involving three players: player ℓ plays in A, B; player r plays on B, C. Finally, the **external Opponent** plays Opponent on A, C, *i.e.* plays moves that have negative polarity on $A \Rightarrow C$. Positive moves in $A \Rightarrow C$ may be due to ℓ or r; we say that they are due to the **global Player**, which we think of as ℓ and r working together as a team.

The conditions *outer-legal*, *left-legal* and *right-legal* have a strong impact on how interactions may unfold. Indeed, recall from below Definition 3.1.4 that each alternating play has a *state*: O if it is Opponent's turn to play (if the play has even length), and P otherwise. The three conditions of Definition 4.1.6 involve three restrictions

$$u \upharpoonright A, B \in \downarrow\uparrow\text{-}\widehat{\mathsf{Plays}}(A \Rightarrow B),$$
$$u \upharpoonright B, C \in \downarrow\uparrow\text{-}\widehat{\mathsf{Plays}}(B \Rightarrow C),$$
$$u \upharpoonright A, C \in \downarrow\uparrow\text{-}\widehat{\mathsf{Plays}}(A \Rightarrow C)$$

for $u \upharpoonright A, C \in \downarrow\uparrow\text{-}\widehat{\mathsf{Plays}}(A \Rightarrow C)$. Accordingly, where the polarity state of a play is summarized by a single letter O or P, the polarity state of an interaction is summed up by three letters, keeping track of the state of these three restrictions, in order. There are strong constraints as to how we can jump between these states: for instance, in state OOO no move can be played in B, as no polarity in B can simultaneously satisfy the two constraints that the next move should be an Opponent move in both $A \Rightarrow B$ and $B \Rightarrow C$. Overall, it is fairly easy to show that only three states are reachable from the empty interaction: OOO, OPP, and POP; and the possible transitions between those are cleanly summarized in the **state diagram for interactions** in Figure 4.2.

We are often interested by interactions ending with a visible move from the global Player (which yield even-length plays of the composition). Those always have the shape

$$e^-(b \dots b)e^+ \dots e^-(b \dots b)e^+ ,$$

i.e. sequences of synchronized moves in b sandwiched by a pair of visible moves from the external Opponent, and global Player – this immediately follows from the diagram.

Composition. It is now time to define the composition of two strategies:

Definition 4.1.7. *Consider $\sigma : A \Rightarrow B$ and $\tau : B \Rightarrow C$ two strategies.*
*An **interaction** between σ and τ is any $u \in \mathtt{I}(A, B, C)$ such that $u \upharpoonright A, B \in \sigma$ and*

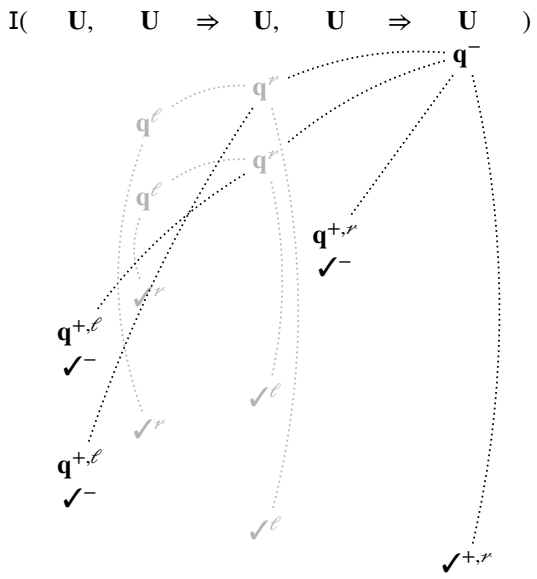


Figure 4.3: $u \in [\![x : \mathbb{U} \vdash \lambda y.\, y;\ x : \mathbb{U} \to \mathbb{U}]\!] \parallel [\![f : \mathbb{U} \to \mathbb{U} \vdash \lambda x.\, f\ (f\ x) : \mathbb{U} \to \mathbb{U}]\!]$.

$u \upharpoonright A, C \in \tau$. *We write* $\sigma \parallel \tau$ *for the set of all interactions between* σ *and* τ.
We may then define the ***composition*** $\tau \odot \sigma = \{u \upharpoonright A, C \mid u \in \sigma \parallel \tau\} : A \Rightarrow C$.

The process that to an interaction $u \in \sigma \parallel \tau$ associates $u \upharpoonright A, C \in \tau \odot \sigma$ is called **hiding**, hence completing the *parallel interaction plus hiding* paradigm. If $s \in \tau \odot \sigma$, we call any $u \in \sigma \parallel \tau$ such that $s = u \upharpoonright A, C$ a **witness** for s.

We give in Figure 4.3 an example interaction between two innocent strategies. In the diagram, moves are annotated with the player responsible: $-$ if it is the external Opponent, $+$ if it is the global Player, and then ℓ or r. This interaction is a witness to the play of the composition keeping the moves in black only.

When composing alternating strategies, the following property is crucial:

Lemma 4.1.8 (Unique witness)**.** *Consider* $\sigma : A \Rightarrow B$ *and* $\tau : B \Rightarrow C$ *strategies.*
For all $s \in \tau \odot \sigma$*, there is a unique* $u \in \sigma \parallel \tau$ *such that* $u \upharpoonright A, C = s$.

Proof. Direct via the state diagram for interactions and determinism of σ and τ. □

The next proposition states that composition is well-defined; and that all the conditions on strategies introduced in the previous chapter are stable under composition.

Proposition 4.1.9. *Take* $\sigma : A \Rightarrow B, \tau : B \Rightarrow C$ *strategies as in Definition 3.1.5.*

Then, $\tau \odot \sigma : A \Rightarrow C$ is a strategy in the sense of Definition 3.1.5. Moreover,

if σ and τ are P-visible, then so is $\tau \odot \sigma$,
if σ and τ are well-bracketed, then so is $\tau \odot \sigma$,
if σ and τ are single-threaded, then so is $\tau \odot \sigma$,
if σ and τ are innocent, then so is $\tau \odot \sigma$.

Proof. For Definition 3.1.5, *non-empty* and *prefix-closed* are clear, while *receptive* and *determinism* are direct via Lemma 4.1.8 and the state diagram of interactions.

For some of the further conditions, the proof of stability under composition can be fairly elaborate – we omit them, but give references. Composition of P-visibility is proved *e.g.* in [Clairambault, 2010, Proposition 2.2.6]; composition of well-bracketing is proved *e.g.* in [Clairambault, 2010, Proposition 2.3.3]; for single-threadedness, it follows from Harmer's theorem [Harmer, 1999, Proposition 3.5.5] that those are exactly the comonoid morphisms; and finally, a nice detailed proof of stability of innocence under composition appears in [Harmer, 2004, Proposition 3.3.2]. □

4.1.3 A Category of Arenas and Strategies

We now provide the categorical structure. Though we shall not give details for any of the required proofs, we shall still review the main arguments and provide references.

Associativity. For associativity, one first introduces the set $\mathtt{I}(A, B, C, D)$ of **ternary interactions**: those legal pointing sequences u on $((A \Rightarrow B) \Rightarrow C) \Rightarrow D$ such that

$$\begin{array}{rcl} u \upharpoonright A, B & \in & \downarrow\uparrow\text{-}\mathsf{Pla\tilde{y}s}(A \Rightarrow B)\,, \\ u \upharpoonright B, C & \in & \downarrow\uparrow\text{-}\mathsf{Pla\tilde{y}s}(B \Rightarrow C)\,, \\ u \upharpoonright C, D & \in & \downarrow\uparrow\text{-}\mathsf{Pla\tilde{y}s}(C \Rightarrow D)\,, \\ u \upharpoonright A, D & \in & \downarrow\uparrow\text{-}\mathsf{Pla\tilde{y}s}(A \Rightarrow D)\,. \end{array}$$

The key argument is the so-called "zipping lemma" ([Harmer, 2004, Lemma 2.6.1]):

Lemma 4.1.10. *Consider $u \in \mathtt{I}(A, C, D)$ and $v \in \mathtt{I}(A, B, C)$ s.t. $u \upharpoonright A, C = v \upharpoonright A, C$. There is a unique $w \in \mathtt{I}(A, B, C, D)$ s.t. $w \upharpoonright A, C, D = u$, $w \upharpoonright A, B, C = v$.*

This is proved by induction on u. As the name suggests, the idea is to "zip" u and v together, using subsequences from v to provide the moves in B missing from u. There is also a mirror lemma zipping $u \in \mathtt{I}(A, B, D)$ and $v \in \mathtt{I}(B, C, D)$.

Associativity of composition follows:

Proposition 4.1.11. *Consider $\sigma : A \Rightarrow B, \tau : B \Rightarrow C$ and $\nu : C \Rightarrow D$. Then, $\nu \odot (\tau \odot \sigma) = (\nu \odot \tau) \odot \sigma$.*

Proof. Consider $s \in \nu \odot (\tau \odot \sigma)$. By definition, there is a witness $u \in (\tau \odot \sigma) \parallel \nu$, *i.e.* $u \in \mathtt{I}(A, C, D)$ s.t. $u \upharpoonright A, C \in \tau \odot \sigma$, $u \upharpoonright C, D \in \nu$, and $u \upharpoonright A, D \in \downarrow\uparrow\text{-}\mathsf{Pla\tilde{y}s}(A \Rightarrow D)$. By definition of composition again, there is $v \in \sigma \parallel \tau$ s.t. $v \upharpoonright A, C = u \upharpoonright A, C$. By the

zipping lemma, we can find $w \in \mathtt{I}(A, B, C, D)$ s.t. $w \upharpoonright A, C = v$ and $w \upharpoonright A, C, D = u$; and $w \upharpoonright A, B \in \sigma, w \upharpoonright B, C \in \tau$ and $w \upharpoonright C, D \in \nu$.

One can then show that $w \upharpoonright B, D$ is alternating (this follows from a state diagram for ternary interactions extending that for interactions in Section 4.1.2). Hence $w \upharpoonright B, C, D \in \tau \parallel \nu$, hence $w \upharpoonright B, D \in \nu \odot \tau$. Since we also have $w \upharpoonright A, B \in \sigma$, we have $w \upharpoonright A, B, D \in \sigma \parallel (\nu \odot \tau)$, so $w \upharpoonright A, D \in (\nu \odot \tau) \odot \sigma$ as required.

The other inclusion is symmetric, using the mirror zipping lemma. □

Identities. Identities in game semantics are given by *copycat strategies*.

The intuitive behaviour of copycat is very simple: it copies whatever move Opponent plays on either side, to the matching Player move on the other side, maintaining the invariant that we have the same play on A on both sides. More concretely:

Definition 4.1.12. *Consider A any arena.*

*A **copycat play** on A is any P-visible $s \in \Updownarrow\text{-}\mathsf{Plays}(A_1 \Rightarrow A_2)$ which is:*

$$\text{balanced:} \quad \forall ta^+ \sqsubseteq s,\ ta \upharpoonright A_1 = ta \upharpoonright A_2\,,$$

and we define $\mathfrak{cc}_A : A \Rightarrow A$ as the set of copycat plays on A.

Copycat is a strategy: most conditions are direct, save for determinism that is a bit subtle. The issue is easy to miss as we tend to leave the management of pointers implicit: by *balanced*, an initial move on the left hand side can only immediately follow an initial move on the right hand side. But its *pointer* is not uniquely determined by *balanced*: if there are several initial moves on the right hand side, each can receive the pointer in a way compatible with *balanced*. This is however banned by the P-visible requirement, as in that case only the previous move appears in the P-view.

Copycat is indeed an identity for composition:

Lemma 4.1.13. *Consider $\sigma : A \Rightarrow B$ any strategy. Then, $\mathfrak{cc}_B \odot \sigma = \sigma \odot \mathfrak{cc}_A = \sigma$.*

Proof. By *receptive*, it suffices to prove the equality for even-length plays, *i.e.* Player-ending plays. Consider $s \in \mathfrak{cc}_B \odot \sigma$ with even length – write $\mathfrak{cc}_B : B_1 \Rightarrow B_2$ – and consider its witness $u \in \sigma \parallel \mathfrak{cc}_B$. By the state diagram for interactions, u is in state OOO, therefore $u \upharpoonright B_1, B_2$ is in state OO and is Player-ending, so $u \upharpoonright B_1 = u \upharpoonright B_2$. It follows that $u \upharpoonright A, B_1 = u \upharpoonright A, B_2 = s$, but $u \upharpoonright A, B_1 \in \sigma$, hence $s \in \sigma$.

Reciprocally, from $s \in \sigma$ with even length it is direct to construct by induction on s an interaction $u \in \sigma \parallel \mathfrak{cc}_B$ such that $u \upharpoonright A, B_1 = u \upharpoonright A, B_2 = s$, so that $s \in \mathfrak{cc}_B \odot \sigma$ as required. Symmetrically, we have $\sigma \odot \mathfrak{cc}_A = \sigma$ as well. □

We conclude from all of the above:

Corollary 4.1.14. *There is a category $\Updownarrow$-**Strat** (respectively $\Updownarrow$-**WB**, $\Updownarrow$-**Inn**, $\Updownarrow$-**InnWB**) with arenas as objects, and as morphisms from A to B all P-visible single-threaded strategies (respectively all P-visible well-bracketed strategies, all innocent strategies, all innocent well-bracketed strategies) on $A \Rightarrow B$.*

4.1.4 Cartesian Closed Structure

Next, we show that $\downarrow\uparrow$-**Strat**, $\downarrow\uparrow$-**WB**, $\downarrow\uparrow$-**Inn** and $\downarrow\uparrow$-**InnWB** are cartesian closed.

Cartesian structure. The empty arena 1 is terminal in all four categories: for all arena A, $A \Rightarrow 1 = 1$, inhabited with a unique, empty, strategy. The *product* of A and B was defined in Definition 4.1.1. For arenas A, B, the two projections

$$\pi_A : A \times B \Rightarrow A\,, \qquad \pi_B : A \times B \Rightarrow B$$

are the copycat strategies defined with the obvious adaptation of Definition 4.1.12.

To define the pairing, we prove:

Lemma 4.1.15. *Consider Γ, A, B arenas, and $\sigma : \Gamma \Rightarrow A$, $\tau : \Gamma \Rightarrow B$ single-threaded. There exists a unique (necessarily) single-threaded strategy*

$$\langle \sigma, \tau \rangle : \Gamma \Rightarrow A \times B$$

comprising all $s \in \downarrow\uparrow$-Plays$(\Gamma \Rightarrow A \times B)$ such that for all $t \sqsubseteq s$, $\lceil t \rceil \in \sigma$ or $\lceil t \rceil \in \tau$.

Proof. Analogously to innocent strategies and P-views, single-threaded strategies are uniquely determined by their set of threads, see *e.g.* [Harmer, 2004, Section 2.8]. □

Notice, in this definition, the implicit renaming of $\lceil t \rceil \in \downarrow\uparrow$-Plays$(\Gamma \Rightarrow A \times B)$ to $\downarrow\uparrow$-Plays$(\Gamma \Rightarrow A)$ or $\downarrow\uparrow$-Plays$(\Gamma \Rightarrow B)$ depending whether the initial move is in A or B.

This data satisfies the required equations for cartesian products:

Lemma 4.1.16. *Consider $\sigma : \Gamma \Rightarrow A$, $\tau : \Gamma \Rightarrow B$, and $\nu : \Delta \Rightarrow \Gamma$ single-threaded. Then, the following equations hold:*

$$\begin{array}{rcl} \pi_A \odot \langle \sigma, \tau \rangle & = & \sigma \\ \pi_B \odot \langle \sigma, \tau \rangle & = & \tau \\ \langle \pi_A, \pi_B \rangle & = & \mathbf{c}_{A \times B} \\ \langle \sigma, \tau \rangle \odot \nu & = & \langle \sigma \odot \nu, \tau \odot \nu \rangle\,. \end{array}$$

Proof. The first two equations are direct variations of Lemma 4.1.13. The third is direct from the definition, and the fourth immediate on threads, using again that single-threaded strategies are uniquely determined by their sets of threads. □

From the equations above, the following fact follows immediately.

Corollary 4.1.17. *The categories $\downarrow\uparrow$-**Strat**, $\downarrow\uparrow$-**WB** and $\downarrow\uparrow$-**Inn** are cartesian.*

Cartesian closed structure. Now, we describe the cartesian closure.

For any two arenas A and B, we have already defined their arrow $A \Rightarrow B$. We have:

Lemma 4.1.18. *Consider* Γ, A, *and* B *three arenas. Then we have*

$$\Xi_{\Gamma,A,B} \quad : \quad (\Gamma \times A) \Rightarrow B \quad \cong \quad \Gamma \Rightarrow (A \Rightarrow B)$$

an ***isomorphism of arenas****,* i.e. *a bijection preserving and reflecting all structure.*

Proof. Straightforward from the definition. □

Up to renaming, $\Gamma \times A \Rightarrow B$ and $\Gamma \Rightarrow A \Rightarrow B$ are the same. Thus this means that the currying of $\sigma : \Gamma \times A \Rightarrow B$ can be described by applying $\Xi_{\Gamma,A,B}$ move-by-move:

Lemma 4.1.19. *Consider* Γ, A *and* B *three arenas. Then, there is a bijection*

$$\Lambda_{\Gamma,A,B} \quad : \quad \downdownarrows\text{-}\mathbf{Strat}(\Gamma \times A, B) \quad \simeq \quad \downdownarrows\text{-}\mathbf{Strat}(\Gamma, A \Rightarrow B)$$

preserving and reflecting innocence, and preserving composition in the sense that

$$\Lambda_{\Gamma,A,B}(\sigma) \odot \gamma \quad = \quad \Lambda_{\Delta,A,B}(\sigma \odot (\gamma \times A)) \tag{4.1}$$

for all $\sigma \in \downdownarrows\text{-}\mathbf{Strat}(\Gamma \times A, B)$ *and* $\gamma \in \downdownarrows\text{-}\mathbf{Strat}(\Delta, \Gamma)$.

Proof. A routine manipulation on interactions. □

In turn, this bijection lets us define the evaluation strategy simply as

$$\mathsf{ev}_{A,B} \quad = \quad \Lambda^{-1}_{A,A,B}(\mathbf{cc}_{A \Rightarrow B}) \quad \in \quad \downdownarrows\text{-}\mathbf{Strat}((A \Rightarrow B) \times A, B)$$

and from the above, it becomes routine to prove cartesian closure:

Proposition 4.1.20. $\downdownarrows$-**Strat**, $\downdownarrows$-**WB**, $\downdownarrows$-**Inn** *and* $\downdownarrows$-**InnWB** *are cartesian closed.*

Proof. We need the two equalities for the universal property of cartesian closure, for $\sigma \in \downdownarrows\text{-}\mathbf{Strat}(\Gamma \times A, B)$, $\gamma \in \downdownarrows\text{-}\mathbf{Strat}(\Delta, \Gamma)$ and $\tau \in \downdownarrows\text{-}\mathbf{Strat}(\Gamma, A \Rightarrow B)$:

$$\mathsf{ev}_{A,B} \odot (\Lambda(\sigma) \times A) \quad = \quad \sigma \tag{4.2}$$

$$\Lambda(\mathsf{ev}_{A,B} \odot (\tau \times A)) \quad = \quad \tau\,, \tag{4.3}$$

but those follow from elementary computation using definition of $\mathsf{ev}_{A,B}$, (4.1), neutrality of copycat for composition and the fact that Λ is a bijection. □

Since $\Lambda(-)$ preserves and reflects innocence and well-bracketing, the cartesian closed structure automatically holds for our four categories of interest.

Internal language. This cartesian closed structure allows us to use notations from the λ-calculus to manipulate strategies: these may be unfolded to primitive operations of cartesian closed categories in the standard way [Lambek and Scott, 1988]. For instance, the *application* of $\sigma \in \Updownarrow\text{-}\mathbf{Strat}(\Gamma, A \Rightarrow B)$ to $\tau \in \Updownarrow\text{-}\mathbf{Strat}(\Gamma, A)$ unfolds to

$$\sigma\,\tau = \mathsf{ev}_{A,B} \odot \langle \sigma, \tau \rangle\,,$$

likewise we will sometimes in the sequel use λ-abstraction on expressions for strategies.

4.1.5 Recursion

To handle recursion we must provide a *fixpoint operator*, *i.e.* an innocent strategy

$$\mathcal{Y}_A : (A \Rightarrow A) \Rightarrow A$$

for each arena A such that for every $\sigma \in \Updownarrow\text{-}\mathbf{Strat}(\Gamma, A \Rightarrow A)$, we have

$$\mathcal{Y}_A\,\sigma = \sigma\,(\mathcal{Y}_A\,\sigma)$$

a fixpoint equation. It is solved as usual in denotational semantics, by constructing $\mathcal{Y}_A$ as a least upper bound for an adequate order on strategies – the following is direct:

Proposition 4.1.21. *For arenas A and B, $\Updownarrow\text{-}\mathbf{Strat}(A, B)$, $\Updownarrow\text{-}\mathbf{WB}(A, B)$, $\Updownarrow\text{-}\mathbf{Inn}(A, B)$ and $\Updownarrow\text{-}\mathbf{InnWB}(A, B)$ are partially ordered by inclusion, yieling pointed dcpos.*

All operations on strategies involved in the cartesian closed structure are continuous, making $\Updownarrow\text{-}\mathbf{Strat}$, $\Updownarrow\text{-}\mathbf{WB}$, $\Updownarrow\text{-}\mathbf{Inn}$ and $\Updownarrow\text{-}\mathbf{InnWB}$ enriched over pointed dcpos.

Using this, we can obtain the fixpoint operator as the least upper bound of

$$\begin{array}{rcccc} F & : & \Updownarrow\text{-}\mathbf{Strat}(A \Rightarrow A, A) & \to & \Updownarrow\text{-}\mathbf{Strat}(A \Rightarrow A, A) \\ & & \sigma & \mapsto & \lambda f^{A \Rightarrow A}.\, f\,(\sigma\, f) \end{array}$$

which unfolds to the following, for $\bot$ the minimal strategy just closed under receptivity:

$$\mathcal{Y}_A \quad = \quad \bigcup_{n \in \mathbb{N}} F^n(\bot) \quad \in \quad \Updownarrow\text{-}\mathbf{Strat}(A \Rightarrow A, A) \tag{4.4}$$

satisfying $\mathcal{Y}_A = F\,\mathcal{Y}_A$ by construction – we may then prove the fixpoint equation:

$$\begin{array}{rcl} \mathcal{Y}_A\,\sigma & = & F(\mathcal{Y}_A)\,\sigma \\ & = & (\lambda f^{A \Rightarrow A}.\, f\,(\mathcal{Y}_A\, f))\,\sigma \\ & = & \sigma\,(\mathcal{Y}_A\,\sigma) \end{array}$$

using β-equivalence on the λ-calculus notation [Lambek and Scott, 1988].

4.2 Interpretation of IA

Following [Lambek and Scott, 1988], the cartesian closed structure developed above yields an interpretation of the simply-typed λ-calculus; we also showed the interpretation of the fixpoint combinator. In order to obtain the interpretation of PCF and IA, it remains to provide strategies matching the primitives of our programming languages.

4.2.1 Interpretation of PCF

Let us focus first on the purely functional primitives, *i.e.* on the interpretation of PCF.

Interpretation of types. First of all, to every type A we associate an arena: this is done exactly as in Section 3.2.1, by setting $[\![\mathbb{B}]\!] = \mathbf{B}$, $[\![\mathbb{N}]\!] = \mathbf{N}$ and $[\![\mathbb{U}]\!] = \mathbf{U}$ the arenas of 3.8, 3.9 and 3.7 respectively. Arrow types are interpreted as $[\![A \to B]\!] = [\![A]\!] \Rightarrow [\![B]\!]$ using the arrow arena construction of Definition 4.1.3, yielding

$$[\![A]\!] \in \mathrm{\Downarrow\!\uparrow}\text{-}\mathbf{Strat}$$

an arena for every type A of PCF. This is extended to contexts by

$$[\![x_1 : A_1, \ldots, x_n : A_n]\!] = \bigotimes_{1 \leq i \leq n} [\![A_i]\!] .$$

Interpretation of terms. The general methodology of the interpretation of terms is that of the interpretation of a simply-typed λ-calculus in a cartesian closed category. This means that a term[2] $\Gamma \vdash M : A$ gets interpreted as a morphism in $\mathrm{\Downarrow\!\uparrow}$-$\mathbf{Strat}$:

$$[\![M]\!] \in \mathrm{\Downarrow\!\uparrow}\text{-}\mathbf{Strat}([\![\Gamma]\!], [\![A]\!]) .$$

The primitives for the simply-typed λ-calculus are covered by the cartesian closed structure of $\mathrm{\Downarrow\!\uparrow}$-$\mathbf{Strat}$ (see [Lambek and Scott, 1988] for details). For recursion, we set

$$[\![\mathcal{Y}\, M]\!] = \mathcal{Y}_{[\![A]\!]} \odot [\![M]\!]$$

for $\Gamma \vdash M : A \to A$, using the strategy $\mathcal{Y}_A$ of (5.2). It remains to provide interpretations for all the additional combinators that PCF adds to the simply-typed λ-calculus. First of all, all constants are interpreted by the obvious strategy directly returning the corresponding value. In Figure 4.4, we show the typical plays of strategies for sequential composition, conditionals, successor, and zero test; while Figure 4.6 shows typical plays for the predecessor, and Figure 4.7 shows the typical plays for the let binding strategy. Using these strategies, the strategy is obtained by the clauses in Figure 4.5.

4.2.2 Interpretation of State

We complete the interpretation to provide semantics for full IA.

Interpretation of types. Before we describe the interpretation of stateful primitives, we must provide arenas for the two additional types $\mathbb{V}$ and $\mathbb{S}$ that IA adds to PCF.

Intuitively, these arenas describe the interface through which a program may interact with a reference or a semaphore. For instance, a reference may be written to; or read from – a semaphore may be grabbed and released. It is convenient to split the arena

[2] Or really, a typing derivation – but type annotations in PCF ensure that each term has a unique derivation.

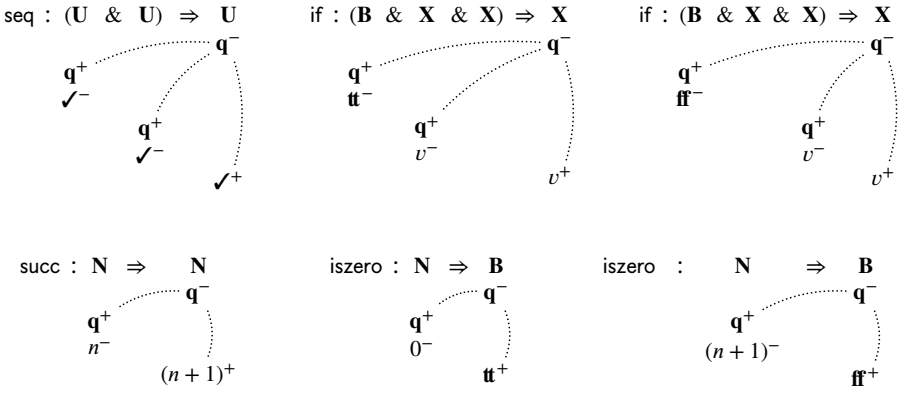
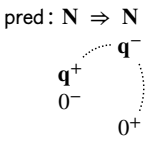


Figure 4.4: Typical plays of basic strategies for PCF

$$
\begin{array}{rcl}
[\![M;\, N]\!] & = & \mathsf{seq} \odot \langle [\![M]\!], [\![N]\!] \rangle \\
[\![\mathbf{if}\ M\ N_1\ N_2]\!] & = & \mathsf{if} \odot \langle [\![M]\!], [\![N_1]\!], [\![N_2]\!] \rangle \\
[\![\mathbf{succ}\ M]\!] & = & \mathsf{succ} \odot [\![M]\!] \\
[\![\mathbf{pred}\ M]\!] & = & \mathsf{pred} \odot [\![M]\!] \\
[\![\mathbf{iszero}\ M]\!] & = & \mathsf{iszero} \odot [\![M]\!] \\
[\![\mathbf{let}\ x = N\ \mathbf{in}\ M]\!] & = & \mathsf{let}_{\mathbf{X},\mathbf{Y}} \odot \langle [\![N]\!], \Lambda([\![M]\!]) \rangle
\end{array}
$$

Figure 4.5: Basic interpretation clauses



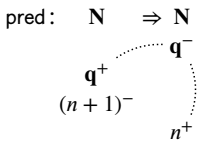


Figure 4.6: Strategy for **pred**

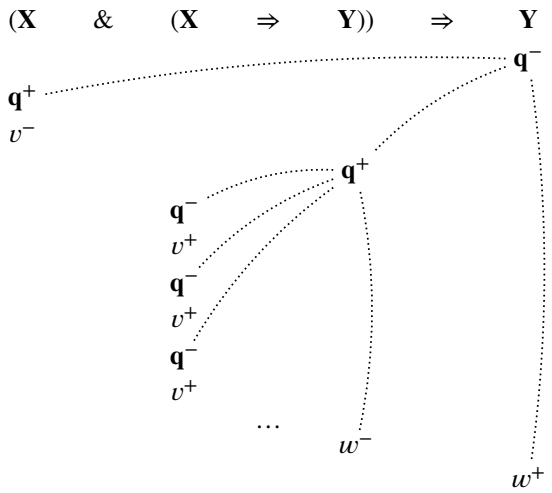


Figure 4.7: Typical plays for $\mathsf{let}_{\mathbf{X},\mathbf{Y}}$

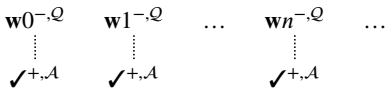


Figure 4.8: $\mathbf{V}_w$

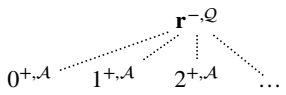


Figure 4.9: $\mathbf{V}_r$

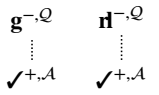


Figure 4.10: $\mathbf{S}$

for references in two: $\mathbf{V}_w$ describes the protocol for *writing* to the memory, and $\mathbf{V}_r$ for *reading* from the memory. Formally, these and $\mathbf{S}$ for semaphores are defined by:

$$\mathbf{V}_w = \bigwith_{n\in\mathbb{N}} \mathbf{U}\,, \qquad \mathbf{V}_r = \mathbf{N}\,, \qquad \mathbf{S} = \mathbf{U}\&\mathbf{U} \tag{4.5}$$

with moves renamed as illustrated in Figures 4.8, 4.9 and 4.10; hopefully making the intention behind the moves self-explanatory. Finally, we pair writing and reading with

$$\mathbf{V} = \mathbf{V}_w\&\mathbf{V}_r\,,$$

and $[\![\mathbb{V}]\!] = \mathbf{V}$, $[\![\mathbb{S}]\!] = \mathbf{S}$ completes the interpretation of all IA types.

Queries. Next, we must provide strategies for the primitives corresponding to state: manipulation and declaration of variables and semaphores. The usual methodology in game semantics is to separate these primitives in two categories. On the one hand, we have the primitives performing queries to the state: assignment, dereferenciation, grab and release. Those are viewed as not inherently stateful, as they simply propagate the operations made by the program to the central memory. On the other hand, reference and semaphore declaration actually introduce the stateful behaviour.

We start with the former, by providing the interpretation of assignment, dereferenciation, grab and release. This is done via the four following clauses:

$$\begin{aligned}
[\![M := N]\!] &= \mathsf{assign} \odot \langle [\![N]\!], [\![M]\!] \rangle \\
[\![!M]\!] &= \mathsf{deref} \odot [\![M]\!] \\
[\![\mathbf{grab}(M)]\!] &= \mathsf{grab} \odot [\![M]\!] \\
[\![\mathbf{release}(M)]\!] &= \mathsf{release} \odot [\![M]\!]
\end{aligned}$$

using the strategies of Figures 4.11, 4.12, 4.13, and 4.14. For the most part those are simply copycat strategies – or more precisely, following (4.5), projections. In particular we insist that they are *innocent strategies*: they are not responsible for the side effect, as their role is simply to forward queries to the actual memory.

Declaration of references. Now, we must provide the interpretation for the declaration of new references, performing the actual side effect.

The *memory cell* implements the stateful behaviour: it acknowledges the write requests; when queried with a read request it outputs the value provided with the latest

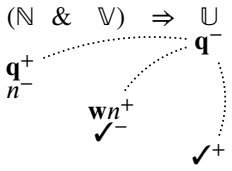


Figure 4.11: assign

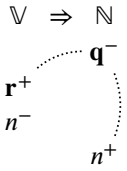


Figure 4.12: deref

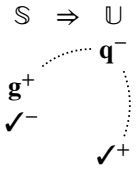


Figure 4.13: grab

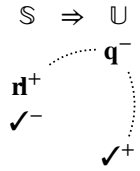


Figure 4.14: release

write. At first ignoring pointers, it seems natural to first capture this behaviour as a language (as in, a set of words) called cell, on moves of $\mathbb{V}$. Hence we define:

$$\mathsf{cell}_n \quad ::= \quad \varepsilon \mid \mathbf{w}k^-\checkmark^+\mathsf{cell}_k \mid \mathbf{r}^-n^+\mathsf{cell}_n$$

and we define cell as the prefix language of cell_0 (matching the fact that new references are initialized to 0). This definition omits pointers, but they are easily reinstated: Opponent moves are initial, while Player moves point to the previous move.

It is tempting to take $\mathsf{cell} : \mathbf{V}$, and for $x : \mathbb{V} \vdash M : A$ a term of IA, to define

$$[\![\mathbf{newref}\ x\ \mathbf{in}\ M]\!] = [\![M]\!] \odot \mathsf{cell}\,,$$

however we run into an issue: cell is not a strategy, since it is not single-threaded. But actually, of course it cannot be single-threaded: single-threadedness ensures that each call is independent, whereas the very nature of cell forces it to link the different queries! To make the definition above single-threaded, we seek to define instead

$$\mathsf{newr}_{\mathbf{X}} : (\mathbf{V} \Rightarrow \mathbf{X}) \Rightarrow \mathbf{X}\,,$$

ensuring that all variable queries belong to the same thread. Fixing $\mathbf{X}$ for now, we set

$$\mathsf{newrt}_n \quad ::= \quad \varepsilon \mid \mathbf{w}k^-\checkmark^+\mathsf{newrt}_k \mid \mathbf{r}^-n^+\mathsf{newrt}_n \mid \upsilon_1^-\upsilon_2^+\mathsf{newrt}_n$$

and set newrt as the prefix language of $\mathbf{q}_2^-\mathbf{q}_1^+\mathsf{newrt}_0$, where the annotations 1 and 2 help distinguish components, writing $(\mathbf{V} \Rightarrow \mathbf{X}_1) \Rightarrow \mathbf{X}_2$. To cell, newrt adds two initial questions $\mathbf{q}_2^-$ and $\mathbf{q}_1^+$ and is prepared at any point to propagate an answer υ_1^- for $\mathbf{q}_1^+$ to the corresponding answer υ_2^+ for $\mathbf{q}_2^-$ – this does not necessarily end the play, and it may happen several times as Opponent may not be well-bracketed. Finally, these plays lack *pointers* – but they are uniquely determined so as to make the plays P-visible.

This defines newrt, a set of *threads*; the corresponding single-threaded strategy is:

Proposition 4.2.1. *Consider* $\mathbf{X}$ *a basic arena,* i.e. *one of* $\mathbf{U}, \mathbf{B}$*, and* $\mathbf{N}$.
Then, there is a unique single-threaded strategy

$$\mathsf{newr}_{\mathbf{X}} : (\mathbf{V} \Rightarrow \mathbf{X}) \Rightarrow \mathbf{X}$$

such that for all $s \in \mathsf{newr}_{\mathbf{X}}$*, we have* $\lceil s \rceil \in \mathsf{newrt}_{\mathbf{X}}$.

$$\begin{array}{rcll} [\![\mathbf{newref}\ x{:=}n\ \mathbf{in}\ M]\!] & = & \mathsf{newr}_{\mathbf{X}} \odot \Lambda([\![M]\!]) & \in \ \downarrow\uparrow\text{-}\mathbf{Strat}([\![\Gamma]\!], [\![A]\!]) \\ [\![\mathbf{newsem}\ x{:=}n\ \mathbf{in}\ M]\!] & = & \mathsf{news}_{\mathbf{X}} \odot \Lambda([\![M]\!]) & \in \ \downarrow\uparrow\text{-}\mathbf{Strat}([\![\Gamma]\!], [\![A]\!])\,. \end{array}$$

Figure 4.15: Interpretation of reference and semaphore declaration

After each Opponent move, there is exactly one active thread: namely, that where Opponent just moved. The single-threaded strategy $\mathsf{newr}_{\mathbf{X}}$ then simply plays as $\mathsf{newrt}_{\mathbf{X}}$ in the active thread, ignoring other threads and never switching between them. Operationally, this corresponds to the fact that state in IA is *local*: each execution of the new reference operation is linked to a separate, dynamically allocated memory cell.

Finally, the interpretation of **newref** is given in Figure 4.15 for $\Gamma, x : \mathbf{V} \vdash M : \mathbf{X}$.

Declaration of semaphores. Now, we perform the same construction for semaphores. The definition is analogous and semaphores are not particularly interesting in a sequential language, so we offer fewer details. The *semaphore cell* is defined first via:

$$\begin{array}{rcl} \mathsf{lock}_0 & ::= & \varepsilon \mid \mathbf{g}^-\checkmark^+\mathsf{lock}_1 \mid \mathbf{rl}^- \\ \mathsf{lock}_{n+1} & ::= & \varepsilon \mid \mathbf{rl}^-\checkmark^+\mathsf{lock}_0 \mid \mathbf{g}^-\,, \end{array}$$

setting lock as the prefix language of lock_0 – pointers are redundant. Observe that in a sequential language, attemping to grab a semaphore that is not free results in divergence; likewise attempting to release a semaphore that is already free yields divergence.

Again, that is not single-threaded. For the actual single-threaded strategy, we set

$$\begin{array}{rcl} \mathsf{newst}_0 & ::= & \varepsilon \mid \mathbf{g}^-\checkmark^+\mathsf{newst}_1 \mid \upsilon_1^-\upsilon_2^+\mathsf{newst}_0 \mid \mathbf{rl}^- \\ \mathsf{newst}_{n+1} & ::= & \varepsilon \mid \mathbf{rl}^-\checkmark^+\mathsf{newst}_0 \mid \upsilon_1^-\upsilon_2^+\mathsf{newst}_{n+1} \mid \mathbf{g}^- \end{array}$$

and set $\mathsf{newst}_{\mathbf{X}}$ as the prefix language of newst_0, where the subscripts 1 and 2 help distinguish components on $(\mathbf{S} \Rightarrow \mathbf{X}_1) \Rightarrow \mathbf{X}_2$. Finally, as for references, we set:

Proposition 4.2.2. *Consider* $\mathbf{X}$ *a basic arena,* i.e. *one of* $\mathbf{U}, \mathbf{B}$*, and* $\mathbf{N}$.
Then, there is a unique single-threaded strategy

$$\mathsf{news}_{\mathbf{X}} : (\mathbf{V} \Rightarrow \mathbf{X}) \Rightarrow \mathbf{X}$$

such that for all $s \in \mathsf{news}_{\mathbf{X}}$*, we have* $\lceil s \rceil \in \mathsf{newst}_{\mathbf{X}}$.

Finally, the interpretation of **newsem** is given in Figure 4.15 for $\Gamma, x : \mathbf{S} \vdash M : \mathbf{X}$. This almost concludes the interpretation of IA, but two final constructions remain.

Bad variables and semaphores. Indeed, recall from Section 2.2.1 that IA includes

$$\frac{\Gamma \vdash M : \mathbb{N} \to \mathbb{U} \qquad \Gamma \vdash N : \mathbb{N}}{\Gamma \vdash \mathbf{mkvar}\ M\ N : \mathbb{V}} \qquad \frac{\Gamma \vdash M : \mathbb{U} \qquad \Gamma \vdash N : \mathbb{U}}{\Gamma \vdash \mathbf{mksem}\ M\ N : \mathbb{S}}$$

the so-called *bad variable* and *bad semaphore* constructs.

Bad variables and semaphores are not sensible programming primitives, they are here by necessity for definability. Indeed, up to isomorphism $\mathbf{V}$ is defined as $(\&_{n\in\mathbb{N}}\mathbf{U})\&\mathbf{N}$, and strategies inhabiting it may not be linked with actual memory cells: they may be formed by pairing arbitrary "write" and "read" methods. Thus, to obtain a finite definability result, one must include such pairings should be formable in the syntax [3].

Through the isomorphisms $\mathbf{V} \cong (\&_{n\in\mathbb{N}}\mathbf{U})\&\mathbf{N}$ and $\mathbf{S} \cong \mathbf{U}\&\mathbf{U}$, the interpretation is:

$$[\![\mathbf{mkvar}\ M\ N]\!] = \langle \langle [\![M]\!]\, n \mid n \in \mathbb{N} \rangle, [\![N]\!] \rangle\,, \qquad [\![\mathbf{mksem}\ M\ N]\!] = \langle [\![M]\!], [\![N]\!] \rangle\,,$$

which are the final two clauses of the interpretation of IA.

4.3 Complements on Alternating Strategies

In this section, we introduce a few complements on the results and constructions of this chapter. Those may be skipped on first reading; they are intended either as a reference, or because they are referred to further along in this monograph.

4.3.1 Complements on Single-Threadedness

First of all, we have introduced in Section 3.3.1 the notion of single-threaded strategies, as a way to enforce that distinct initial moves generate separated copies of the strategy, which may not communicate or share any state. Here we give some complements.

Here we focus on the concept of *threads*:

Definition 4.3.1. *An **alternating thread** on arena A is $t \in \downarrow\uparrow\text{-}\mathsf{Pla\^{y}s}(A)$ with at most one initial move; write $\downarrow\uparrow\text{-}\mathsf{Pla\^{y}s}_{\bullet}(A)$ for the set of alternating threads on A.*

We mean to make formal that if $\sigma : A$ is single-threaded, then it is determined by its set of threads – so it should be equivalently described by a *thread-strategy*:

Definition 4.3.2. *For A an arena, an **alternating thread-strategy** $\sigma : A$ is a set $\sigma \subseteq \downarrow\uparrow\text{-}\mathsf{Pla\^{y}s}_{\bullet}(A)$ satisfying the following conditions:*

$$\begin{array}{rl} \text{non-empty:} & \epsilon \in \sigma\,, \\ \text{prefix-closed:} & \forall s \sqsubseteq t \in \sigma,\ s \in \sigma\,, \\ \text{receptive:} & \forall s \in \sigma,\ sa^- \in \downarrow\uparrow\text{-}\mathsf{Pla\^{y}s}_{\bullet}(A) \implies sa^- \in \sigma\,, \\ \text{deterministic:} & \forall sa^-b_1^+, sa^-b_2^+ \in \sigma,\ sab_1 = sab_2\,, \end{array}$$

*and it is additionally **well-bracketed** if it satisfies*

$$\text{well-bracketing:} \quad \textit{for all } ta^+ \in \sigma\textit{, if } t \textit{ is well-bracketed then so is } ta.$$

[3] A definability result without bad variables can be obtained by switching to *nominal games* [Tzevelekos, 2009], where nominal techniques are employed to attach reference cells to actual memory locations, but the setting gets more complicated. It must also be mentioned that though finite definability fails for IA without bad variables, full abstraction still holds as proved by McCusker [McCusker, 2003]

This is, of course, the same conditions as for Definition 3.1.5, except than they are taken within $\downarrow\uparrow$-$\mathsf{Pla\widehat{y}s}_\bullet(A)$ rather than $\downarrow\uparrow$-$\mathsf{Pla\widehat{y}s}(A)$. Clearly, if $\sigma : A$ is an alternating strategy (single-threaded or not), then $\sigma \cap \downarrow\uparrow$-$\mathsf{Pla\widehat{y}s}_\bullet(A)$ is a thread-strategy.

Reciprocally, we construct a single-threaded strategy from a thread-strategy on A:

Proposition 4.3.3. *Consider σ a thread-strategy on arena A.*
Then, constructing the set $\sigma^!$ comprising all plays $s \in \downarrow\uparrow$-$\mathsf{Pla\widehat{y}s}(A)$ which satisfy

$$\text{threaded-correct:} \quad \textit{for all } ta^+ \sqsubseteq s, \textit{ we have } \lceil ta^+ \rceil \in \sigma,$$

it follows that $\sigma^!$ is a single-threaded strategy on A.

Proof. First, the conditions *non-empty*, *prefix-closed* and *receptive* are clear from the definition. For the rest, we first observe that any $s \in \sigma^!$ is *well-threaded*, in the sense that for all $ta^-b^+ \sqsubseteq s$, b points within $\lceil sa^- \rceil$. Indeed, for each initial move s_i in s, we have $s \upharpoonright i$, comprising the moves of s hereditarily justified by s_i, a thread which by condition *thread-correct* must be in σ thus, in particular, alternating. For each $t \sqsubseteq s$ we may define $t \upharpoonright i$ a prefix of $s \upharpoonright i$. Since each $t \upharpoonright i$ is alternating, it is immediate by induction that for $t \sqsubseteq s$ of even length, all $t \upharpoonright i$ have even length, while if t has odd length, exactly one i_0 is such that $t \upharpoonright i_0$ has odd length. But then, considering $ta^-b^+ \sqsubseteq s$, at ta^- the only thread with odd length must be $\lceil ta^- \rceil$, which b^+ must extend so as to satisfy our invariant; thus b^+ must indeed point within $\lceil sa^- \rceil$.

We prove further conditions. For *deterministic*, if $sa^-b_1^+, sa^-b_2^+ \in \sigma^!$, then we have that $\lceil sa^-b_1^+ \rceil, \lceil sa^-b_2^+ \rceil \in \sigma$, but by our observation above those have the form $ta^-b_1^+, ta^-b_2^+ \in \sigma$ for $ta^- = \lceil sa^- \rceil$. Hence $ta^-b_1^+ = ta^-b_2^+$ by *determinism* of σ, from which follows $sa^-b_1^+ = sa^-b_2^+$. Finally, conditions *well-threaded* and *single-threaded* of Definition 3.3.2 follow directly from our observation above. □

Any thread-strategy yields a single-threaded strategy, but reciprocally all single-threaded strategies have this form, as we show now:

Proposition 4.3.4. *Consider $\sigma : A$ a single-threaded strategy. Then, the set*

$$[\![\sigma]\!] = \{ \lceil s \rceil \mid s \in \sigma \} = \sigma \cap \downarrow\uparrow\text{-}\mathsf{Pla\widehat{y}s}_\bullet(\sigma)$$

is a thread-strategy on A, such that $\sigma = [\![\sigma]\!]^!$.

Proof. It is direct that if $\sigma : A$ is single-threaded, then $[\![\sigma]\!] = \sigma \cap \downarrow\uparrow$-$\mathsf{Pla\widehat{y}s}_\bullet(A)$; from which it is easy that $[\![\sigma]\!]$ satisfies the conditions of Definition 4.3.2.

Now, we prove that $\sigma = [\![\sigma]\!]^!$. For $\subseteq$, consider $s \in \sigma$ and $ta^+ \sqsubseteq s$. Since $ta^+ \in \sigma$, we have $\lceil ta^+ \rceil \in [\![\sigma]\!]$ by definition, hence $s \in [\![\sigma]\!]^!$. For $\supseteq$, consider $s \in [\![\sigma]\!]^!$. It is then direct to prove by induction on s that $s \in \sigma$: for negative extensions, this follows from *receptivity*, for positive extensions from *single-threaded*. □

Thus altogether, this gives single-threaded alternating strategies two representations: as a single-threaded set of plays with many initial moves, or as a set of threads. This exactly reflects the two presentations of innocent strategies, as sets of plays or as sets of P-views, that was described in Section 3.2.4. We observe a useful consequence:

Corollary 4.3.5. *Consider $\sigma, \tau : A$ single-threaded strategies.*
If σ and τ have the same threads, then they are equal.

Proof. Straightforward from Proposition 4.3.4. □

Further conditions. Finally, we show how this correspondence preserves the further conditions we considered on alternating strategies. Observe first that the notions of *well-bracketing* and *P-visibility*, respectively introduced in Definitions 3.2.6 and 3.2.8, are conditions on plays and as such transparently apply to thread-strategies. Then:

Lemma 4.3.6. *Consider $\sigma : A$ a thread-strategy. Then:*

(1) *$\sigma^!$ is well-bracketed iff σ is well-bracketed,*
(2) *$\sigma^!$ is P-visible iff σ is P-visible.*

Proof. (1) Only if. Straightforward since $\sigma \subseteq \sigma^!$. *If.* We show by induction on s that for all $s \in \sigma^!$, the pending question is always in the current thread – the result follows.
(2) Only if. As above, follows from $\sigma \subseteq \sigma^!$. *If.* Follows from the easy observation that the P-view is always contained within the current thread. □

4.3.2 Factorization and Definability

With the details of the interpretation in place, we revisit some of the properties of the semantics presented in Chapter 3, providing key steps of the definability proof.

Innocent definability. First of all, recall that all finite innocent well-bracketed strategies are definable within PCF (Theorem 3.2.14). But this result only concerns PCF types – we must first generalize it to all IA types, including references and semaphores:

Theorem 4.3.7. *Let A be an* IA *type, and $\sigma : [\![A]\!]$ be finite well-bracketed innocent.*
There is an IA *term $\vdash M : A$, without* **newref** *or* **newsem**, *s.t. $[\![M]\!] = \sigma$.*

Proof. Direct variation of Theorem 3.2.14, crucially using **mkvar** and **mksem**. □

Factorization. Thus, extending types from PCF to IA does not affect the innocent definability result. But how can it be generalized to *non-innocent* strategies?

For IA, the main argument is a *factorization* result [Abramsky and McCusker, 1996]:

Proposition 4.3.8. *Consider an arena $A = A_1 \Rightarrow \ldots \Rightarrow A_n \Rightarrow \mathbf{X}$, and $\sigma : A$ visible, well-bracketed, single-threaded. Then there is an innocent $\mathbf{Inn}(\sigma) : \mathbf{V} \Rightarrow A$ such that*

$$\mathsf{newr}_A \odot \mathbf{Inn}(\sigma) = \sigma$$

for newr_A obtained from $\mathsf{newr}_{\mathbf{X}}$ via the cartesian closed structure in the obvious way.
*Moreover, if σ is finite (*resp. *computable), then so is $\mathbf{Inn}(\sigma)$.*

Sketch. As an innocent strategy, $\mathbf{Inn}(\sigma)$ can only depend on the P-view and hence cannot act as σ, which depends on the full play. But the idea is that $\mathbf{Inn}(\sigma)$ can access the full play indirectly, by maintaining in the reference (an encoding of) the full play. More precisely, each time it receives an Opponent move a^-, $\mathbf{Inn}(\sigma)$ first reads the reference and obtains (the encoding of) a play $s \in \sigma$. Assuming $sa^-b^+ \in \sigma$, Player stores (the encoding of) sa^-b^+ in the reference, then plays b^+. □

This provides the missing argument for finite definability (and hence intensional full abstraction) for IA: if $[\![A]\!] = A_1 \Rightarrow \ldots \Rightarrow A_n \Rightarrow \mathbf{X}$ and $\sigma : [\![A]\!]$ is a finite visible, well-bracketed and single-threaded strategy, then one first obtains a finite innocent and well-bracketed $\mathbf{Inn}(\sigma) : \mathbf{V} \Rightarrow [\![A]\!]$ by Proposition 4.3.8. By Theorem 4.3.7, there is $\vdash M : \mathbb{V} \to A$ in IA such that $[\![M]\!] = \mathbf{Inn}(\sigma)$, but then, as required,

$$[\![\lambda x_1 \ldots x_n.\, \mathbf{newref}\ r\ \mathbf{in}\ M\ x_1 \ldots x_n\, r]\!] = \sigma : [\![A]\!]$$

concluding the proof of finite definability for IA – note that if Theorem 4.3.7 is extended to a *universality result* as is expected following Conjecture 3.2.15, we get that every *computable* P-visible well-bracketed single-threaded strategy in IA is definable.

4.3.3 Summary of Results

Now that we have introduced the proper categories supporting the interpretations underlying Chapter 3, we restate for future reference the main theorems of that chapter with proper reference to the target categories of the interpretations:

Proposition 4.3.9. *There is a computationally adequate interpretation of* IA *in* $\downarrow\uparrow$-**Strat**.

Proof. First stated as Proposition 4.3.9. □

Theorem 4.3.10. *The interpretation of Proposition 4.3.9 refines into an interpretation of* PCF *in* $\downarrow\uparrow$-**InnWB**, *which is intensionally fully abstract.*

Proof. First stated as Theorem 3.2.16. □

Theorem 4.3.11. *The interpretation of Proposition 4.3.9 refines into an interpretation of* IA *in* $\downarrow\uparrow$-**WB**, *which is intensionally fully abstract.*

Proof. First stated as Theorem 3.3.5. □

4.4 Conclusions and Historical Notes

The compositional setting presented here is robust to a number of variations: for instance, *determinism* can be relaxed, obtaining an intensionally fully abstract model for an extension of IA with non-deterministic choice[4] [Harmer and McCusker, 1999]. Non-deterministic strategies can additionally be weighted by *probabilities*, yielding a model intensionally fully abstract for an extension of IA with probabilistic choice

[4]That is not the case with AJM games – determinism is crucial in the way AJM games handle uniformity.

[Danos and Harmer, 2000]. However, neither of these two extensions support a notion of innocence banning state: they do not restrict to intensionally fully abstract models for either non-deterministic or probabilistic PCF. Among other things, this is what this monograph aims to solve, as motivated in the introduction.

Beyond those non-deterministic and probabilistic extensions, this alternating compositional setting presented here has seen so many extensions and applications that it would be a daunting task to attempt to survey them exhaustively, and we leave this beyond the scope of this short conclusion. Many extensions keep the general structure of plays and strategies, changing mostly the arenas: a crucial example of that is game semantics for call-by-value [Abramsky and McCusker, 1997, Honda and Yoshida, 1999], or for polarized versions of linear logic [Laurent, 2002]. Other extensions build on the present compositional mechanism, but bring additional structure to moves and/or plays, such as *names* [Tzevelekos, 2009] or *exception pointers* [Laird, 2001a].

A more drastic change is to abandon *alternation* in order to model concurrent computation [Laird, 2001b, Ghica and Murawski, 2008], but this does affect rather deeply the compositional machinery presented in this chapter: we present this next.

Chapter 5

Non-Alternating Game Semantics

In this chapter, we continue our introduction to game semantics by presenting the non-alternating version of the games model seen so far. The chapter roughly follows the outline of the previous two chapters: first, we introduce non-alternating plays and strategies appealing to operational intuitions in the style of Chapter 3. Then, we introduce the category $\circlearrowleft$-**Strat**, describe the interpretation of $\mathsf{IA}_{/\!/}$, and prove its properties.

The constructions of this chapter mostly follow [Ghica and Murawski, 2008].

5.1 Concurrency and Non-Alternation

Thus, let us start again from the intuitions described in Section 3.1, representing interactive executions as plays, and ask ourselves the following question: how should plays be amended if execution is no longer sequential, but becomes concurrent?

5.1.1 Non-Alternating Plays

Non-alternation. As a first, example, consider the following program M from $\mathsf{PCF}_{/\!/}$

$$x : \mathbb{N}, y : \mathbb{N} \vdash \mathbf{let} \begin{pmatrix} x_1 & = & x \\ x_2 & = & y \end{pmatrix} \mathbf{in}\ x_1 + x_2 : \mathbb{N},$$

which evaluates x and y in parallel, then returns their sum. Or more precisely, the intention of the program is to evaluate x and y in parallel, but really the execution order depends on the scheduler. It may still be the case that x is evaluated before y, as in the first example of Section 3.1.1. Or, perhaps, the scheduler will evaluate y before x. Or

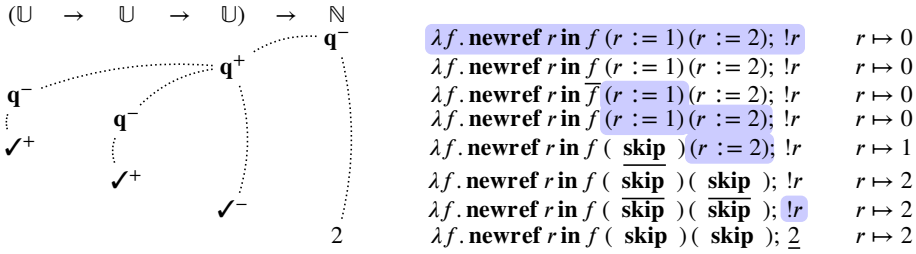


Figure 5.1: Operational content of a non-alternating play

they might indeed be evaluated simultaneously, so we must account for the dialogue

O : "What is the return value?"
P : "What is the value of x?"
P : "What is the value of y?"
O : "The value of x is 12."
O : "The value of y is 30."
P : "The value of $x + y$ is 42."

where the evaluation request for y does not wait for the return value for x.

This naturally brings us to the following variation over Definition 3.1.4:

Definition 5.1.1. *A **non-alternating pre-play** on A is a legal pointing sequence on A. We write $\circlearrowleft$-PrePlays(A) for the set of non-alternating pre-plays on A.*

This is simply Definition 3.1.4 without the condition *alternating*. In other words, *non-alternating pre-play* is a synonym for *legal pointing string*, though we prefer to keep the names separate for conceptual clarity.

Plays and executions. Non-alternating pre-plays are represented with the same conventions as alternating plays. As for alternating plays, the non-alternating pre-plays of a term may be thought of operationally, as illustrated in Figure 5.1.

This example also shows that even for programs which do not use parallel evaluation as a primitive, the switch from alternating plays to non-alternating plays is impactful: plays may also describe executions where it is Opponent who first plays concurrently. In Figure 5.1, Opponent, playing for the argument $f : \mathbb{U} \to \mathbb{U} \to \mathbb{U}$, simultaneously calls its two arguments. This triggers a *race*, as the two write requests are being evaluated simultaneously, yielding a non-deterministic result. So in non-alternating games, terms from IA – a deterministic language – may still have non-deterministic behaviour.

Figure 5.1 is misleading in one respect: it makes it seem like the Player move $\checkmark^+$ *coincides* with the memory update. In reality, it is an acknowledgment for which we can only know for sure that it comes *after* the memory update: the message may stay stuck

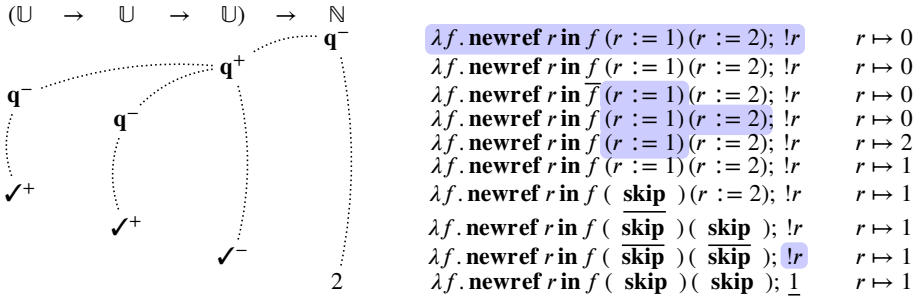
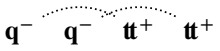


Figure 5.2: Alternative result

in a buffer for a while. The behaviour in Figure 5.2, with exactly the same play until the final result, is just as possible. The consequence of this is that the order in which consecutive acknowledgments arrive is *not* representative of the order in which memory operations are handled: it is irrelevant and cannot influence subsequent behaviour.

Well-bracketing. As in sequential games, constructing the games model will require us to introduce a notion of *well-bracketing*. In our presentation of sequential games, *plays* were not well-bracketed; instead well-bracketing was a property of *strategies*[1]. Here we make the opposite choice and impose it directly on plays: defining cleanly non-alternating strategies for the primitives of $\mathsf{IA}_{/\!/}$ without well-bracketing becomes very challenging, as we are overwhelmed by the number of plays.

As in the alternating case, non well-bracketed plays allow behaviours typical of control operators such as **callcc** – with the new addition of parallel versions, such as the **fork** operator on $\mathbb{B}$ that answers **tt** and **ff** *in parallel*, see [Castellan, 2015]. As in the alternating case, these behaviours must be banned via an adequate condition. By analogy we still call it *well-bracketing*, though this is a bit of a misnomer: for instance,



is a perfectly acceptable play for the constant **tt**, though the third move does not answer the pending question. Intuitively, this play is an interleaving of two evaluations of **tt** progressing independently. While each, taken independently, is indeed well-bracketed in the sense of alternating game semantics, this is blurred out in their interleaving – the notion of "pending question" does not make sense anymore.

Thus we must constrain the interplay between questions and answers differently:

Definition 5.1.2. *Consider A an arena, and s a non-alternating pre-play.*

[1]For modeling IA, both choices are possible, but our discussion on the semantic cube in Section 3.3.3 requires non well-bracketed plays as doing otherwise is incompatible with the interpretation of **callcc**.

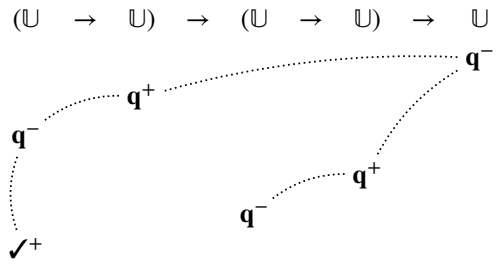


Figure 5.3: Logical well-bracketing is weaker than well-bracketing

*We say that s is **logically well-bracketed** iff it satisfies the conditions:*

fork: *for $t\ q \overset{\frown}{\ } t'\ m \sqsubseteq s$, q is unanswered in t'.*

join: *for $t\ q \overset{\frown}{\ } t'\ a^{\mathcal{A}} \sqsubseteq s$, all questions justified by q are answered in t'.*

By *answer-closing* of Definition 3.2.4, moves labelled q are necessarily questions in this definition. Rather than *chronological*, the constraint is *logical*; it is about the hierarchical relationship between functions and arguments. *Fork* means that if a call returns, it can neither return again nor investigate its arguments anymore; *join* means that a call cannot return if there are un-terminated argument calls currently running.

For $s \in \updownarrow\text{-}\mathsf{Pla\overset{\frown}{y}s}(A)$ alternating, if s is well-bracketed (in the sense of Definition 3.2.5) then it is logically well-bracketed (in the sense of Definition 5.1.2. However, the converse does not hold as illustrated by Figure 5.3. We may then define:

Definition 5.1.3. *A **non-alternating play** on A is a (logically) well-bracketed non-alternating pre-play on A. We write $\circlearrowleft\text{-}\mathsf{Pla\overset{\frown}{y}s}(A)$ for alternating legal plays on A.*

Recall that the set of alternating legal plays on A was written $\updownarrow\text{-}\mathsf{Pla\overset{\frown}{y}s}(A)$, where $\updownarrow$ evokes the transition between states O and P, depending on whose turn it is. Here, $\circlearrowleft$ evokes the fact that there is only one state, where both players can potentially play.

It is a puzzling fact that logical well-bracketing may replace chronological well-bracketing in alternating strategies also, and still get intensional full abstraction for IA (but not definability), despite the example in Figure 5.3. This boils down to the fact that logically well-bracketed *complete* plays (*i.e.* P- and O-visible, and all questions have an answer) are automatically well-bracketed – so *e.g.* Figure 5.3 cannot be completed.

5.1.2 Non-Alternating Strategies

Now, for strategies, which conditions of Definition 3.1.5 survive? *Non-empty*, *prefix-closed* and *receptive* still make sense. However *deterministic* is unreasonable, firstly because of the non-determinism introduced by the scheduler's choices, and secondly because a non-deterministic choice may be defined in $\mathsf{IA}_{/\!/}$ (see **choice** in Section 2.2.3).

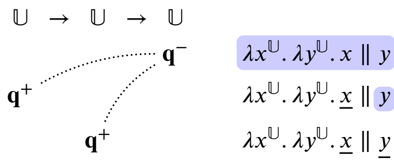


Figure 5.4: Two consecutive P-moves

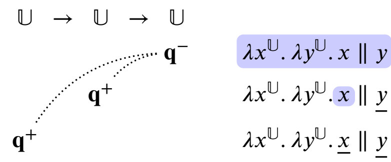


Figure 5.5: The other order

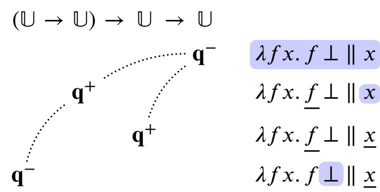


Figure 5.6: Two exchangeable moves

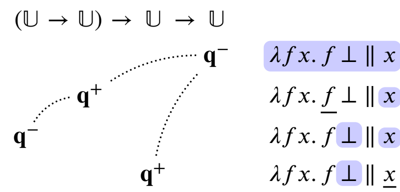


Figure 5.7: The other order

As a first approximation of non-alternating strategies, we set:

Definition 5.1.4. *Consider A an arena.*
A ***non-alternating prestrategy*** $\sigma : A$ *is a set* $\sigma \subseteq \circlearrowleft\text{-}\widehat{\mathsf{Plays}}(A)$ *satisfying:*

$$\begin{aligned} \text{non-empty:} &\quad \epsilon \in \sigma\,, \\ \text{prefix-closed:} &\quad \forall s \sqsubseteq t \in \sigma,\ s \in \sigma\,. \end{aligned}$$

Though *receptive* still makes sense, it is more useful technically to let prestrategies not be receptive, and impose receptivity only later on in Definition 5.1.5.

Saturation. Perhaps less obvious is the fact that we need a new condition.

We already mentioned that in a non-alternating play, part of the order in which moves appear is irrelevant. Imagine a program features consecutive Player moves:

$$\dots m_1^+ m_2^+ \cdots \in \circlearrowleft\text{-}\widehat{\mathsf{Plays}}(A)\,.$$

Since the play is non-alternating, according to our operational intuition this means that two threads are currently under execution. So we are as in Figure 5.4, which illustrates that: *(1)* Player has no control over the order in which the two positive moves appear, which is the scheduler's prerogative; and *(2)* if the positive moves are exchanged, we reach nevertheless the same program state. So the order of two contiguous Player moves is irrelevant, they can be exchanged without affecting the future of the execution.

A symmetric situation is that of two consecutive Opponent moves, in which case, likewise their order is irrelevant and they can be exchanged without affecting the future of execution. Finally, consider a program in $\mathsf{IA}_{/\!/}$ with a play of the form

$$\dots m_1^+ m_2^- \cdots \in \circlearrowleft\text{-}\mathsf{Pla\widehat{y}s}(A)$$

where m_2 is *not* justified by m_1. In that case we are as in Figure 5.6 and the positive m_1 can always be postponed until after the negative m_2, reaching the same program state, and thus not affecting the future computation. But finally, one cannot always postpone an Opponent move after a Player move: the Player move may actually depend on the Opponent move. Overall, the relevant information in $s \in \circlearrowleft\text{-}\mathsf{Pla\widehat{y}s}(A)$ is only that which survives permutations as described above; and accordingly we shall require that non-alternating strategies should be *saturated* under those.

Integrating this condition, the definition of non-alternating strategies becomes:

Definition 5.1.5. *Consider A an arena.*
*A **non-alternating strategy** $\sigma : A$, or **$\circlearrowleft$-strategy** for short, is a prestrategy s.t.:*

$$\begin{array}{rl} \text{receptive:} & \forall s \in \sigma,\ sa^- \in \circlearrowleft\text{-}\mathsf{Pla\widehat{y}s}(A) \ \Rightarrow\ sa^- \in \sigma\,, \\ \text{courteous:} & \textit{for } sabt \in \sigma \textit{ with } \mathrm{pol}(a) = + \textit{ or } \mathrm{pol}(b) = -, \\ & \textit{then if } sbat \in \circlearrowleft\text{-}\mathsf{Pla\widehat{y}s}(A),\ sbat \in \sigma \textit{ as well.} \end{array}$$

where it is understood that $sabt$ and $sbat$ have the same pointers.

The earliest appearance of such a saturation that we are aware of is by Selinger [Selinger, 1997]. It seemed to have first appeared in game semantics in [Laird, 2001b], and was called *saturation* in [Ghica and Murawski, 2008]. We refer to it as *courtesy*, a terminology imported from [Melliès and Mimram, 2007]: this is for compatibility with the terminology in the forthcoming concurrent games, where the same behaviour is imposed causally rather than by a saturation condition.

It is notable that non-alternating game semantics is obtained "only" by changing the nature of plays: the arenas, representing types, will remain the same.

5.1.3 Single-Threadedness

Are all non-alternating strategies sensible *w.r.t.* $\mathsf{IA}_{/\!/}$?

It is clear that it is not the case: the same phenomenon as in Section 3.3.1 occurs. As in the alternating case, we must impose a condition ensuring that the state is local, in that no information flows between distinct instances of the program. By analogy with the alternating case, we call this *single-threadedness*. But we must warn the reader that this by no means implies that our strategies can only have a single "thread" as in concurrent programming! *Single-threadedness* means that each initial move spawns an independent copy of this program, even if that program is multi-threaded.

But single-threadedness cannot be imported transparently from alternating strategies. Though the *current thread* (Definition 3.3.1) seemingly still makes sense, Player may

have control over several threads simultaneously, as in the play on **B** displayed below

$$\mathbf{q}^- \quad \mathbf{q}^- \quad \mathbf{tt}^+ \quad \mathbf{tt}^+$$

which will belong to the strategy for the constant **tt**. The last move fails condition *well-threaded* of Definition 3.3.2 even though **tt** does behave independently on all copies.

So we have to approach the issue a bit differently. First, define:

Definition 5.1.6. *A **non-alternating thread** on arena A is $t \in \circlearrowleft\text{-}\mathsf{Pla}\widehat{\mathsf{y}}\mathsf{s}(A)$ with at most one initial move; write $\circlearrowleft\text{-}\mathsf{Pla}\widehat{\mathsf{y}}\mathsf{s}_\bullet(A)$ for the set of non-alternating threads on A.*

Single-threaded behaviour is defined first via single threads, which invites:

Definition 5.1.7. *Consider A an arena.*
*A **thread-strategy** $\sigma : A$ is a set $\sigma \subseteq \circlearrowleft\text{-}\mathsf{Pla}\widehat{\mathsf{y}}\mathsf{s}_\bullet(A)$ satisfying:*

$$\text{receptive:} \quad \forall s \in \sigma,\ sa^- \in \circlearrowleft\text{-}\mathsf{Pla}\widehat{\mathsf{y}}\mathsf{s}_\bullet(A) \ \Rightarrow\ sa^- \in \sigma\,,$$

as well as non-empty, prefix-closed *and* courteous *as in Definitions 5.1.4 and 5.1.5.*
*As in Definition 5.1.4, a **thread-prestrategy** is a prestrategy $\sigma \subseteq \circlearrowleft\text{-}\mathsf{Pla}\widehat{\mathsf{y}}\mathsf{s}_\bullet(A)$.*

Note that unlike in Definition 5.1.4, *receptive* must be restricted to threads.

Given a thread-strategy $\sigma : A$, a full non-alternating strategy may be defined by interleaving copies of σ. Given $s, t \in \circlearrowleft\text{-}\mathsf{Pla}\widehat{\mathsf{y}}\mathsf{s}(A)$, $s \sqcup t \subseteq \circlearrowleft\text{-}\mathsf{Pla}\widehat{\mathsf{y}}\mathsf{s}(A)$ denotes the set of **interleavings** of s and t. If $X, Y \subseteq \circlearrowleft\text{-}\mathsf{Pla}\widehat{\mathsf{y}}\mathsf{s}(A)$, $X \sqcup Y$ is defined as the union of all $s \sqcup t$, for $s \in X$ and $t \in Y$. Finally, if $X \subseteq \circlearrowleft\text{-}\mathsf{Pla}\widehat{\mathsf{y}}\mathsf{s}(A)$, we set

$$\begin{array}{rcl} X^{(0)} & = & \{\varepsilon\} \\ X^{(n+1)} & = & X \sqcup X^{(n)} \end{array}$$

and $X^! \subseteq \circlearrowleft\text{-}\mathsf{Pla}\widehat{\mathsf{y}}\mathsf{s}(A)$, the **iterated interleaving** of X, is $X^! = \cup_{n \in \mathbb{N}} X^{(n)}$. We have:

Definition 5.1.8. *Consider A an arena, and $\sigma : A$ a non-alternating strategy on A.*
*We say that σ is **single-threaded** if there is $\tau : A$ a thread-strategy such that $\sigma = \tau^!$.*

In that case τ is uniquely determined, as it must be the set of threads in σ. If $\sigma : A$ is a non-alternating strategy on A, write $\sigma^\bullet$ for its set of threads – σ is single-threaded iff $\sigma = (\sigma^\bullet)^!$. It also directly follows from Definition 5.1.8 that we have:

Lemma 5.1.9. *Consider $\sigma, \tau : A$ single-threaded non-alternating strategies.*
Then, $\sigma = \tau$ if and only if $\sigma^\bullet = \tau^\bullet$.

5.1.4 Full Abstraction for $\mathtt{IA}_{/\!/}$

As we shall spell out in Section 5.2, arenas and single-threaded non-alternating strategies form a cartesian closed category $\circlearrowleft$-**Strat** which supports the interpretation of $\mathtt{IA}_{/\!/}$. More precisely, types and contexts are interpreted as arenas exactly as in Chapter 4 ($\mathtt{IA}_{/\!/}$ brings no new type), and a term $\Gamma \vdash M : A$ is interpreted as a morphism

$$[\![M]\!] \in \circlearrowleft\text{-}\mathbf{Strat}([\![\Gamma]\!], [\![A]\!])\,.$$

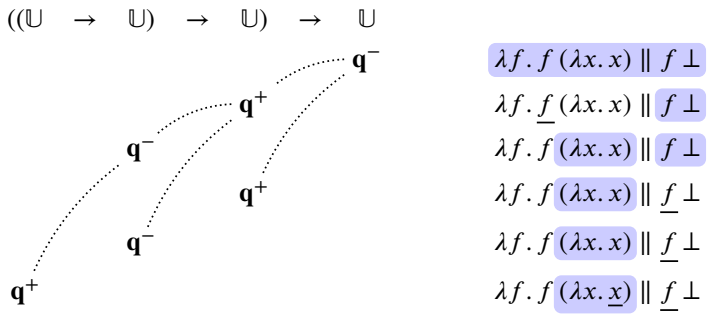


Figure 5.8: A realizable non P-visible play

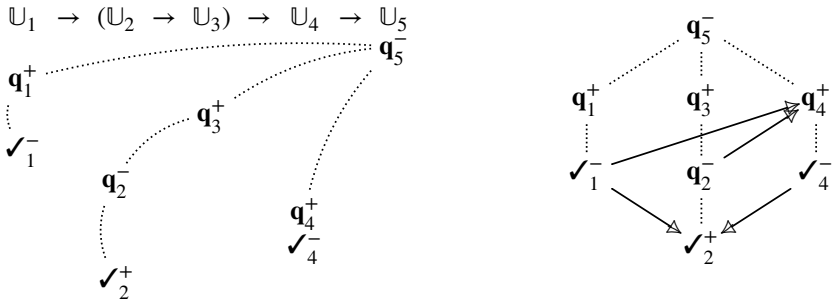


Figure 5.9: Example of definability of plays in $\mathsf{IA}_{/\!/}$

Realizing plays. But do we have more than $\mathsf{IA}_{/\!/}$? It is natural to expect that the answer is yes: in the alternating setting, *P-visibility* is required to ban higher-order references (see Section 3.3.2), and we have not imposed any such condition here.

But P-visibility is not adequate: strategies from $\mathsf{IA}_{/\!/}$ will not satisfy it, even though we only have ground type references. Figure 5.8 illustrates why: several threads in parallel may cause executions which are undistinguishable, viewed from the outside, from a non-local control flow due to a higher-order reference. In Figure 5.8, one cannot observe that the occurrence of the last move after the next-to-last is entirely coincidental, in that the two moves really belong to completely independent threads.

This phenomenon is general: given a play s, it is fairly easy to produce a term that can realize s, though typically along with many other plays. We can do this via a purely parallel term that simply mimics the hierarchical dependencies imposed by justification pointers. For instance, for s on the left hand side of Figure 5.9, one obtains the term

$$\lambda x^{\mathbb{U}_1} f^{\mathbb{U}_2 \to \mathbb{U}_3} y^{\mathbb{U}_4}. (x \parallel f\ \mathbf{skip} \parallel y);\ \bot. \tag{5.1}$$

Definability. Now, can we produce a term that realizes *exactly* s, and not more? Not in the strictest sense, since strategies interpreting terms will always be sets of plays satisfying Definition 5.1.5. But there is always a *smallest* strategy containing s:

Lemma 5.1.10. *Consider $\sigma : A$ a thread-prestrategy.*
Then, there is a unique least (for inclusion) thread-strategy $\widehat{\sigma} : A$ s.t. $\sigma \subseteq \widehat{\sigma}$. Its plays are deduced from σ by applying receptive *and* courteous *from Definition 5.1.5.*

Regarding $s \in \circlearrowleft\text{-}\widehat{\mathsf{Plays}}_\bullet(A)$ as a thread-prestrategy implicitely by prefix-closure, we have $\widehat{s} : A$ its **saturation**, a thread-strategy, and $\widehat{s}^! : A$ a single-threaded strategy.

The saturation process generates many plays, which are however succinctly summarized via a causal dependency between moves: a move depends on another when they *cannot be permuted by courtesy*. This happens if one points to the other; or for a Player move appearing *after* an Opponent move in s. It is helpful conceptually to represent these constraints via a diagram on moves, and this is done on the rhs of Figure 5.9: we draw the dependency due to pointers as dotted lines (read from top to bottom), and the dependency from negative moves to positive moves as arrows $\multimap\!\!\triangleright$. The resulting object is close to a concurrent strategy in the sense of Part II – by design, $\widehat{s}$ comprises exactly the plays that are linearizations of this diagram (with pointers following dotted lines).

Now, we observe that in our example, the purely parallel term of (5.1) captures *exactly* the static/dotted dependencies on the rhs of Figure 5.9. To capture $\widehat{s}$, we must also account for the arrows $\multimap\!\!\triangleright$, and this is done by exploiting the memory. For that we define two helper functions. If $M : \mathbb{V}$, we write $\mathbf{set}(M) : \mathbb{U}$ for $M := 1$, and $\mathbf{test}(M) : \mathbb{U}$ for $\mathbf{if}\ (\mathbf{iszero}\ !M)\ \bot\ \mathbf{skip}$ which converges iff $!M$ is non-zero. Then:

$$\lambda x^{\mathbb{U}_1} f^{\mathbb{U}_2 \to \mathbb{U}_3} y^{\mathbb{U}_4}. \left(\begin{pmatrix} x; \\ \mathbf{set}(\checkmark_1^-) \end{pmatrix} \parallel \begin{pmatrix} f\ (\mathbf{set}(\mathbf{q}_2^-); \\ \mathbf{test}(\checkmark_4^-); \\ \mathbf{grab}(\checkmark_2^+); \\ \mathbf{skip}) \end{pmatrix} \parallel \begin{pmatrix} \mathbf{test}(\checkmark_1^-); \\ \mathbf{test}(\mathbf{q}_2^-); \\ y; \\ \mathbf{set}(\checkmark_4^-) \end{pmatrix} \right); \bot$$

borrows the shape of (5.1), additionally signaling the $\multimap\!\!\triangleright$-links through memory (we omit reference and semaphore creations wrapping the term). We use one fresh reference (initialized to 0) for each Opponent move, which gets set to 1 when the Opponent move occurs. Finally, we must ensure that Opponent replications does not cause a duplication of Player moves by prompting re-evaluation of the corresponding subterms, so that we only obtain linearizations of the diagram on the right hand side of Figure 5.9. For that we add semaphores which are grabbed, but never released.

Done systematically for arbitrary plays [Ghica and Murawski, 2008], we get:

Proposition 5.1.11. *Consider A an arena, and $s \in \circlearrowleft\text{-}\widehat{\mathsf{Plays}}_\bullet(A)$.*
Then, there is a term $\vdash M : A$ of $\mathtt{IA}_{/\!/}$ such that $[\![M]\!] = \widehat{s}^!$.

From this, it is deduced in [Ghica and Murawski, 2008] that we have:

Theorem 5.1.12. *Arenas and single-threaded strategies form an intensionally fully abstract model for $\mathtt{IA}_{/\!/}$. Moreover, given $\Gamma \vdash M, N : A$, we have*

$$M \simeq N \quad \Leftrightarrow \quad \mathsf{comp}([\![M]\!]) = \mathsf{comp}([\![N]\!])\,,$$

where $\mathsf{comp}(\sigma)$ *is the set of* ***non-alternating complete plays****,* i.e. *non-alternating plays such that all questions have an answer.*

5.2 The Ambiant Cartesian Closed Category ↺-Strat

As in the alternating case, we now focus on the compositional construction of non-alternating strategies from terms, starting with the categorical structure.

5.2.1 Constructing ↺-Strat

As for alternating strategies, a **non-alternating strategy from** A **to** B is a non-alternating strategy on $A \Rightarrow B$. The composition of non-alternating strategies largely follows the same route as for alternating strategies in Section 4.1.2, with a few minor changes.

Interactions. Consider $\sigma : A \Rightarrow B$ and $\tau : B \Rightarrow C$ non-alternating strategies. As in Section 4.1.2, we start with an appropriate notion of *interactions*:

Definition 5.2.1. *Consider A, B and C arenas.*

A ***non-alternating interaction*** *on A, B, C is a legal pointing sequence on $(A \Rightarrow B) \Rightarrow C$ satisfying the conditions* fork *and* join *of Definition 5.1.2.*

We write $\circlearrowleft\text{-}\mathtt{I}(A, B, C)$ the set of all non-alternating interactions on A, B, C.

It is useful to compare with Definition 4.1.6 in the alternating case: the conditions *outer-legal*, *left-legal* and *right-legal*, which amount to all restrictions being alternating, become unnecessary: all restrictions are automatically valid non-alternating plays.

We use the same terminology and conventions as in the alternating case (notably regarding the polarities of moves, see below Definition 4.1.6) for non-alternating interactions. However, the state diagram of Figure 4.2 disappears. Just as plays, non-alternating interactions have no well-defined state, and both players can play anywhere at any time provided the move they play is justified and compatible with well-bracketing.

Composition. We now define composition, first for *prestrategies*:

Definition 5.2.2. *Consider $\sigma : A \Rightarrow B$ and $\tau : B \Rightarrow C$ non-alternating prestrategies.*

An ***interaction*** *between σ and τ is any $u \in \circlearrowleft\text{-}\mathtt{I}(A, B, C)$ such that $u \upharpoonright A, B \in \sigma$ and $u \upharpoonright B, C \in \tau$. We write $\sigma \parallel \tau$ for the set of all interactions between σ and τ.*

We may then define the ***composition*** $\tau \odot \sigma = \{u \upharpoonright A, C \mid u \in \sigma \parallel \tau\} : A \Rightarrow C$.

As in the alternating case, if $s \in \tau \odot \sigma$ then by definition there must be some $u \in \sigma \parallel \tau$ such that $s = u \upharpoonright A, C$, called a **witness** for s. However, unlike in the alternating case, the witness is no longer unique. Composition is associative:

Proposition 5.2.3. *Consider $\sigma : A \Rightarrow B, \tau : B \Rightarrow C$ and $\nu : C \Rightarrow D$ prestrategies.*

Then, $(\nu \odot \tau) \odot \sigma = \nu \odot (\tau \odot \sigma)$.

Proof. Same proof as in the alternating case: one defines the set $\mathtt{IA}(A, B, C, D)$ of ternary interactions as comprising well-bracketed legal pointing sequence on $((A \Rightarrow$

$B) \Rightarrow C) \Rightarrow D$. Then, one proves a "zipping lemma" with almost the same statement as Lemma 4.1.10 – with only the uniqueness clause missing – and the proof follows transparently as in Proposition 4.1.11 (uniqueness of witness is not needed). □

Note that associativity holds already for composition of prestrategies, without relying on *receptivity* or *courtesy*. However, we will need these conditions shortly.

We mention the following preservation proposition [Ghica and Murawski, 2008]:

Proposition 5.2.4. *Consider $\sigma : A \Rightarrow B$ and $\tau : B \Rightarrow C$ prestrategies. Then:*

if σ and τ are strategies, then so is $\tau \odot \sigma$,
if σ and τ are single-threaded, then so is $\tau \odot \sigma$.

Copycat. Composition of non-alternating strategies is as direct as in the alternating case – perhaps simpler, as interactions are "wilder", not having to follow the rigid structure imposed by alternation and the state diagram. But non-alternation also means that identities cannot be as simple as in Definition 4.1.12: copycat must handle Opponent moves arriving in bulk, possibly before earlier Opponent moves are propagated.

Nevertheless, the non-alternating copycat can be succinctly defined by:

Definition 5.2.5. *Consider A any arena, and $\mathbf{cc}^{\bullet}_A$ the set of* threads *of $\mathbf{cc}_A$.*

*The **non-alternating copycat** on A, written $\circlearrowleft$-$\mathbf{cc}_A$, is simply defined as $\widehat{\mathbf{cc}^{\bullet}_A}^{!}$.*

So $\circlearrowleft$-$\mathbf{cc}_A$ is the smallest non-alternating strategy including the alternating copycat. Despite the overload of notations, we will often still refer to the non-alternating copycat simply with $\mathbf{cc}_A$ – hopefully, the context should always be sufficient to disambiguate. We shall reserve $\circlearrowleft$-$\mathbf{cc}_A$ for when we need to be extra explicit. Likewise, we shall use $\downuparrows$-$\mathbf{cc}_A$ for the alternating copycat, when we feel it is useful to emphasize that this is not $\circlearrowleft$-$\mathbf{cc}_A$.

This definition is succinct but not so explicit, so we illustrate it with a play of the non-alternating copycat in Figure 5.2.1. As long as Opponent "plays in turn", copycat remains alternating. But if Opponent plays several moves in a row – before the previous ones can be propagated – then copycat maintains a buffer of moves to be forwarded and plays them (not necessarily in the same order as they arrived), attempting to restore the same state between the right hand side and the left hand side.

As required, the non-alternating copycat is neutral for composition with respect to non-alternating strategies. In fact, we have the following proposition:

Proposition 5.2.6. *Consider A, B arenas, and $\sigma : A \Rightarrow B$ a prestrategy.*
Then, σ is a non-alternating strategy iff $\circlearrowleft$-$\mathbf{cc}_B \odot \sigma = \sigma \odot \circlearrowleft$-$\mathbf{cc}_A = \sigma$.

Proof. Direct adaptation of [Ghica and Murawski, 2008, Lemma 16]. □

It follows that we have:

Corollary 5.2.7. *There is a category $\circlearrowleft$-**Strat** with arenas as objects, and as morphisms from A to B all single-threaded strategies on $A \Rightarrow B$.*

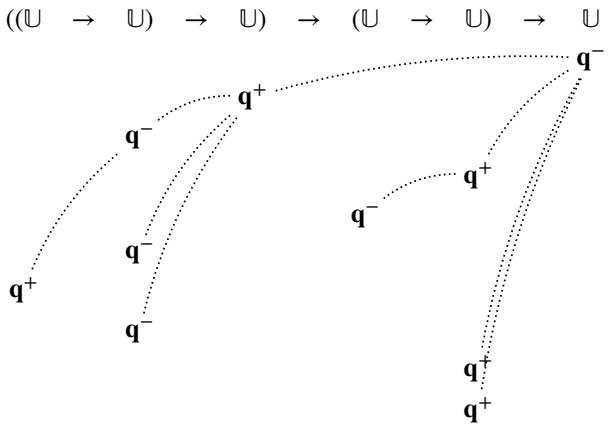


Figure 5.10: An example play of the non-alternating copycat

5.2.2 Cartesian Closed Structure

The cartesian closed structure essentially follows that of ↓↑-**Strat**.

Cartesian structure. On objects, the cartesian structure is defined as in ↓↑-**Strat**: the empty arena 1 with no moves is a terminal object, and the product of two arenas A and B is $A \times B$ as defined in Definition 4.1.1. For arenas A, B, the projections

$$\circlearrowleft\text{-}\pi_A : A \times B \Rightarrow A\,, \qquad \circlearrowleft\text{-}\pi_B : A \times B \Rightarrow B$$

are the copycat strategies defined with the obvious adaptation of Definition 5.2.5.

To define the pairing, we prove:

Lemma 5.2.8. *Consider Γ, A, B arenas, and $\sigma : \Gamma \Rightarrow A$, $\tau : \Gamma \Rightarrow B$ single-threaded non-alternating strategies. Then, there exists a single-threaded non-alternating*

$$\langle \sigma, \tau \rangle : \Gamma \Rightarrow A \times B\,,$$

whose set of threads is (up to relabeling) the disjoint union of the threads of σ and τ.

Proof. Straightforward by definition of single-threaded non-alternating strategies. □

Again, this data satisfies the required equations for cartesian products:

Lemma 5.2.9. *Consider $\sigma : \Gamma \Rightarrow A$, $\tau : \Gamma \Rightarrow B$, and $\nu : \Delta \Rightarrow \Gamma$ single-threaded. Then, the following equations hold:*

$$\begin{aligned}
\circlearrowleft\text{-}\pi_A \odot \langle \sigma, \tau \rangle &= \sigma \\
\circlearrowleft\text{-}\pi_B \odot \langle \sigma, \tau \rangle &= \tau \\
\langle \circlearrowleft\text{-}\pi_A, \circlearrowleft\text{-}\pi_B \rangle &= \circlearrowleft\text{-}\mathfrak{a}_{A \times B} \\
\langle \sigma, \tau \rangle \odot \nu &= \langle \sigma \odot \nu, \tau \odot \nu \rangle\,.
\end{aligned}$$

Proof. The first two equalities are a variation of Proposition 5.2.6, first proved for threads and extended to strategies via Lemma 5.1.9. The last two are direct on threads and generalized by Lemma 5.1.9. □

It follows from these equations that we have the desired structure:

Corollary 5.2.10. *The category* $\circlearrowleft$-**Strat** *is cartesian.*

Cartesian closed structure. Now, we describe the cartesian closure.

On arenas, $A \Rightarrow B$ is of course the same construction as for alternating strategies. We have seen in Lemma 4.1.18 that there is an isomorphism of arenas

$$\Xi_{\Gamma,A,B} \quad : \quad (\Gamma \times A) \Rightarrow B \quad \cong \quad \Gamma \Rightarrow (A \Rightarrow B)$$

which, as in the alternating case, immediately yields a bijection:

Lemma 5.2.11. *Consider* Γ, A *and* B *three arenas. Then, there is a bijection*

$$\Lambda_{\Gamma,A,B} \quad : \quad \circlearrowleft\text{-}\mathbf{Strat}(\Gamma \times A, B) \quad \simeq \quad \circlearrowleft\text{-}\mathbf{Strat}(\Gamma, A \Rightarrow B)$$

preserving composition in the sense that

$$\Lambda_{\Gamma,A,B}(\sigma) \odot \gamma \quad = \quad \Lambda_{\Delta,A,B}(\sigma \odot (\gamma \times A))$$

for all $\sigma \in \circlearrowleft\text{-}\mathbf{Strat}(\Gamma \times A, B)$ *and* $\gamma \in \circlearrowleft\text{-}\mathbf{Strat}(\Delta, \Gamma)$.

Proof. A routine manipulation on interactions. □

In turn, this bijection lets us define the evaluation strategy simply as

$$\circlearrowleft\text{-}\mathsf{ev}_{A,B} \quad = \quad \Lambda^{-1}_{A,A,B}(\circlearrowleft\text{-}\mathbf{cc}_{A \Rightarrow B}) \quad \in \quad \circlearrowleft\text{-}\mathbf{Strat}((A \Rightarrow B) \times A, B)$$

and from the above, it becomes routine to prove cartesian closure:

Proposition 5.2.12. *The category* $\circlearrowleft$-**Strat** *is cartesian closed.*

Proof. As for Proposition 4.1.20. □

5.2.3 Recursion

As for alternating strategies, non-alternating strategies are partially ordered by inclusion. This partial order is preserved by all operations, yielding:

Proposition 5.2.13. *For any two arenas* A *and* B, $\circlearrowleft$-$\mathbf{Strat}(A, B)$ *is partially ordered by inclusion, yielding a pointed dcpo. All operations on strategies involved in the cartesian closed structure are continuous, making* $\circlearrowleft$-**Strat** *enriched over pointed dcpos.*

Proof. Direct verifications. □

As in the alternating case, we define

$$\begin{array}{rcccl} F & : & \circlearrowleft\text{-}\mathbf{Strat}(A \Rightarrow A, A) & \to & \circlearrowleft\text{-}\mathbf{Strat}(A \Rightarrow A, A) \\ & & \sigma & \mapsto & \lambda f^{A \Rightarrow A}.\, f\,(\sigma\, f) \end{array}$$

a continuous operation whose least fixed point

$$\circlearrowleft\text{-}\mathcal{Y}_A \quad = \quad \bigcup_{n \in \mathbb{N}} F^n(\bot) \quad \in \quad \circlearrowleft\text{-}\mathbf{Strat}(A \Rightarrow A, A) \tag{5.2}$$

satisfies $\circlearrowleft\text{-}\mathcal{Y}_A = F\,\circlearrowleft\text{-}\mathcal{Y}_A$ by construction and provides the interpretation of recursion.

5.3 Interpretation of $\mathsf{IA}_{/\!/}$

5.3.1 Interpretation of $\mathsf{PCF}_{/\!/}$

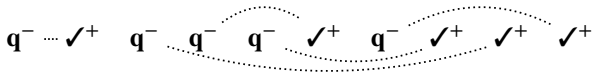
First of all, *types* and *contexts* of PCF are interpreted exactly as in $\Updownarrow$-**Strat**.

For *terms*, any $\Gamma \vdash M : A$ gets interpreted as a morphism

$$[\![M]\!] \in \circlearrowleft\text{-}\mathbf{Strat}([\![\Gamma]\!], [\![A]\!])\,.$$

As for $\Updownarrow$-**Strat**, the primitives for the simply-typed λ-calculus and for recursion are interpreted using the structure above. It remains to provide interpretations for all the additional combinators that PCF adds to the simply-typed λ-calculus.

PCF. Constants are interpreted by single-threaded strategies returning directly the corresponding value – due to non-alternation, this means they have plays such as



for skip : **U**, interleaving various calls in an asynchronous fashion – the generating thread-strategy is simpler to describe, with only one maximal play $\mathbf{q}^-\checkmark^+$.

The interpretation of PCF is obtained via the same clauses as in alternating games, in Figure 4.5. This uses single-threaded non-alternating strategies seq, if, succ, iszero and pred whose thread-strategies are exhaustively presented as comprising prefixes of plays of the form displayed in Figures 4.4 and 4.6. Non-alternation does not induce additional plays, because we required that non-alternating plays should be well-bracketed thus Opponent can answer at most once. The strategy let comprises more plays than its alternating version in Figure 4.7. We omit however its definition, obtained as an obvious simplification of the new operation that $\mathsf{PCF}_{/\!/}$ adds to PCF: the parallel let.

$\mathsf{PCF}_{/\!/}$. To complete the interpretation of $\mathsf{PCF}_{/\!/}$, we need a single-threaded strategy

$$\mathsf{plet}_{\mathbf{X},\mathbf{Y}} \qquad : \qquad \mathbf{X} \Rightarrow \mathbf{X} \Rightarrow (\mathbf{X} \Rightarrow \mathbf{X} \Rightarrow \mathbf{Y}) \Rightarrow \mathbf{Y}$$

for the parallel let. Intuitively, its behaviour is simple enough: upon being called, plet will interrogate its first two arguments in parallel. Upon receiving two values, it will

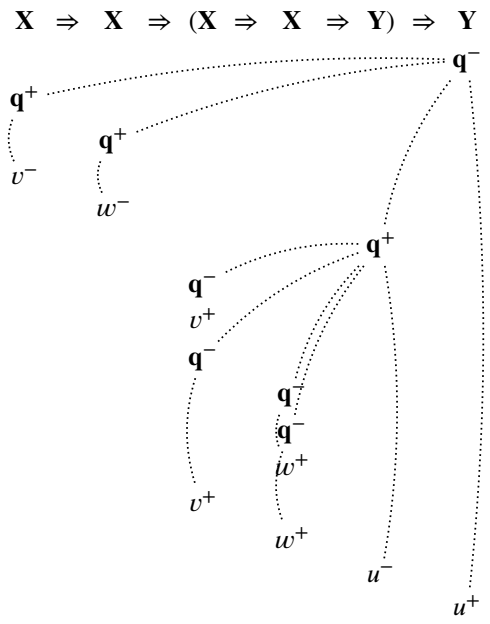


Figure 5.11: Typical play of $\mathsf{plet}_{\mathbf{X},\mathbf{Y}}$.

interrogate its function argument and provide it the two values. We show one of its typical plays, in Figure 5.11. It acts in two subsequent stages: first the parallel evaluation of arguments, and then the call to the functional argument with the memoized values.

To define $\mathsf{plet}_{\mathbf{X},\mathbf{Y}}$ formally, we define sets of pointing sequences for these two stages. Annotating types as $\mathbf{X}_1 \Rightarrow \mathbf{X}_2 \Rightarrow (\mathbf{X} \Rightarrow \mathbf{X} \Rightarrow \mathbf{Y}) \Rightarrow \mathbf{Y}$, for v and w values of $\mathbf{X}$, we set

$$\begin{aligned}\mathsf{force}_{v,w} \quad ::= \quad & \mathbf{q}_1^+ v_1^- \mathbf{q}_2^+ w_2^- \mid \mathbf{q}_2^+ w_2^- \mathbf{q}_1^+ v_1^- \mid \mathbf{q}_1^+ \mathbf{q}_2^+ v_1^- w_2^- \\ & \mid \mathbf{q}_1^+ \mathbf{q}_2^+ w_2^- v_1^- \mid \mathbf{q}_2^+ \mathbf{q}_1^+ v_1^- w_2^- \mid \mathbf{q}_2^+ \mathbf{q}_1^+ w_2^- v_1^-\end{aligned}$$

where it is understood that v_1^- points to $\mathbf{q}_1^+$ and w_2^- points to $\mathbf{q}_2^+$.

Next, for the second stage we define another set of pointing sequences, written $\mathsf{eval}_{v,w}$. Its elements are obtained by first considering the threads in the strategy

$$\lambda x^{\mathbf{X}}.\, \lambda y^{\mathbf{X}}.\, \lambda f^{\mathbf{X}\Rightarrow\mathbf{X}\Rightarrow\mathbf{Y}}.\, f\, v\, w : \mathbf{X} \Rightarrow \mathbf{X} \Rightarrow (\mathbf{X} \Rightarrow \mathbf{X} \Rightarrow \mathbf{Y}) \Rightarrow \mathbf{Y}\,,$$

defined exploiting the cartesian closed structure of $\mathcal{O}$-**Strat** and the corresponding internal language. From such threads, elements of $\mathsf{eval}_{v,w}$ are obtained by removing the initial question. Finally, we form the set of pointing sequences

$$\mathsf{prelet}_{\mathbf{X},\mathbf{Y}} \quad = \quad \bigcup_{v,w} \mathbf{q}^- \cdot \mathsf{force}_{v,w} \cdot \mathsf{eval}_{v,w}$$

completed into a set of plays by reinstating the missing pointers in the unique possible way, with missing pointers assigned to the initial move $\mathbf{q}^-$. Finally, the non-alternating thread-strategy for $\mathsf{plet}_{\mathbf{X},\mathbf{Y}}$ is obtained as interleavings of all prefixes of $\mathsf{prelet}_{\mathbf{X},\mathbf{Y}}$.

Altogether, we obtain $\mathsf{plet}_{\mathbf{X},\mathbf{Y}} : \mathbf{X} \Rightarrow \mathbf{X} \Rightarrow (\mathbf{X} \Rightarrow \mathbf{X} \Rightarrow \mathbf{Y}) \Rightarrow \mathbf{Y}$ and we complete the interpretation of $\mathsf{PCF}_{/\!/}$ by adding to Figure 4.4 the additional clause

$$[\![\Gamma \vdash \mathbf{let} \begin{pmatrix} x_1 & = & N_1 \\ x_2 & = & N_2 \end{pmatrix} \mathbf{in}\ M : \mathbb{Y}]\!] = \mathsf{plet}_{\mathbf{X},\mathbf{Y}} \odot_! \langle [\![N_1]\!], [\![N_2]\!], \Lambda^!(\Lambda^!([\![M]\!]))\rangle$$

5.3.2 Interpretation of State

The types $\mathbb{V}$ and $\mathbb{S}$ are interpreted by the same arenas as in Section 4.2.2.

Queries. As in the alternating case, we set the interpretation of queries as

$$\begin{array}{rcl} [\![M := N]\!] & = & \mathsf{assign} \odot \langle [\![N]\!], [\![M]\!]\rangle \\ [\![!M]\!] & = & \mathsf{deref} \odot [\![M]\!] \\ [\![\mathbf{grab}(M)]\!] & = & \mathsf{grab} \odot [\![M]\!] \\ [\![\mathbf{release}(M)]\!] & = & \mathsf{release} \odot [\![M]\!] \end{array}$$

using the single-threaded non-alternating strategies with threads as in Figures 4.11, 4.12, 4.13, and 4.14 – the same basic plays as in the alternating case.

Declaration. Now, we must provide the interpretation for the declaration of new references and semaphores. For that, we need non-alternating versions of the strategies for reference and semaphore declaration introduced in Section 4.2.2.

Rather than going again through the concrete definition of these strategies, we set

$$\circlearrowleft\text{-}\mathsf{newr}_{\mathbf{X}} = \widehat{\mathsf{newr}^{\bullet}_{\mathbf{X}}}^{!}, \qquad \circlearrowleft\text{-}\mathsf{news}_{\mathbf{X}} = \widehat{\mathsf{news}^{\bullet}_{\mathbf{X}}}^{!}$$

via the saturation process of Lemma 5.1.10, and use those in:

$$\begin{array}{rcl} [\![\mathbf{newref}\ x := n\ \mathbf{in}\ M]\!] & = & \circlearrowleft\text{-}\mathsf{newr}_{\mathbf{X}} \odot \Lambda([\![M]\!]) \ \in\ \circlearrowleft\text{-}\mathbf{Strat}([\![\Gamma]\!], [\![A]\!]) \\ [\![\mathbf{newsem}\ x := n\ \mathbf{in}\ M]\!] & = & \circlearrowleft\text{-}\mathsf{news}_{\mathbf{X}} \odot \Lambda([\![M]\!]) \ \in\ \circlearrowleft\text{-}\mathbf{Strat}([\![\Gamma]\!], [\![A]\!])\,. \end{array}$$

Bad variables and semaphores. As in the alternating case, through the isomorphisms $\mathbf{V} \cong (\&_{n\in\mathbb{N}}\mathbf{U})\&\mathbf{N}$ and $\mathbf{S} \cong \mathbf{U}\&\mathbf{U}$, the interpretation is:

$$[\![\mathbf{mkvar}\ M\ N]\!] = \langle \langle [\![M]\!]\, n \mid n \in \mathbb{N}\rangle, [\![N]\!]\rangle\,, \qquad [\![\mathbf{mksem}\ M\ N]\!] = \langle [\![M]\!], [\![N]\!]\rangle\,,$$

which concludes the interpretation of $\mathsf{IA}_{/\!/}$.

5.3.3 Summary of Results

This concludes the construction of the cartesian closed category $\circlearrowleft$-**Strat**, as well as the interpretation of $\mathsf{IA}_{/\!/}$. As in the alternating case, we conclude the technical content of this chapter by summing up a few key results for future reference.

Proposition 5.3.1. *There is an adequate interpretation of* $\mathsf{IA}_{/\!/}$ *in* $\circlearrowleft$-**Strat**.

Proof. The interpretation is as described in its chapter. Computational adequacy corresponds to the particular case of Theorem 5.1.12 instantiated on ground type. □

Theorem 5.3.2. *The interpretation of Proposition 5.3.1 refines into an interpretation of* IA *in* $\circlearrowleft$-**Strat**, *which is intensionally fully abstract.*

Proof. First stated as Theorem 5.1.12. □

5.4 Conclusions and Historical Notes

The first to consider a non-alternating version of play-based game semantics was Laird, who proposed a fully abstract model for a message-passing language called Idealized CSP [Laird, 2001b]. This model had much in common with the one presented here, however plays are equipped with additional "concurrency pointers", which explicitly mark the different threads acting in a non-alternating plays. Ghica and Murawski's model was first published in 2004 [Ghica and Murawski, 2004], in conference format. It is strongly related to the non-alternating games of Melliès and Mimram published soon after [Melliès and Mimram, 2007], which contains the definition of a category of games and linear non-alternating strategies, along with an adequate notion of innocence.

Just like in the alternating case in the presence of state, non-alternating games form an effectively presentable fully abstract model, which lead to decision algorithms for observational equivalence – though for very restricted languages: first bounding the number of threads via a type system [Ghica et al., 2006] so that strategies are representable as finite automata, and more recently via other finitary restrictions that allow a representation of non-alternating strategies via more elaborate notions of data automata [Dixon et al., 2021a, Dixon et al., 2021b].

It is noteworthy, in the definability process outlined in Proposition 5.1.11, that it seems we really *need* semaphores in order to capture a single play. Murawski explored what happened without semaphores [Murawski, 2010]; he proved that then strategies are closed under a stuttering behaviour that affect observational equivalence. This shows that, perhaps surprisingly, semaphores are *not* definable just via references up to observational equivalence – another of those very results which illustrates well the strength of game semantics.

This concludes the preliminary part. One of the purposes of the rest of this monograph will be to construct and explore $\multimap$-**Strat**, a *causal* version of $\circlearrowleft$-**Strat**.

Part II

Thin Concurrent Games

Introduction to Part II

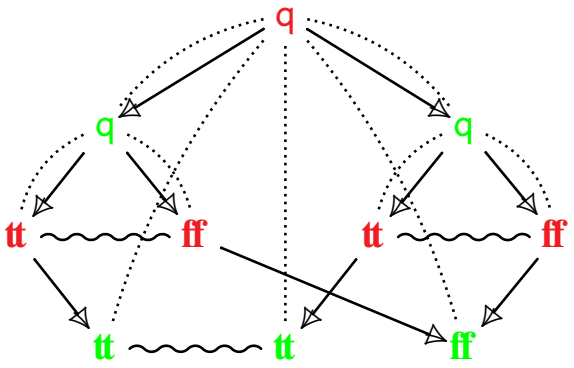


Figure 5.12: To survive pointer showers, remember to bring your parallel-or umbrella!

Now that we have spent significant time building up a background on traditional play-based game semantics, let us forget it all and start from scratch!

Concurrent games reject the premise that there exists a global clock; instead relying on a *causal* presentation of program behaviour. This idea was pioneered by Abramsky and Melliès [Abramsky and Melliès, 1999], and its consequences were mainly explored by Melliès in his sequence of insightful papers on *asynchronous games* [Melliès, 2003, Melliès, 2004a, Melliès, 2004b, Melliès, 2005, Melliès and Mimram, 2007]. Inheriting from this, the definitions on which we build were first proposed by Rideau and Winskel [Rideau and Winskel, 2011], also building on other work [Faggian and Hyland, 2002, Curien and Faggian, 2005, Faggian and Piccolo, 2009] inspired by *ludics* – we rediect to Section 6.5 for a longer discussion on the history of these notions.

While thin concurrent games share much conceptually with the game semantics of Part I, they build on a completely different – and more sophisticated – technical basis: the theory of event structures. This enhances their expressiveness, but with the cost of an involved technical development, for two reasons.

Firstly, event structures are more elaborate than the usual objects used in game semantics, such as sequences of moves. Manipulating them rigorously is more complex than for plays, involving a demanding – but (perhaps subjectively) rather elegant – infrastructure involving significant mathematical engineering and ingenuity.

Secondly, we cannot get away with some of the usual informal practice of traditional game semantics. Indeed, researchers in game semantics have sometimes adopted a somewhat informal mathematical style – the strategy does this, if Opponent plays this then the strategy reacts by playing that, *etc*. While this has been criticized, it is true that such phrasings in the literature can usually be unambiguously made formal if needed (though in practice this is rarely done); but this is thanks to the fact that the mathematical objects involved are rather simple and closely follow their natural language description. So it is both the curse and the blessing of concurrent games that when strategies are event structures, such informal descriptions are no longer reasonable.

So, we must prepare the reader for a long mathematical development, in part because of the complexity of our strategies, because the development is carried out with more mathematical precision than is the standard in many game semantics texts, but also because we have taken the space to make the development as pedagogical as possible.

The purpose of Part II is the construction of so-called "thin concurrent games" along with the categorical structure needed to interpret the simply-typed λ-calculus, leaving for Part III the interpretation of $\mathsf{IA}_{/\!/}$ and its fragments. We consider that the key contribution of Part II is the bicategory **TCG**, a setting for strategies-as-event-structures able to handle symmetry and the interchangeability of moves as required for copy indices and uniformity (in the sense of Section 3.4). We arrive there in two stages: first, in Chapter 6 we develop the bicategory **CG** of concurrent games *without symmetry*; then Chapter 7 introduces **TCG**, a conservative extension of **CG** equipping it to handle symmetry. Finally, **TCG** is only a bicategory and as such does not yet have the further categorical structure required to handle the interpretation of programming language; for that, in Chapter 8 we construct **NTCG**, essentially a model of intuitionistic linear logic with an exponential ! so that the Kleisli category $\mathbf{NTCG}_!$ is cartesian closed.

Chapter 6

Basic Concurrent Games

In this chapter, relying on the intuitions introduced previously for traditional game semantics, we introduce the bicategory **CG**, the foundation for concurrent games on event structures. We do this for pedagogical reasons: **CG** is only a stepping stone, as it does not yet support the exponential (!) – this will require the addition of *symmetry* in Chapter 7. But it seems better to first construct **CG** separately, so as to give the proper spotlight to quite a few new conceptual ideas, instead of introducing everything simultaneously.

CG is a non-deterministic extension of Melliès and Mimram's *non-alternating asynchronous games [Melliès and Mimram, 2007]*; it was first introduced by Rideau and Winskel in [Rideau and Winskel, 2011] (see also [Castellan et al., 2017a]). The presentation given here is new, exploiting insights obtained over the years, from our experience of working with concurrent games.

6.1 Games and Strategies as Event Structures

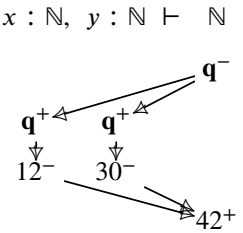
6.1.1 Basic Intuitions on Concurrent Strategies

Parallelism. In the beginning of Section 5.1.1, we motivated non-alternating strategies by considering the plays in a parallel implementation of the arithmetic sum

$$x : \mathbb{N}, y : \mathbb{N} \vdash \mathbf{let} \begin{pmatrix} x_1 & = & x \\ x_2 & = & y \end{pmatrix} \mathbf{in}\ x_1 + x_2 : \mathbb{N},$$

which we argued required to account for all the orders in which the evaluations of x and y could be performed by the scheduler. Rather than writing down all these interleavings,

an alternative could be to write – naively for now – a diagram of the form:

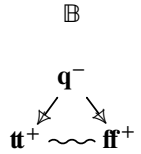


the Hasse diagram of a *partial order* in which the two sub-computations occur in independent branches. As for sequential plays in game semantics, this diagram is read from top to bottom: first, Opponent initiates computation with $\mathbf{q}^-$. But then, there are *two* follow-ups: the two occurrences of $\mathbf{q}^+$, prompting the evaluation of the two arguments. Despite them being in the same row, they are not thought of as simultaneous. Instead they are *independent*: there is no well-defined ordering between them. Concurrent strategies display explicitely this independence, whereas the non-alternating game semantics of Chapter 5 would convey it implicitely by allowing all interleavings.

In a diagram as above, the vertical axis no longer represents time. Instead, the partial order is thought of as expressing *causality*: in order to observe the two occurrences of $\mathbf{q}^+$, we *must* have seen $\mathbf{q}^-$. In order to observe (this occurrence of) the answer 42, we *must* have seen the two arguments converge with values 12 and 30.

Non-determinism. In such diagrams, the branching structure indicates parallelism, not non-determinism. But then, how is non-determinism going to be represented?

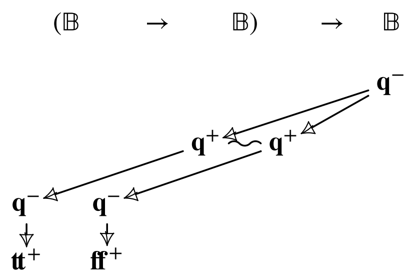
Our answer is to mark branchings that correspond to non-deterministic choice:



In this diagram, which represents the non-deterministic choice of a boolean, $\mathbf{tt}^+$ and $\mathbf{ff}^+$ are linked with a wiggly line indicating *conflict*: only one of them can be seen in a given execution. This marks that the branching is a *non-deterministic* rather than a *parallel* branching. Such situations with these two branching structures are captured by the mathematical notion of an *event structure*, introduced formally in Section 6.1.2.

Concurrent strategies explicitely display all non-deterministic choices, even when

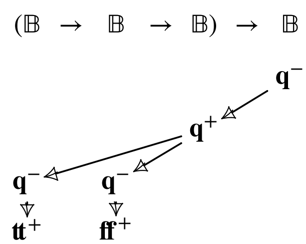
they do not yet yield visibly different behaviours. For instance, in the diagram



(6.1)

corresponding *e.g.* to the term $\lambda f^{\mathbb{B}\to\mathbb{B}}.\,(f\ \mathbf{tt}) \oslash (f\ \mathbf{ff})$ for $\oslash$ a non-deterministic choice operator, the two conflicting occurrences of $\mathbf{q}^+$ correspond to the *same* observable behaviour, but they should be kept separate because they have distinct futures. In general, concurrent strategies remember all non-deterministic choices even when they lead to no observable difference: $\mathbf{tt} \oslash \mathbf{tt}$ will *not* yield the same concurrent strategy as $\mathbf{tt}$.

Causality. Though we have motivated the use of partial order models for representing strategies by examples from parallel and non-deterministic computation, we shall see throughout this monograph that it has a deep impact even for pure sequential deterministic computation. As an example, the term $\lambda f.\, f\ \mathbf{tt}\ \mathbf{ff}$ admits the causal behaviour



where it is apparent that the two branches where Opponent explores the two arguments are *causally independent* from each other. In fact, in that case the tree structure coincides with the tree of *P-views* as described in Section 3.2.4, exposing its causal nature.

6.1.2 Formalizing Concurrent Strategies

Such diagrams as above, mixing causal dependency and conflict, are certain *event structures* – the partial order model at the core of concurrent games.

Event structures. We first define *event structures*[1]:

Definition 6.1.1. *An **event structure** (**es**) is a triple $E = (|E|, \leq_E, \#_E)$, where $|E|$ is a (countable) set of **events**, $\leq_E$ is a partial order called **causal dependency** and $\#_E$ is an*

[1] In the literature, those event structures are called *prime event structures with binary conflict*.

irreflexive symmetric binary relation on $|E|$ called ***conflict****, satisfying:*

$$\begin{array}{rl} \text{finite causes:} & \forall e \in E,\ [e]_E = \{e' \in E \mid e' \leq_E e\} \text{ is finite,} \\ \text{conflict inheritance:} & \forall e_1 \#_E e_2,\ \forall e_2 \leq_E e_2',\ e_1 \#_E e_2'\,. \end{array}$$

A given event has only finitely many causes: this fundamental axiom means that any event can appear in a finite execution. *Conflict inheritance* is sometimes called the *vendetta axiom*: conflicts transfer to children, and to their children, and so on for ever. It ensures that two non-conflicting events can always appear together in a finite execution (as the causal dependencies of non-conflict events must be non-conflicting).

The diagrams appearing earlier in this section may be read as event structures. For E an event structure, the **immediate causal dependency** $\rightarrowtriangle_E$ is defined as $e \rightarrowtriangle_E e'$ if $e <_E e'$ with no event in between: for all $e'' \in E$, $e \leq_E e'' \leq_E e'$ we have $e = e''$ or $e'' = e'$. Likewise, the **immediate conflict** $e \sim\!\!\!\sim_E e'$ means that $e \#_E e'$ and the conflict is not inherited: if $e'' <_E e$, $\neg(e'' \#_E e')$ and if $e'' <_E e'$, $\neg(e \#_E e'')$. These two notations are intensively used, in particular when drawing event structures. For instance, in the diagram (6.1), $\mathbf{tt}^+$ and $\mathbf{ff}^+$ are in conflict (as they depend on conflicting events), but not in immediate conflict – so they are not linked with a wiggly line.

The most fundamental definition on an event structure is *configurations* – one may regard an event structure as a concrete representation of its set of configurations:

Definition 6.1.2. *A (finite)* ***configuration*** *of event structure E is a finite $x \subseteq |E|$ s.t.:*

$$\begin{array}{rl} \text{down-closed:} & \forall e \in x,\ \forall e' \in E,\ e' \leq_E e \ \Rightarrow\ e' \in x \\ \text{consistent:} & \forall e, e' \in x,\ \neg(e \#_E e')\,. \end{array}$$

We write $\mathscr{C}(E)$ for the set of finite configurations on E.

Configurations are the *states* of an event structure: conflict-free finite sets of events with all causal dependencies satisfied. Though a configuration is merely a set, it inherits a partial order which is often drawn in pictures from that of the ambiant event structure.

The set $\mathscr{C}(E)$ is naturally ordered by inclusion; it is the *domain of configurations*. Configurations are typically ranged over by variables x, y, z. For $x, y \in \mathscr{C}(E)$, we write $x \multimap y$ if x is immediately below y in the inclusion order, *i.e.* there is $e \in E$ such that $e \notin x$ and $y = x \cup \{e\}$ – we also write $x \vdash_E e$ and say that x **enables** e.

Games. Before we formalize strategies, let us look at *games*.

As for arenas, our *games* present observable computational events, along with their dependencies and conflict. In spirit, the way this is done is very close to the arenas of Chapter 3. For instance, the games representing respectively $\mathbb{B}$ and $\mathbb{U}$ are:

$$\mathsf{B} \quad = \quad \begin{array}{ccc} & \mathbf{q}^- & \\ \cdots & & \cdots \\ \mathbf{tt}^+ & \sim\!\!\!\sim & \mathbf{ff}^+ \end{array} \qquad\qquad \mathsf{U} \quad = \quad \begin{array}{c} \mathbf{q}^- \\ \vdots \\ \checkmark^+ \end{array} \tag{6.2}$$

where the first is to be compared with the arena for booleans in traditional arena games in Figure 3.8. It is the same, except that the two answers are conflicting: unlike in

traditional game semantics, concurrent games are inherently linear (or rather, affine) and a question may be by default answered at most once. This hints at a deeper difference: in Figure 3.8, the initial question may appear arbitrarily many times in a play. In contrast, the intention here is that a given move can be played at most once: an execution such as at the beginning of Section 3.3.1 would make sense not on B but on its "bang"

$$!\mathsf{B} \quad = \quad \begin{array}{ccccccc} & \mathbf{q}^-_0 & & \mathbf{q}^-_1 & & \mathbf{q}^-_2 & \\ \mathbf{tt}^+_0 & \sim\sim & \mathbf{ff}^+_0 \quad \mathbf{tt}^+_1 & \sim\sim & \mathbf{ff}^+_1 \quad \mathbf{tt}^+_2 & \sim\sim & \mathbf{ff}^+_2 \end{array} \quad \ldots \tag{6.3}$$

involving *copy indices* as in AJM games (see Section 3.4) – but we push this operation aside until Chapter 7, as developing it properly also requires a model in which one can express that all copies are interchangeable, which plain event structures cannot do.

Games feature causal dependency and conflict, hence are *also* event structures:

Definition 6.1.3. *A **game** is an es $A = (|A|, \leq_A, \#_A)$ with a **polarity function***

$$\mathrm{pol}_A : |A| \to \{-, +\} .$$

The polarity function simply indicates whether a given move is due to Player (+) or Opponent (−). Though games are event structures, we draw them with specific conventions: in particular, as in (6.2) and following our earlier convention for arenas, immediate causal dependency in games is drawn with dotted lines rather than with the usual symbol $\rightarrowtriangle$. Having distinct notations for immediate causal dependency in games and strategies will allow us later on to have diagrams with two immediate dependencies superimposed, from a game and from a strategy.

In practice, games in the sense of Definition 6.1.3 are way too wild for our purposes. The games arising from the interpretation of types will have a very specific shape: they are negative (Opponent starts), alternating (immediate causal links are always between different players), forestial (there is no causal merge) and conflict is local (between two moves with the same predecessor, if any). But for the sake of economy, we shall impose such conditions only when they become technically required.

Prestrategies. If both strategies and games are to be event structures, what does it mean that an event structure "plays" on another event structure? It is natural to model such a situation as an event structure (the strategy) *labeled* by an event structure (the game), where the labelling function should be a *map of event structures*:

Definition 6.1.4. *Consider E, F two es. A **map of event structures** $f : E \to F$ is*

$$f : |E| \to |F|$$

a function on events satisfying the further two conditions below:

configuration-preserving: *for all $x \in \mathscr{C}(E)$, $fx \in \mathscr{C}(F)$,*
locally injective: *for all $e_1, e_2 \in x \in \mathscr{C}(E)$, if $fe_1 = fe_2$ then $e_1 = e_2$.*

This is a kind of simulation map. It transports valid states into valid states while preserving atomicity: within a configuration $x \in \mathscr{C}(E)$, *local injectivity* ensures that f induces a bijection $f : x \simeq fx$; adding an event to x must add an event to fx. We shall review a few basic properties of maps of event structures in Section 6.1.3, but for now they let us introduce a first approximation of strategies-as-event-structures:

Definition 6.1.5. *A **prestrategy** on game A is a tuple $\sigma = (|\sigma|, \leq_\sigma, \#_\sigma, \partial_\sigma)$ where*

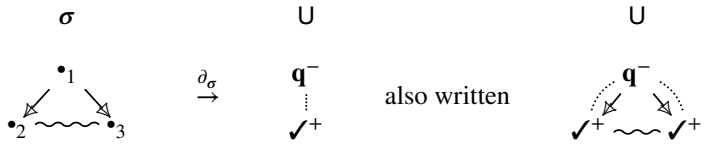
$$\partial_\sigma : \sigma \to A$$

is a map of event structures.

Observe that here σ is a plain event structures, not assumed equipped with a polarity function. Nevertheless, there is an induced notion of polarity on σ, obtained as $\mathrm{pol}_\sigma(s) = \mathrm{pol}_A(\partial_\sigma(s))$ – we shall use this implicitly from now on.

In contrast with Definition 3.1.2, a prestrategy is not a collection of standalone executions (plays), but one *global* object, an event structure σ, aggregating all possible executions. The events in $|\sigma|$ are not events of the game: they are internal to the prestrategy, only linked to the game via the *display map* ∂_σ.

For instance, we may have a prestrategy (in fact, a strategy) on U:



with the obvious mapping $\partial_\sigma(\bullet_1) = \mathbf{q}^-$ and $\partial_\sigma(\bullet_2) = \partial_\sigma(\bullet_3) = \checkmark^+$. We generally prefer drawing (pre)strategies as on the right hand side, which retains all the relevant information, only omitting the *names* of events in σ (*i.e.* $\bullet_1, \bullet_2, \bullet_3$) [2].

Strategies. The definition of prestrategies ignores polarities, thus not all prestrategies are computationally sensible – we need conditions analogous to Definition 5.1.5:

Definition 6.1.6. *A prestrategy σ on game A is a **strategy** if it satisfies:*

courteous: *for all $s_1 \rightarrowtriangle_\sigma s_2$, if $\mathrm{pol}(s_1) = +$ or $\mathrm{pol}(s_2) = -$ then $\partial_\sigma(s_1) \rightarrowtriangle_A \partial_\sigma(s_2)$,*
receptive: *for all $x \in \mathscr{C}(\sigma)$, for all $\partial_\sigma(x) \vdash_A a^-$,*
there is a unique $x \vdash_\sigma s$ such that $\partial_\sigma(s) = a^-$,

We write $\sigma : A$ to mean that σ is a strategy on game A.

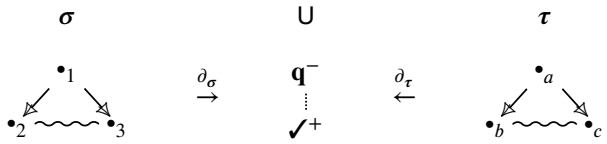
Receptive forces strategies to acknowledge uniquely any possible move by Opponent. Finally, *courtesy* gets two birds with one stone. Firstly, it forbids causal links of the form $s_1 \rightarrowtriangle_\sigma s_2^-$ which are not in the game – a strategy cannot force Opponent to wait before playing a move available in the game. Secondly, it also forbids causal links of the form $s_1^+ \rightarrowtriangle_\sigma s_2^+$ not in the game. This is to be understood as an *asynchrony* requirement:

[2] This diagram exploits our convention to write immediate causal links differently in games and strategies.

while a program may play two positive moves in a row, it cannot control in which order those will reach the Opponent; the moves are thought of as propagating independently in an asynchronous medium (*say*, the internet), so that their order may change.

This completes our first notion of *concurrent strategy*.

Mediating between strategies. Hopefully, it is apparent to the reader why strategies are formalized as certain maps $\partial_\sigma : \sigma \to A$, with σ built on a set of events separated from A. However, this also means that equations between strategies should not be expected to hold up to *equality* – otherwise the two maps of event structures



(6.4)

would denote different strategies, though they only differ via the *name* they assign events. To account for that, we shall compare strategies via the following *morphisms*:

Definition 6.1.7. *Consider σ and τ two (pre)strategies on a game A.*

*A **morphism** from σ to τ, written $f : \boldsymbol{\sigma} \Rightarrow \boldsymbol{\tau}$, is a map of es $f : \sigma \to \tau$ satisfying:*

$$\begin{array}{rl} \text{compatibility with display maps:} & \partial_{\boldsymbol{\tau}} \circ f = \partial_{\boldsymbol{\sigma}}, \\ \text{rigidity:} & \textit{for all } s \leq_{\boldsymbol{\sigma}} t, \textit{ we have } f\,s \leq_{\boldsymbol{\tau}} f\,t. \end{array}$$

*An **isomorphism of strategies** $f : \sigma \cong \tau$ is simply an invertible morphism. We sometimes write $\sigma \cong \tau$ for the mere existence of an isomorphism between σ and τ.*

We assume rigidity here as it simplifies some proofs, notably later on in the presence of symmetry. Our next goal is to organize concurrent strategies into a bicategory **CG**; but for that we will need a bit of the basic theory on event structures.

6.1.3 Basic Properties of Event Structures and Maps

First, event structures and their maps (as in Definition 6.1.4) form a category **ES**. For E an event structure, we often write $e \in E$ as an alias for $e \in E$. If $e \in E$,

$$[e]_E = \{e' \in E \mid e' \leq_E e\}$$

is the set of **causal dependencies** of e. It is an easy exercise to prove that $[e]_E \in \mathscr{C}(E)$ – we shall use that fact without mentioning it. We say that $x \in \mathscr{C}(E)$ is a **prime configuration** iff it has a top element, or equivalently if it is $x = [e]_E$ for some $e \in E$. It is clear that events of E are in one-to-one correspondence with prime configurations.

Compatibility with the order. Event structures are posets, so one may expect their maps to preserve the order. But this is not at all the intention, maps of event structures are not meant to be monotone. The following is a perfectly acceptable map:

$$1 \longrightarrow\!\!\!\!\!\rightarrow 2 \qquad \to \qquad 1 \qquad 2$$

sending two events in causal dependency to causally independent moves. Thus causality is not preserved, but we do have a property in the other direction:

Lemma 6.1.8. *Consider $f : E \to F$ a map of event structures.*
Then, for all $e, e' \in x \in \mathscr{C}(E)$, if $fe \leq_F fe'$ then $e \leq_E e'$.

Proof. Since $e' \in E$, we have $[e']_E \in \mathscr{C}(E)$, hence $f[e']_E \in \mathscr{C}(F)$ since f preserves configurations. But by construction, $fe' \in f[e']_E$. Since $f[e']_E \in \mathscr{C}(F)$, it is in particular down-closed. Hence, since $fe \leq_F fe'$, we must have $fe \in f[e']_E$. But this means that there is some $e'' \in [e']_E$ such that $fe'' = fe$ – in other words, there is $e'' \leq_E e'$ such that $fe'' = fe$. Now, we remark that we have $e' \in x$, so as x is down-closed, we must have $e'' \in x$ as well. Thus $e, e'' \in x$ have the same image by f, thus $e = e''$ by local injectivity. Hence, we have proved that $e \leq_E e'$ as required. □

Recall that we have defined prestrategies as maps $\partial_\sigma : \sigma \to A$. For $\sigma : A$, the lemma above means that if $s, s' \in x \in \mathscr{C}(\sigma)$, if $\partial_\sigma s \leq_A \partial_\sigma s'$ then $s \leq_\sigma s'$ – in other words, any "static" causal dependency imposed by the game must be respected by the strategy as well. But causality is reflected only within configurations: the map

$$\begin{array}{ccc} 1 \rightsquigarrow 1' & & \\ \downarrow & \to & \downarrow \\ 2 & & 2 \end{array}$$

is valid, though we have $f1' \leq f2$ but $\neg(1' \leq 2)$ as they are in conflict.

The maps of event structures that *are* monotone are called **rigid**. In that case:

Lemma 6.1.9. *Consider $f : E \to F$ a rigid map of event structures.*
Then, f preserves immediate causality.

Proof. Left as an exercise to the reader. □

Rigid maps $f : E \to F$ are too constrained to be interesting as strategies, but they are useful as a sort of inclusions of E within F, collapsing non-deterministic branches. The *morphisms between strategies*, as defined in Definition 6.1.7, are rigid maps.

In Definition 6.1.7 we encountered for the first time the *isomorphisms* in the category of event structures, as induced by Definition 6.1.4. It is worth pointing out explicitly that such isomorphisms of event structures (and thus, isomorphisms of strategies) are simply *renamings* of the events, preserving and reflecting all structure:

Lemma 6.1.10. *Consider E, F event structures, $f : E \to F$ an isomorphism of es.*
Then, f is a bijection between events, satisfying the following two properties:

(1) *for all $e, e' \in E$, $e \leq_E e'$ iff $f(e) \leq_F f(e')$,*
(2) *for all $e, e' \in E$, $e \#_E e'$ iff $f(e) \#_F f(e')$.*

Proof. Left as an exercise to the reader. □

Reciprocally, bijections satisfying properties *(1)* and *(2)* above are obviously isomorphisms of event structures – clearly, the two strategies of (6.4) are isomorphic.

Mapification. The above shows that a map of es $f : E \to F$ should not be viewed as a map of posets; instead it should be seen as a concrete presentation of

$$f : \mathscr{C}(E) \to \mathscr{C}(F)$$

a monotone function between the generated domains of configurations (the sets of configurations, ordered by inclusion). In fact, we will often reason on event structures via their domains of configurations; likewise we shall build maps of event structures via their action on configurations – we call the lemma below the "mapification" lemma, letting us build maps of event structures from functions on configurations:

Lemma 6.1.11. *Consider $f : \mathscr{C}(E) \to \mathscr{C}(F)$ preserving unions and cardinality. Then, there is a unique map of es $\widehat{f} : E \to F$ s.t. for all $x \in \mathscr{C}(E)$, $\widehat{f}(x) = f(x)$.*

Proof. Existence. As f preserves unions it is monotone. As it preserves cardinality, it preserves $\emptyset$ and $\multimap$. The key point is that it preserves *covering squares*:

$$\begin{array}{ccc} x & \overset{a}{\multimap} & y \\ b \frown & & \frown b \\ u & \underset{a}{\multimap} & v \end{array}$$

for $a \neq b$. Indeed as f preserves $\multimap$, $f(x) \overset{b'}{\multimap} f(y)$ and $f(x) \overset{a'}{\multimap} f(u)$. But also $f(v) = f(y \cup u) = f(y) \cup f(u)$ since f preserves unions, thus $f(v) = f(x) \cup \{a', b'\}$. Hence,

$$\begin{array}{ccc} f(x) & \overset{a'}{\multimap} & f(y) \\ b' \frown & & \frown b' \\ f(u) & \underset{a'}{\multimap} & f(v) \end{array}$$

Now, we are in position to define $\widehat{f}$. Given $e \in E$, write

$$[e)_E = \{e' \in E \mid e' <_E e\}$$

its set of **strict causal dependencies**, which is automatically a configuration. Of course,

$$[e)_E \quad \overset{e}{\multimap} \quad [e]_E$$

so that $f[e)_E \overset{e'}{\multimap} f[e]_E$ for some $e' \in F$ – set $\widehat{f}(e) = e'$. Now, for any $x \overset{e}{\multimap} y$, we have

$$\begin{array}{ccccccccccccc} & [e)_E & \overset{e_1}{\multimap} & x_1 & \overset{e_2}{\multimap} & x_2 & \overset{e_3}{\multimap} & x_3 & \dots & x_{n-1} & \overset{e_n}{\multimap} & x & \\ e & \frown & & \frown & & \frown & & \frown & & \frown & & \frown & e \\ & [e]_E & \underset{e_1}{\multimap} & y_1 & \underset{e_2}{\multimap} & y_2 & \underset{e_3}{\multimap} & y_3 & \dots & y_{n-1} & \underset{e_n}{\multimap} & y & \end{array}$$

a tiling of covering squares, which are all preserved by f, implying $fx \overset{\widehat{f(e)}}{\multimap} fy$. But now, for any $x \in \mathscr{C}(E)$, one may reach it in $\mathscr{C}(E)$ adding events one by one, *i.e.* there is

$$\emptyset = x_0 \overset{e_1}{\multimap} x_1 \overset{e_2}{\multimap} \dots x_{n-1} \overset{e_n}{\multimap} x_n = x$$

a **covering chain** in E obtained as any linearization of x according to $\leq_E$. But then

$$\emptyset = f(x_0) \overset{\widehat{f}(e_1)}{\multimap} f(x_1) \overset{\widehat{f}(e_2)}{\multimap} \dots f(x_{n-1}) \overset{\widehat{f}(e_n)}{\multimap} f(x_n) = f(x)$$

from the observation above. But this means that $f(x) = \widehat{f}(x)$, the latter defined by the direct image. It immediately follows that $\widehat{f}$ is a map of event structures: as it coincides with f of configuration it must preserve configurations, and a failure of local injectivity would immediately contradict the fact that f preserves cardinality.

Uniqueness. For $h : E \to F$ s.t. $h(x) = f(x)$ for all $x \in \mathscr{C}(E)$, then if $e \in E$,

$$f([e)_E) \quad \overset{h(e)}{\multimap} \quad f([e]_E)$$

so that $h(e) = \widehat{f}(e)$ as required, concluding the proof. □

Of course, reciprocally, if $f : E \to F$ is a map of event structures then $f : \mathscr{C}(E) \to \mathscr{C}(F)$ must preserve cardinality and unions as it is defined as a direct image. It follows that to test equality of maps of es, it suffices to evaluate them on configurations:

Lemma 6.1.12. *Consider $f, g : E \to F$ maps of event structures.*
If for all $x \in \mathscr{C}(E)$ we have $f(x) = g(x)$, then $f = g$.

Proof. Obvious by the *uniqueness* clause of Lemma 6.1.11. □

Another particularly useful consequence of the mapification lemma is:

Proposition 6.1.13. *Consider E, F two es, and $f : \mathscr{C}(E) \cong \mathscr{C}(F)$ an order-iso.*
Then, there is a unique $\widehat{f} : E \cong F$ an iso in **ES** *s.t. for all $x \in \mathscr{C}(E)$, $\widehat{f}(x) = f(x)$.*

Proof. Firstly, f preserves unions as it is an order-isomorphisms and unions are least upper bounds in $\mathscr{C}(E)$. Secondly, as f is a bijection preserving and reflecting inclusion, it must preserve $\multimap$ and thus preserves cardinality. Hence, by Lemma 6.1.11 there is $\widehat{f} : E \to F$ a map of event structures such that for all $x \in \mathscr{C}(E)$, $\widehat{f}(x) = f(x)$. The same argument applies to f^{-1}, yielding an inverse to f by Lemma 6.1.12. □

Paired with Lemma 6.1.10, we see that given two event structures E and F, an order-iso between the domains of configurations $\mathscr{C}(E)$ and $\mathscr{C}(F)$ induces a concrete *renaming* between E and F, a bijection between events preserving and reflecting all structure. In the future, this will provide our main tool to establish isomorphisms between event structures: by exhibiting an order-isomorphism between the domains of configurations.

Parallel composition. Next we introduce the *parallel composition* of event structures, an operation which shall play a central role throughout the development.

Definition 6.1.14. *Consider E_1 and E_2 two event structures.*
*Their **simple parallel composition**, written $E_1 \parallel E_2$, has components:*

$$\begin{array}{rrcl} \text{events:} & |E_1 \parallel E_2| & = & |E_1| + |E_2|\,, \\ \text{causality:} & (i,e) \leq_{E_1 \parallel E_2} (j,e') & \Leftrightarrow & i = j \;\&\; e \leq_{E_i} e'\,, \\ \text{conflict:} & (i,e) \#_{E_1 \parallel E_2} (j,e') & \Leftrightarrow & i = j \;\&\; e \#_{E_i} e'\,. \end{array}$$

This simply puts the two event structures side by side, with no interaction. We shall also apply this construction to configurations: if $x_1 \in \mathscr{C}(E_1)$ and $x_2 \in \mathscr{C}(E_2)$, then $x_1 \parallel x_2 \in \mathscr{C}(E_1 \parallel E_2)$ is defined as the same tagged disjoint union as above, *i.e.* $x_1 \parallel x_2 = x_1 + x_2$. Moreover, all configurations on $E_1 \parallel E_2$ have this form:

Lemma 6.1.15. *Consider E_1 and E_2 two event structures. Then, we have*

$$(- \parallel -) : \mathscr{C}(E_1) \times \mathscr{C}(E_2) \cong \mathscr{C}(E_1 \parallel E_2)$$

an order-isomorphism, with the order on $\mathscr{C}(E_1) \times \mathscr{C}(E_2)$ set as the product order.

Proof. Straightforward. □

Along this isomorphism, all configurations of $E_1 \parallel E_2$ are uniquely written as pairs – accordingly, we shall often write $x_1 \parallel x_2 \in \mathscr{C}(E_1 \parallel E_2)$ for arbitrary configurations on a parallel composition. Note that up to isomorphism, $E_1 \parallel E_2$ is the *unique* event structure whose domain of configurations is order-isomorphic to $\mathscr{C}(E_1) \times \mathscr{C}(E_2)$ – this immediately follows from Lemma 6.1.15 and Proposition 6.1.13.

Basic properties of strategies. To put our preliminaries into play, we mention a few facts on strategies. The proofs are relatively elementary with the ingredients introduced so far. We omit them, but invite the reader to work them out as exercises.

Recall that if $\sigma : A$ is a prestrategy on game A, then events of σ inherit a polarity from A via $\mathrm{pol}_\sigma(s) = \mathrm{pol}_A(\partial_\sigma(s))$. Our first lemma states two properties of strategies in the particular case where the game A is forestial and alternating (which is almost always the case when interpreting types of programming languages).

Lemma 6.1.16. *Consider a game A* forestial *and* alternating *as in Definition 3.1.3.*
Then, we have the following properties:

(1) *σ is alternating: if $s_1 \rightarrowtriangle_\sigma s_2$, then $\mathrm{pol}_\sigma(s_1) \neq \mathrm{pol}_\sigma(s_2)$.*
(2) *consider $s_1, s_2 \in x \in \mathscr{C}(\sigma)$ such that $\mathrm{pol}_\sigma(s_1) = +$ or $\mathrm{pol}_\sigma(s_2) = -$.*
Then $s_1 \rightarrowtriangle_\sigma s_2$ iff $\partial_\sigma s_1 \rightarrowtriangle_A \partial_\sigma s_2$.

Thus for such games, the causal structure of strategies is very much constrained: it is alternating, and only immediate causal links $a_1^- \rightarrowtriangle_A a_2^+$ in the game may be postponed by the strategy. In general and even without alternation, a strategy may only introduce new immediate conflicts (with respect to the game) between positive events:

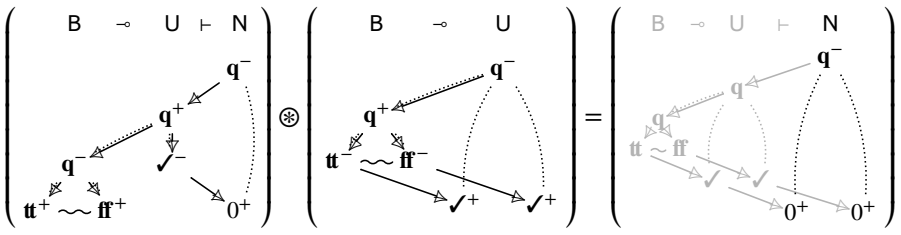


Figure 6.1: Example of an interaction and composition

Lemma 6.1.17. *Consider A a game, and $\sigma : A$ a strategy.*
If $s_1 \sim_\sigma s_2$ with $\mathrm{pol}_\sigma(s_1) = -$ or $\mathrm{pol}_\sigma(s_2) = -$, then $\partial_\sigma(s_1) \sim_A \partial_\sigma(s_2)$ as well.

Now, we move to the *composition* of (pre)strategies.

6.2 Composition of Prestrategies

Before we can phrase composition, we must define *strategies between games*. As in sequential games, a strategy *from a game A to a game B* is a strategy that plays on A and B in parallel, except that the polarities in A are reversed.

Strategies between games. If A is a game, its **dual** $A^\perp$ has the same components as A, except for $\mathrm{pol}_{A^\perp} = -\mathrm{pol}_A$. The **simple parallel composition** of Definition 6.1.14 extends to games with $\mathrm{pol}_{A\parallel B}(1, a) = \mathrm{pol}_A(a)$ and $\mathrm{pol}_{A\parallel B}(2, b) = \mathrm{pol}_B(b)$. Finally, the **hom-game** $A \vdash B$ is defined as $A^\perp \parallel B$ – by Lemma 6.1.15 we have an order-iso

$$(- \vdash -) : \mathscr{C}(A) \times \mathscr{C}(B) \cong \mathscr{C}(A \vdash B),$$

and we often write configurations of $A \vdash B$ as $x_A \vdash x_B$ for $x_A \in \mathscr{C}(A)$ and $x_B \in \mathscr{C}(B)$.

Now if A and B are games, a **strategy from A to B** is simply defined as a strategy on $A \vdash B$ – the same definition applies to prestrategies.

Towards composition. Now, consider two strategies $\sigma : A \vdash B$ and $\tau : B \vdash C$ that we wish to compose, forming a composite strategy $\tau \odot \sigma : A \vdash C$.

Intuitively, the computation of $\tau \odot \sigma$ proceeds in two steps. First, we form the *interaction* $\tau \circledast \sigma$; an event structure displayed to $A \parallel B \parallel C$. The interaction has three kinds of events: those displayed to A are directly imported from σ, those displayed to C come from τ, while those in B are thought of *synchronizations* between events of σ and events of τ whose display on B match. The *causality* of $\tau \circledast \sigma$ is obtained as (the transitive closure) of the causal dependencies imposed by σ and by τ. Figure 6.1

displays the interaction of two strategies[3], and again the conflict is that imported from the two strategies. In Figure 6.1 on the right hand side, the synchronized events are grayed out while those in the outer interface remain in black – those are called *visible*. The *composition* $\boldsymbol{\tau} \odot \boldsymbol{\sigma}$ is obtained from $\boldsymbol{\tau} \circledast \boldsymbol{\sigma}$ by keeping the visible events only.

This intuition is quite natural, but rather hard to capture formally in a way that is both mathematically rigorous and a tractable basis for further developments – much more so than the composition of sequential strategies as in Part I, which is already non-trivial. And while our forthcoming development of composition will indeed follow the route above, it has been crucial in working with concurrent games that the composition of strategies may be characterized almost relationally, via the proposition:

Proposition 6.2.1. *Consider A, B, C games, and $\boldsymbol{\sigma} : A \vdash B$ and $\boldsymbol{\tau} : B \vdash C$ strategies. Then there is a strategy $\boldsymbol{\tau} \odot \boldsymbol{\sigma} : A \vdash C$, unique up to iso, s.t. there are order-isos:*

$$(- \odot -) \;:\; \{(x^{\boldsymbol{\tau}}, x^{\boldsymbol{\sigma}}) \in \mathscr{C}^+(\boldsymbol{\tau}) \times \mathscr{C}^+(\boldsymbol{\sigma}) \mid \textit{causally compatible}\} \;\simeq\; \mathscr{C}^+(\boldsymbol{\tau} \odot \boldsymbol{\sigma})$$

such that for $x^{\boldsymbol{\sigma}} \in \mathscr{C}^+(\boldsymbol{\sigma})$ and $x^{\boldsymbol{\tau}} \in \mathscr{C}^+(\boldsymbol{\tau})$ causally compatible,

$$\partial_{\boldsymbol{\tau} \odot \boldsymbol{\sigma}}(x^{\boldsymbol{\tau}} \odot x^{\boldsymbol{\sigma}}) = x^{\boldsymbol{\sigma}}_A \vdash x^{\boldsymbol{\tau}}_C$$

where we write $\partial_{\boldsymbol{\sigma}} x^{\boldsymbol{\sigma}} = x^{\boldsymbol{\sigma}}_A \vdash x^{\boldsymbol{\sigma}}_B$ and $\partial_{\boldsymbol{\tau}} x^{\boldsymbol{\tau}} = x^{\boldsymbol{\tau}}_B \vdash x^{\boldsymbol{\tau}}_C$.

Here, $\mathscr{C}^+(-)$ denotes the restriction of configurations to those that are $+$*-covered*, *i.e.* whose maximal events are positive – without trailing Opponent moves. Likewise, *causally compatible* refers to a notion that we shall introduce soon: intuitively, $x^{\boldsymbol{\sigma}} \in \mathscr{C}(\boldsymbol{\sigma})$ and $x^{\boldsymbol{\tau}} \in \mathscr{C}(\boldsymbol{\tau})$ are causally compatible if they reach the same state in $\boldsymbol{B}$ (*i.e.* $x^{\boldsymbol{\sigma}}_B = x^{\boldsymbol{\tau}}_B$), and *their synchronization induces no deadlocks*. These two notions will be covered in depth soon, but the message for now is that for most purposes, we can ignore the concrete definition of $\boldsymbol{\sigma}$ and $\boldsymbol{\tau}$ and treat its ($+$-covered) configurations simply as certain pairs of ($+$-covered) configurations of $\boldsymbol{\sigma}$ and $\boldsymbol{\tau}$. In addition, this completely captures the composition of $\boldsymbol{\sigma}$ and $\boldsymbol{\tau}$ – there is *one* strategy (up to iso) whose $+$-covered configurations are such causally compatible pairs, and it is $\boldsymbol{\tau} \odot \boldsymbol{\sigma}$.

Although noticed somewhat late (say around 2020), this proposition has become central in recent developments in concurrent games, as it lets us reason on the composition of strategies while leaving encapsulated the rather elaborate concrete definition. For most purposes, this is all you need to know about composition!

However, this still leaves us with the task of *proving* it, and thus to construct the composition concretely – first the *interaction*, and then its *hiding*. In the present section we focus on *prestrategies*, while Section 6.3 contains the extension to *strategies*.

6.2.1 Interaction of prestrategies

The next goal is to define *interactions* of prestrategies. But before that, we define formally the notion of *causal compatibility* appearing in Proposition 6.2.1.

[3]The game $\mathsf{B} \multimap \mathsf{U}$ is defined only in Section 7.1.4, and likewise for other forthcoming examples; hopefully this is not an obstacle to understanding. Though do note that in such examples, the game can be read from the diagram by only keeping dotted lines and ignoring immediate causal links $\rightarrowtriangle$.

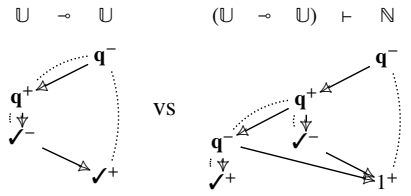


Figure 6.2: Matching, secured

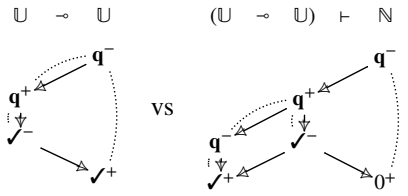


Figure 6.3: Matching, non-secured

Causally compatible pairs. Let us fix $\sigma : A \vdash B$ and $\tau : B \vdash C$ prestrategies. For configurations $x^\sigma \in \mathscr{C}(\sigma), x^\tau \in \mathscr{C}(\tau)$, as a convention we write their display as

$$\partial_\sigma(x^\sigma) = x^\sigma_A \parallel x^\sigma_B \in \mathscr{C}(A \vdash B), \qquad \partial_\tau(x^\tau) = x^\tau_B \parallel x^\tau_C \in \mathscr{C}(B \vdash C),$$

this convention will be used silently from now on.

First, we capture when $x^\sigma \in \mathscr{C}(\sigma)$ and $x^\tau \in \mathscr{C}(\tau)$ may successfully *synchronise*:

Definition 6.2.2. *Two configurations $x^\sigma \in \mathscr{C}(\sigma)$ and $x^\tau \in \mathscr{C}(\tau)$ are **causally compatible** if* (1) *they are **matching**: $x^\sigma_B = x^\tau_B = x_B$; and* (2) *if the composite bijection*

$$\varphi[x^\sigma, x^\tau] \quad : \quad x^\sigma \parallel x^\tau_C \quad \overset{\partial_\sigma \parallel x^\tau_C}{\simeq} \quad x^\sigma_A \parallel x_B \parallel x^\tau_C \quad \overset{x^\sigma_A \parallel \partial_\tau^{-1}}{\simeq} \quad x^\sigma_A \parallel x^\tau,$$

*using local injectivity of ∂_σ and ∂_τ, is **secured**, in the sense that the relation*

$$(m, n) \lhd (m', n') \qquad \Leftrightarrow \qquad m <_{\sigma \parallel C} m' \quad \vee \quad n <_{A \parallel \tau} n',$$

defined on (the graph of) $\varphi[x^\sigma, x^\tau]$ from the causal constraints of σ and τ, is acyclic.

Two matching $x^\sigma \in \mathscr{C}(\sigma)$, $x^\tau \in \mathscr{C}(\tau)$ agree on the state reached in B; and this induces a synchronization between the events matching in B. But this is not enough to capture a sensible notion of execution: some matching pairs might not be *reachable*, in the sense that σ and τ impose incompatible constraints as to the order in which the state should be reached. To illustrate this we show in Figures 6.2 and 6.3 two attempted synchronizations between configurations of the strategy on the left hand side of Figure 7.6 and that for the identity $\lambda x^{\mathbb{U}}.\, x$. In both cases, the configurations are matching. In Figure 6.2, the synchronization is successful and yields a causally compatible pair. However, in Figure 6.3 the induced bijection is not *secured*: the two strategies impose opposite constraints as to the order in which the two ✓ moves are to be played.

This disambiguates the statement of Proposition 6.2.1 – we want an *event structure* $\tau \odot \sigma$ whose (+-covered) configurations correspond to (+-covered) causally compatible pairs. Its construction involves several steps, starting with the *interaction*.

Interactions. Given $\sigma : A \vdash B$ and $\tau : B \vdash C$ we mean to construct their interaction $\tau \circledast \sigma$. An *interaction* is not quite a strategy, because it involves events in B that have no clear polarity. Accordingly, it is convenient to define a variation:

Definition 6.2.3. *An **interaction** on A, B, C is an event structure $\boldsymbol{\mu} = (|\boldsymbol{\mu}|, \leq_{\boldsymbol{\mu}}, \#_{\boldsymbol{\mu}})$ with*

$$\partial : |\boldsymbol{\mu}| \to |A \parallel B \parallel C|$$

a display map subject to the following conditions:

rule-abiding: *for all $x \in \mathscr{C}(\boldsymbol{\mu})$, $\partial(x) \in \mathscr{C}(A \parallel B \parallel C)$,*
locally injective: *for all $s_1, s_2 \in x \in \mathscr{C}(\boldsymbol{\mu})$, if $\partial(s_1) = \partial(s_2)$ then $s_1 = s_2$.*

i.e. $\partial : \boldsymbol{\mu} \to A \parallel B \parallel C$ *is a* map of event structures.

In other words an interaction on A, B, C is a prestrategy on $A \parallel B \parallel C$, but it is conceptually clearer to keep the notions separate. Now we must define the interaction, written $\boldsymbol{\tau} \circledast \boldsymbol{\sigma}$, of $\boldsymbol{\sigma}$ and $\boldsymbol{\tau}$; but what should be its *events*? Intuitively, each event of $\boldsymbol{\tau} \circledast \boldsymbol{\sigma}$ arises as the synchronization of an event in $\boldsymbol{\sigma} \parallel C$ and one in $A \parallel \boldsymbol{\tau}$ – note the padding out so that both $\boldsymbol{\sigma} \parallel C$ and $A \parallel \boldsymbol{\tau}$ cover $A \parallel B \parallel C$.

But it is not that simple, as can be observed in Figure 6.1, showing an interaction $\boldsymbol{\tau} \circledast \boldsymbol{\sigma}$ of $\boldsymbol{\sigma} : \mathsf{B} \multimap \mathsf{U}$ and $\boldsymbol{\tau} : \mathsf{B} \multimap \mathsf{U} \vdash \mathsf{N}$. The two final events 0^+ of $\boldsymbol{\tau} \circledast \boldsymbol{\sigma}$ are two unsynchronized versions of the *same* event in $\boldsymbol{\tau}$. Yet they are different: they have different causal histories, following the earlier non-deterministic choice in $\boldsymbol{\tau}$. This has to do with an important property of the model we already mentioned: it retains the *non-deterministic branching point* – any event carries its entire causal history.

A pair of synchronized events with a causal history is a *prime secured bijection*:

Prime secured bijections. If $x^{\boldsymbol{\sigma}} \in \mathscr{C}(\boldsymbol{\sigma})$ and $x^{\boldsymbol{\tau}} \in \mathscr{C}(\boldsymbol{\tau})$ are causally compatible, from Definition 6.2.2 we know that the relation on the graph of $\varphi[x^{\boldsymbol{\sigma}}, x^{\boldsymbol{\tau}}]$ defined as

$$(m, n) \lhd (m', n') \qquad \Leftrightarrow \qquad m <_{\boldsymbol{\sigma} \parallel C} m' \quad \vee \quad n <_{A \parallel \boldsymbol{\tau}} n' \,,$$

is acyclic, so its reflexive transitive closure yields a partial order $\leq_{\varphi[x^{\boldsymbol{\sigma}}, x^{\boldsymbol{\tau}}]}$. We define:

Definition 6.2.4. *A **secured bijection** between $\boldsymbol{\sigma} : A \vdash B$ and $\boldsymbol{\tau} : B \vdash C$ is any partial order $\varphi[x^{\boldsymbol{\sigma}}, x^{\boldsymbol{\tau}}]$, for $x^{\boldsymbol{\sigma}} \in \mathscr{C}(\boldsymbol{\sigma})$ and $x^{\boldsymbol{\tau}} \in \mathscr{C}(\boldsymbol{\tau})$ causally compatible.*
We write $\mathscr{B}(\boldsymbol{\sigma}, \boldsymbol{\tau})$ for the set of secured bijections between $\boldsymbol{\sigma}$ and $\boldsymbol{\tau}$.

We shall be particularly interested in those secured bijections with a top element:

Definition 6.2.5. *We say $\varphi \in \mathscr{B}(\boldsymbol{\sigma}, \boldsymbol{\tau})$ is **prime** if $\leq_\varphi$ has a top $\mathsf{top}(\varphi) = (m, n)$.*
We write $\mathscr{B}^\top(\boldsymbol{\sigma}, \boldsymbol{\tau})$ for the set of prime secured bijections between $\boldsymbol{\sigma}$ and $\boldsymbol{\tau}$.

So we regard $\varphi \in \mathscr{B}^\top(\boldsymbol{\sigma}, \boldsymbol{\tau})$ with $\mathsf{top}(\varphi) = (m, n)$ as the synchronized pair (m, n) *together with* a causal explanation for that synchronization. Those will provide the *events* of the interaction. To proceed, we will need the lemma:

Lemma 6.2.6. *Consider $x^{\boldsymbol{\sigma}}, y^{\boldsymbol{\sigma}} \in \mathscr{C}(\boldsymbol{\sigma})$, $x^{\boldsymbol{\tau}}, y^{\boldsymbol{\tau}} \in \mathscr{C}(\boldsymbol{\tau})$. Then, we have:*

(1) *If $x^{\boldsymbol{\sigma}}, x^{\boldsymbol{\tau}}$ caus. comp., $y^{\boldsymbol{\sigma}}, y^{\boldsymbol{\tau}}$ caus. comp., $x^{\boldsymbol{\sigma}} \cup y^{\boldsymbol{\sigma}} \in \mathscr{C}(\boldsymbol{\sigma})$ and $x^{\boldsymbol{\tau}} \cup y^{\boldsymbol{\tau}} \in \mathscr{C}(\boldsymbol{\tau})$, then $x^{\boldsymbol{\sigma}} \cup y^{\boldsymbol{\sigma}}$ and $x^{\boldsymbol{\tau}} \cup y^{\boldsymbol{\tau}}$ are causally compatible.*
(2) *If $x^{\boldsymbol{\sigma}} \cup y^{\boldsymbol{\sigma}} \in \mathscr{C}(\boldsymbol{\sigma})$ and $x^{\boldsymbol{\tau}} \cup y^{\boldsymbol{\tau}} \in \mathscr{C}(\boldsymbol{\tau})$ are causally compatible, if $x^{\boldsymbol{\sigma}}, x^{\boldsymbol{\tau}}$ matching and $y^{\boldsymbol{\sigma}}, y^{\boldsymbol{\tau}}$ matching, then they are causally compatible,*

and in both cases, we have $\varphi[x^{\boldsymbol{\sigma}} \cup y^{\boldsymbol{\sigma}}, x^{\boldsymbol{\tau}} \cup y^{\boldsymbol{\tau}}] = \varphi[x^{\boldsymbol{\sigma}}, x^{\boldsymbol{\tau}}] \cup \varphi[y^{\boldsymbol{\sigma}}, y^{\boldsymbol{\tau}}]$.

Proof. Straightforward by local injectivity of ∂_{σ} and ∂_{τ}. □

We are now in position to form an event structure:

Proposition 6.2.7. *The tuple* $(|\tau \circledast \sigma|, \leq_{\tau\circledast\sigma}, \#_{\tau\circledast\sigma})$ *defined with*

$$\begin{array}{rrcl} \text{events:} & |\tau \circledast \sigma| & = & \{\varphi[x^{\sigma}, x^{\tau}] \mid \textit{prime, for } x^{\sigma}, x^{\tau} \textit{ causally compatible}\} \\ \text{causality:} & \phi \leq_{\tau\circledast\sigma} \psi & \Leftrightarrow & \phi \subseteq \psi \\ \text{conflict:} & \neg(\phi \#_{\tau\circledast\sigma} \psi) & \Leftrightarrow & \exists x^{\sigma}, x^{\tau} \textit{ causally compatible},\ \phi, \psi \subseteq \varphi[x^{\sigma}, x^{\tau}]. \end{array}$$

forms an event structure. Moreover, there are monotone functions

$$\begin{array}{rcccl} \chi_{\sigma,\tau} & : & \mathscr{C}(\tau \circledast \sigma) & \to & \mathscr{B}(\sigma, \tau) \\ & & x & \mapsto & \cup x \end{array}$$

$$\begin{array}{rcccl} \bar{\chi}_{\sigma,\tau} & : & \mathscr{B}(\sigma, \tau) & \to & \mathscr{C}(\tau \circledast \sigma) \\ & & \varphi & \mapsto & \{\psi \in \mathscr{B}^{\top}(\sigma, \tau) \mid \psi \subseteq \varphi\} \end{array}$$

forming an order-isomorphism, with both sets ordered by inclusion.

Proof. The conditions for an event structure are easily checked. It is immediate that $\bar{\chi}_{\sigma,\tau}$ is well-formed, for $\chi_{\sigma,\tau}$ it follows from Lemma 6.2.6; both maps are monotone with respect to inclusion. Next we observe that for all $\varphi \in \mathscr{B}(\sigma, \tau)$, we have

$$\varphi \quad = \quad \cup\{\psi \in \mathscr{B}^{\top}(\sigma, \tau) \mid \psi \subseteq \varphi\}\,;$$

indeed if $(m, n) \in \varphi$ we set $\psi = [(m, n)]_{\varphi}$ the set of all pairs $(m', n') \in \varphi$ such that $(m', n') \leq_{\varphi} (m, n)$ (recall from above Definition 6.2.4 that causal compatibility endows the graph of any secured bijection with a partial order); by construction, $\psi \in \mathscr{B}^{\top}(\sigma, \tau)$ and $\psi \subseteq \varphi$, hence $(m, n) \in \cup\{\psi \in \mathscr{B}^{\top}(\sigma, \tau) \mid \psi \subseteq \varphi\}$ – the other inclusion is direct. Altogether, this shows that $\chi_{\sigma,\tau} \circ \bar{\chi}_{\sigma,\tau}\, \varphi = \varphi$ for all $\varphi \in \mathscr{B}(\sigma, \tau)$ a secured bijection.

Now consider $X \in \mathscr{C}(\tau \circledast \sigma)$ along with $\varphi[x^{\sigma}, x^{\tau}] \in \bar{\chi}_{\sigma,\tau} \circ \chi_{\sigma,\tau}\, X$, and take $(m, n) = \mathsf{top}(\varphi[x^{\sigma}, x^{\tau}])$. By definition, $(m, n) \in \cup X$, but this means that there is $\psi \in X$ such that $(m, n) \in \psi$. But then it follows from local injectivity that $\varphi[x^{\sigma}, x^{\tau}] \subseteq \psi$, hence $\varphi[x^{\sigma}, x^{\tau}] \in X$ as X is down-closed. The other inclusion is straightforward. □

We have now constructed an event structure with configurations order-isomorphic to the secured bijections between σ and τ. But furthermore:

Proposition 6.2.8. *There is an order-isomorphism*

$$(- \circledast -) : \{(x^{\sigma}, x^{\tau}) \in \mathscr{C}(\sigma) \times \mathscr{C}(\tau) \mid \textit{causally compatible}\} \simeq \mathscr{C}(\tau \circledast \sigma)\,,$$

with causally compatible pairs ordered by pairwise inclusion.

Proof. By definition, $\mathscr{B}(\sigma, \tau)$ is in order-isomorphism with the set of causally compatible pairs (x^{σ}, x^{τ}). We conclude by composition with that of Proposition 6.2.7. □

This is starting to look like Proposition 6.2.1. In the sequel, we will almost always reason on $\mathscr{C}(\boldsymbol{\tau} \circledast \boldsymbol{\sigma})$ via this correspondence, and routinely write any configuration $x \in \mathscr{C}(\boldsymbol{\tau} \circledast \boldsymbol{\sigma})$ as $x^{\boldsymbol{\tau}} \circledast x^{\boldsymbol{\sigma}} \in \mathscr{C}(\boldsymbol{\tau} \circledast \boldsymbol{\sigma})$, without mentioning the use of this proposition.

In passing, let us mention a few lemmas. Firstly, using the mapification lemma will require us to have information on the cardinal of configurations of the interaction:

Lemma 6.2.9. *Consider $x^{\boldsymbol{\tau}} \circledast x^{\boldsymbol{\sigma}} \in \mathscr{C}(\boldsymbol{\tau} \circledast \boldsymbol{\sigma})$.*
Then, $\mathsf{card}(x^{\boldsymbol{\tau}} \circledast x^{\boldsymbol{\sigma}}) = \mathsf{card}(x^{\boldsymbol{\sigma}}_A) + \mathsf{card}(x^{\boldsymbol{\sigma}}_B) + \mathsf{card}(x^{\boldsymbol{\tau}}_C)$.

Proof. Straightforward from the Propositions 6.2.7 and 6.2.8. □

Similarly, we need compatibility of configurations of the interation with unions:

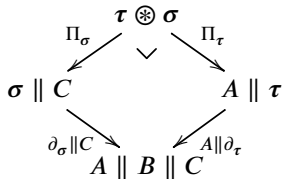
Lemma 6.2.10. *Consider $x^{\boldsymbol{\sigma}}, y^{\boldsymbol{\sigma}} \in \mathscr{C}(\boldsymbol{\sigma})$ and $x^{\boldsymbol{\tau}}, y^{\boldsymbol{\tau}} \in \mathscr{C}(\boldsymbol{\tau})$ satisfying either of the conditions of Lemma 6.2.6. Then, we have the equality:*

$$(x^{\boldsymbol{\tau}} \cup y^{\boldsymbol{\tau}}) \circledast (x^{\boldsymbol{\sigma}} \cup y^{\boldsymbol{\sigma}}) = (x^{\boldsymbol{\tau}} \circledast x^{\boldsymbol{\sigma}}) \cup (y^{\boldsymbol{\tau}} \circledast y^{\boldsymbol{\sigma}})\,.$$

Proof. By Lemma 6.2.6, accross the order-isomorphism of Proposition 6.2.7. □

The interaction pullback. As an important aside, we have:

Lemma 6.2.11. *There are maps of event structures $\Pi_{\boldsymbol{\sigma}}$ and $\Pi_{\boldsymbol{\tau}}$ making the diagram*



a pullback in the category of event structures and their maps. Moreover, for any $x^{\boldsymbol{\tau}} \circledast x^{\boldsymbol{\sigma}} \in \mathscr{C}(\boldsymbol{\tau} \circledast \boldsymbol{\sigma})$, we have $\Pi_{\boldsymbol{\sigma}}(x^{\boldsymbol{\tau}} \circledast x^{\boldsymbol{\sigma}}) = x^{\boldsymbol{\sigma}} \parallel x^{\boldsymbol{\tau}}_C$ and $\Pi_{\boldsymbol{\tau}}(x^{\boldsymbol{\tau}} \circledast x^{\boldsymbol{\sigma}}) = x^{\boldsymbol{\sigma}}_A \parallel x^{\boldsymbol{\tau}}$.

Proof. We first define the projection maps, as

$$\begin{array}{rcccl} \Pi_{\boldsymbol{\sigma}} & : & \boldsymbol{\tau} \circledast \boldsymbol{\sigma} & \to & \boldsymbol{\sigma} \parallel C \\ & & \varphi & \mapsto & \pi_1(\mathsf{top}(\varphi)) \end{array} \qquad \begin{array}{rcccl} \Pi_{\boldsymbol{\tau}} & : & \boldsymbol{\tau} \circledast \boldsymbol{\sigma} & \to & A \parallel \boldsymbol{\tau} \\ & & \varphi & \mapsto & \pi_2(\mathsf{top}(\varphi))\,, \end{array}$$

and check the two conditions for these to be maps of event structures. For *rule-abiding*, consider $x \in \mathscr{C}(\boldsymbol{\tau} \circledast \boldsymbol{\sigma})$. By Proposition 6.2.7, $\cup x \in \mathscr{B}(\boldsymbol{\sigma}, \boldsymbol{\tau})$. But furthermore,

$$\cup x \,:\, \Pi_{\boldsymbol{\sigma}} x \simeq \Pi_{\boldsymbol{\tau}} x\,. \tag{6.5}$$

Indeed, if $m \in \Pi_{\boldsymbol{\sigma}} x$ there is some $\varphi \in x$ such that $m = \pi_1(\mathsf{top}(\varphi))$, but then clearly $m \in \mathsf{dom}(\cup x)$. Reciprocally, if $m \in \mathsf{dom}(\cup x)$, this means there is $\varphi \in x$ such that $(m, n) \in \varphi$ for some n. But then, considering the set defined as

$$[(m,n)]_\varphi = \{(m', n') \in \varphi \mid (m', n') \leq_\varphi (m, n)\}\,,$$

one can check that $[(m,n)]_\varphi \in \mathscr{B}^\top(\sigma,\tau)$, and $[(m,n)]_\varphi \subseteq \varphi$ by construction, so $[(m,n)]_\varphi \in x$ as x is *down-closed*. But so, $m = \Pi_\sigma[(m,n)]_\varphi$ so $m \in \Pi_\sigma\, x$ as required. It also follows that $\Pi_\sigma x \in \mathscr{C}(\sigma \parallel C)$, and using (6.5) and Proposition 6.2.7,

$$\begin{aligned}\Pi_\sigma\,(x^\tau \circledast x^\sigma) &= \Pi_\sigma\,(\bar{\chi}_{\sigma,\tau}\,\varphi[x^\sigma, x^\tau]) \\ &= \mathsf{dom}(\cup(\bar{\chi}_{\sigma,\tau}\,\varphi[x^\sigma, x^\tau])) \\ &= \mathsf{dom}(\varphi[x^\sigma, x^\tau])\end{aligned}$$

which is $x^\sigma \parallel x^\tau_C$ – and likewise, $\Pi_\tau(x^\tau \circledast x^\sigma) = x^\sigma_A \parallel x^\tau \in \mathscr{C}(A \parallel \tau)$. For local injectivity, assume that $\varphi, \psi \in x$ are such that $\Pi_\sigma\varphi = \Pi_\sigma\psi$, write it m. But $\varphi, \psi \subseteq \cup x \in \mathscr{B}(\sigma,\tau)$, and there is a unique n such that $(m,n) \in \cup x$ as it is a bijection. But we must have $\varphi = [(m,n)]_{\cup x}$ and likewise for ψ, so $\varphi = \psi$.

That the diagram commutes and satisfies the universal property of the diagram is routine, using the above along with Lemma 6.1.11 (the conditions for mapification come from Lemmas 6.2.9 and 6.2.10) and Proposition 6.2.7. □

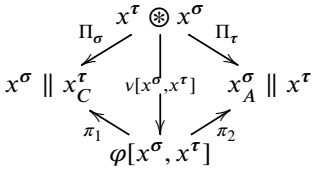
In fact, the category **ES** of event structures and their maps has all pullbacks. The construction of $\tau \circledast \sigma$ given here closely follows the construction of pullbacks in **ES**, which in turn is a restriction of the usual synchronizing product of event structures. So this is old technology here, essentially the same used by Winskel in the early 80s to give semantics to parallel composition of CCS processes [Winskel, 1982].

In the sequel we shall not exploit the universal property of this pullback, but we will often refer to its projections: after this section we shall often treat the interaction construction as a black box. The *projections* will be our main device to read back an event $p \in \tau \circledast \sigma$ as a synchronization of $\Pi_\sigma p \in \sigma \parallel C$ and $\Pi_\tau p \in A \parallel \tau$.

Finally, we have seen that configurations of the interaction correspond to secured bijections. But the connection is even tighter: both configurations and secured bijections are partially ordered sets, and it will be important for later on to keep in mind that this correspondence is, locally, an order-isomorphism. This is expressed by:

Lemma 6.2.12. *Consider $x^\sigma \in \mathscr{C}(\sigma)$ and $x^\tau \in \mathscr{C}(\tau)$ causally compatible.*

Then, there is an order-isomorphism $\nu[x^\sigma, x^\tau] : x^\tau \circledast x^\sigma \cong \varphi[x^\sigma, x^\tau]$ such that



commutes, in the category of finite sets and bijections. Moreover, this assignment is monotone, in the sense that if $x^\sigma \subseteq y^\sigma$ and $x^\tau \subseteq y^\tau$, then $\nu[x^\sigma, x^\tau] \subseteq \nu[y^\sigma, y^\tau]$.

Proof. We set the order-isomorphism $\nu[x^\sigma, x^\tau]$ as follows:

$$\begin{aligned}\nu[x^\sigma, x^\tau] \quad : \quad x^\tau \circledast x^\sigma &\to \varphi[x^\sigma, x^\tau] \\ \psi &\mapsto \mathsf{top}(\psi)\,,\end{aligned}$$

it is a direct that this defines an order-iso, and that this assignment is monotone. □

The interaction. At last, this provides us with the ingredients for the interaction.

Proposition 6.2.13. *There is an interaction $\tau \circledast \sigma$, unique up to iso, s.t. there is*

$$(- \circledast -) : \{(x^\tau, x^\sigma) \in \mathscr{C}(\tau) \times \mathscr{C}(\sigma) \mid x^\sigma, x^\tau \text{ causally compatible}\} \simeq \mathscr{C}(\tau \circledast \sigma)$$

an order-iso s.t. $\partial_{\tau\circledast\sigma}(x^\tau \circledast x^\sigma) = x^\sigma_A \parallel x_B \parallel x^\tau_C$ for all x^σ, x^τ causally compatible.

Proof. Existence. The event structure $\tau \circledast \sigma$ is defined in Lemma 6.2.11, with display map $\partial_{\tau\circledast\sigma} : \tau\circledast\sigma \to A \parallel B \parallel C$ defined as either way around the square. The order-iso for configurations comes from Proposition 6.2.8.

Uniqueness. Immediate from the order-isos, by Lemma 6.1.12. □

Here, an isomorphism of interactions simply means an isomorphism of event structures commuting with the display maps. The construction of $\tau\circledast\sigma$ is elaborate – there is no going around it; but fortunately, Proposition 6.2.13 packages most of what we need about the interaction, and may often be used as a black box. It is suspiciously close to Proposition 6.2.1, which may be surprising since this is not composition but interaction, without hiding. We shall see that the restriction to +-covered configurations in Proposition 6.2.1 does a lot to span the distance between interaction and composition.

Structure of causality. Proposition 6.2.13 gives a neat *state-based* description of the interaction, putting the focus on configurations. This is sufficient for many purposes, but it is sometimes necessary to reason on individual events of the interaction.

As explained before, any event $p \in \tau \circledast \sigma$ may be regarded as a synchronization between its projections $\Pi_\sigma p \in \sigma \parallel C$ and $\Pi_\tau p \in A \parallel \tau$, along with a causal history for this synchronization. These projections help us classify every $p \in \tau \circledast \sigma$ into:

(1) $\Pi_\sigma(p) = (1, s)$ with $s \in \sigma$, and $\Pi_\tau(p) = (1, a)$ with $a \in A$,
(2) $\Pi_\sigma(p) = (1, s)$ with $s \in \sigma$, and $\Pi_\tau(p) = (2, t)$ with $t \in \tau$,
(3) $\Pi_\sigma(p) = (2, c)$ with $c \in C$, and $\Pi_\tau(p) = (2, t)$ with $t \in \tau$.

In case *(1)*, the only relevant projection is $\Pi_\sigma(p) = (1, s)$ as $\Pi_\tau(p) = \partial_\sigma(s)$. We write $p_\sigma = s$ and p_τ is undefined, and we say that p **occurs in** A. In case *(3)*, the relevant projection is $\Pi_\tau(p) = (2, t)$ as $\Pi_\sigma(p) = \partial_\tau(t)$. We write $p_\tau = t$ and p_σ is undefined, and p **occurs in** C. Finally, in case *(2)* the two projections $\Pi_\sigma(p) = (1, s)$ and $\Pi_\tau(p) = (2, t)$ are relevant, but we must have $\partial_\sigma(s) = (2, b)$ and $\partial_\tau(t) = (1, b)$ for some $b \in B$. We write $p_\sigma = s$, $p_\tau = t$, we say that p **occurs in** B. If p occurs in A or C we also say that it is **visible**, while if it occurs in B it is **synchronized** or **invisible**.

Immediate causal links between events of $\tau \circledast \sigma$ must originate from σ or τ:

Lemma 6.2.14. *Consider $p, p' \in \tau \circledast \sigma$, and assume $p \rightarrowtriangle_{\tau\circledast\sigma} p'$.*
Then $p_\sigma \rightarrowtriangle_\sigma p'_\sigma$ or $p_\tau \rightarrowtriangle_\tau p'_\tau$, where p_σ, p_τ are defined whenever used.

Proof. Consider $x^\tau \circledast x^\sigma \in \mathscr{C}(\tau \circledast \sigma)$ such that $p, p' \in x^\tau \circledast x^\sigma$. Write $\mathsf{top}(p) = (m, n)$ and $\mathsf{top}(p') = (m', n')$. By Lemma 6.2.12, we have an immediate causal link

$$(m, n) \rightarrowtriangle_{\varphi[x^\sigma, x^\tau]} (m', n'),$$

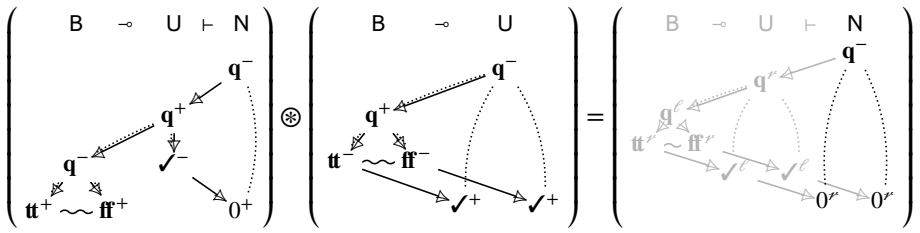


Figure 6.4: Example of an interaction with polarity information

which easily implies, by definition of $\leq_{\varphi[x^\sigma,x^\tau]}$, that $m \rightarrowtriangle_{\sigma\|C} m'$ or $n \rightarrowtriangle_{A\|\tau} n'$.

If $m \rightarrowtriangle_{\sigma\|C} m'$, then either $m = (1,s), m' = (1,s')$ and $s \rightarrowtriangle_\sigma s'$ with $s = p_\sigma$ and $s' = p'_\sigma$ defined as required; or, $m = (2,c)$ and $m' = (2,c')$ with $c \rightarrowtriangle_C c'$. In that case, $n = (2,t)$ and $n' = (2,t')$ and $t = p_\tau$ and $t' = p'_\tau$ defined. Moreover, $c \rightarrowtriangle_C c'$ implies $\partial_\tau(t) \rightarrowtriangle_{B\vdash C} \partial_\tau(t')$, so $t <_\tau t'$ by Lemma 6.1.8. If this causal link was not immediate, this would directly contradict $(m,n) \rightarrowtriangle_{\varphi[x^\sigma,x^\tau]} (m',n')$ – so $p_\tau \rightarrowtriangle_\tau p'_\tau$ as required. □

For σ and τ *strategies*, we can track down the responsible of a move via a polarity analysis. Events of $\tau \circledast \sigma$ cannot sensibly be assigned a polarity in $\{-,+\}$, as σ and τ disagree on B. A more useful polarity is $\mathrm{pol}_{\tau\circledast\sigma} : |\tau \circledast \sigma| \to \{-,\ell,r\}$ set as:

$$\begin{array}{lcll} \mathrm{pol}_{\tau\circledast\sigma}(p) & = & \ell & \text{if } p_\sigma \text{ is defined and } \mathrm{pol}_\sigma(p_\sigma) = +, \\ \mathrm{pol}_{\tau\circledast\sigma}(p) & = & r & \text{if } p_\tau \text{ is defined and } \mathrm{pol}_\tau(p_\tau) = +, \\ \mathrm{pol}_{\tau\circledast\sigma}(p) & = & - & \text{otherwise.} \end{array}$$

We show in Figure 6.4 a version of Figure 6.1 with those polarities. Then:

Lemma 6.2.15. *If $\sigma : A \vdash B$ and $\tau : B \vdash C$ are strategies and $p \rightarrowtriangle_{\tau\circledast\sigma} p'$, then*

(1) *if $\mathrm{pol}_{\tau\circledast\sigma}(p') = \ell$, then $p_\sigma \rightarrowtriangle_\sigma p'_\sigma$,*
(2) *if $\mathrm{pol}_{\tau\circledast\sigma}(p') = r$, then $p_\tau \rightarrowtriangle_\tau p'_\tau$,*
(3) *if $\mathrm{pol}_{\tau\circledast\sigma}(p') = -$, then $\partial_{\tau\circledast\sigma}(p) \rightarrowtriangle_{A\|B\|C} \partial_{\tau\circledast\sigma}(p')$*

where again, $p_\sigma, p'_\sigma, p_\tau$ and p'_τ are defined whenever used.

Proof. (1) By Lemma 6.2.14, p_σ, p'_σ defined and $p_\sigma \rightarrowtriangle_\sigma p'_\sigma$ and we are done; or p_τ, p'_τ defined and $p_\tau \rightarrowtriangle_\tau p'_\tau$. Since $\mathrm{pol}_{\tau\circledast\sigma}(p') = \ell$, $\mathrm{pol}_\tau(p'_\tau) = -$. By *courteous*, $\partial_\tau(p_\tau) \rightarrowtriangle_{B\vdash C} \partial_\tau(p'_\tau)$; hence m occurs in B and $\partial_\sigma(p_\sigma) \rightarrowtriangle_{A\vdash B} \partial_\sigma(p'_\sigma)$. By Lemma 6.1.8, $p_\sigma <_\sigma p'_\sigma$, and it is direct that the causality must be immediate. *(2)* is symmetric.

(3) Assume p' occurs in A, the other case is symmetric. In that case only p'_σ is defined, so Lemma 6.2.14 entails that p_σ is defined and $p_\sigma \rightarrowtriangle_\sigma p'_\sigma$. But $\mathrm{pol}_\sigma(p'_\sigma) = -$, so by courtesy $\partial_\sigma(p_\sigma) \rightarrowtriangle_{A\vdash B} \partial_\sigma(p'_\sigma)$, from which the conclusion follows. □

Next, we perform the hiding and introduce the *composition* of prestrategies.

6.2.2 Composition of prestrategies

Hiding. Composition consists simply in removing all synchronized events:

Definition 6.2.16. *The **composition** of $\sigma : A \vdash B$ and $\tau : B \vdash C$ comprises:*

$$\begin{array}{rcl} |\tau \odot \sigma| & = & \{p \in \tau \circledast \sigma \mid p \text{ occurs in } A \text{ or } C\}\,, \\ p_1 \leq_{\tau\odot\sigma} p_2 & \Leftrightarrow & p_1 \leq_{\tau\circledast\sigma} p_2\,, \\ p_1 \#_{\tau\odot\sigma} p_2 & \Leftrightarrow & p_1 \#_{\tau\circledast\sigma} p_2\,. \end{array}$$

with display map $\partial_{\tau\odot\sigma} : |\tau \odot \sigma| \to |A \vdash C|$ obtained as restriction of $\partial_{\tau\circledast\sigma}$.

It is straightforward that $\tau \odot \sigma$ is an event structure. In Figure 6.4, the composition simply keeps the events in black. This means that it has two conflicting positive events, both corresponding to 0^+: the model records the point of non-deterministic branching even when it brings no observable difference.

Though this does not appear in pictures, we insist that events of $\tau \odot \sigma$ *are* certain events of $\tau \circledast \sigma$. Thus an event of the composition always carries a unique causal explanation: *itself*, regarded as an event of the interaction $\tau \circledast \sigma$. Accordingly, we set:

Definition 6.2.17. *Consider $x \in \mathscr{C}(\tau \odot \sigma)$. Its **interaction witness** is defined as*

$$[x]_{\tau\circledast\sigma} = \{p \in \tau \circledast \sigma \mid \exists p' \in x,\ p \leq_{\tau\circledast\sigma} p'\} \in \mathscr{C}(\tau \circledast \sigma)$$

its down-closure in $\tau \circledast \sigma$.

This entails – and it shall play a crucial role in the development – that configurations of the composition are in one-to-one correspondence with those configurations of the interaction whose maximal events are visible. From this idea, we get:

Proposition 6.2.18. *There is an order-isomorphism*

$$(- \odot -) : \{(x^\tau, x^\sigma) \in \mathscr{C}(\tau) \times \mathscr{C}(\sigma) \mid x^\sigma, x^\tau \text{ minimal } \textit{caus. comp.}\} \simeq \mathscr{C}(\tau \odot \sigma)$$

*where x^σ, x^τ **minimal** means that the maximal elements of $\varphi[x^\sigma, x^\tau]$ occur in A or C.*

Proof. We decompose the order-isomorphism in two steps.

As a first step, we note that the isomorphism of Proposition 6.2.13 refines to:

$$(- \circledast -) : \{(x^\tau, x^\sigma) \in \mathscr{C}(\tau) \times \mathscr{C}(\sigma) \mid x^\sigma, x^\tau \text{ min. caus. comp.}\} \simeq \mathscr{C}^v(\tau \circledast \sigma)$$

where $\mathscr{C}^v(\tau \circledast \sigma)$ refers to configurations whose maximal events are *visible*, *i.e.* occur in A or C – this is clear by Lemma 6.2.12. Next, we have the order-iso formed with:

$$\begin{array}{rcl} \mathscr{C}^v(\tau \circledast \sigma) & \to & \mathscr{C}(\tau \odot \sigma) \\ x & \mapsto & x \cap V \end{array} \qquad\qquad \begin{array}{rcl} \mathscr{C}(\tau \odot \sigma) & \to & \mathscr{C}^v(\tau \circledast \sigma) \\ x & \mapsto & [x]_{\tau\circledast\sigma} \end{array}$$

where V denotes visible events – it is straightforward that this defines an order-iso. □

This shows how even through the hiding operation, configurations of the composition can still be regarded as certain pairs of configurations of the compound strategies. We are certainly starting to get close to our target statement of Proposition 6.2.1.

Note that the operation from left to right is well-defined even if $x^{\sigma} \in \mathscr{C}(\sigma)$ and $x^{\tau} \in \mathscr{C}(\tau)$ are causally compatible but not *minimal* causally compatible. In that case we shall still write $x^{\tau} \odot x^{\sigma} = (x^{\tau} \circledast x^{\sigma}) \cap V \in \mathscr{C}(\tau \odot \sigma)$, even though x^{σ} and x^{τ} cannot necessarily be recovered if the interaction has some maximal synchronized events.

The next lemma states a simple property of this construction, useful for reference:

Lemma 6.2.19. *For $x^{\sigma} \in \mathscr{C}(\sigma)$ and $x^{\tau} \in \mathscr{C}(\tau)$ minimal causally compatible, we have*

$$x^{\tau} \odot x^{\sigma} = (x^{\tau} \circledast x^{\sigma}) \cap V \qquad\qquad x^{\tau} \circledast x^{\sigma} = [x^{\tau} \odot x^{\sigma}]_{\tau \circledast \sigma},$$

where V is the set of visible events.

Proof. By def. of $x^{\tau} \odot x^{\sigma}$ through the order-iso $\mathscr{C}^{v}(\tau \circledast \sigma) \simeq \mathscr{C}(\tau \odot \sigma)$ above. □

It will also be useful later on that this is compatible with unions:

Lemma 6.2.20. *Consider $x^{\sigma}, y^{\sigma} \in \mathscr{C}(\sigma)$ and $x^{\tau}, y^{\tau} \in \mathscr{C}(\tau)$ satisfying either of the conditions of Lemma 6.2.6. Then, we have the equality:*

$$(x^{\tau} \odot x^{\sigma}) \cup (y^{\tau} \odot y^{\sigma}) = (x^{\tau} \cup y^{\tau}) \odot (x^{\sigma} \cup y^{\sigma}).$$

Proof. By Lemma 6.2.10, we have $(x^{\tau} \cup y^{\tau}) \circledast (x^{\sigma} \cup y^{\sigma}) = (x^{\tau} \circledast x^{\sigma}) \cup (y^{\tau} \circledast y^{\sigma})$, thus $(x^{\tau} \cup y^{\tau}) \odot (x^{\sigma} \cup y^{\sigma}) = (x^{\tau} \odot x^{\sigma}) \cup (y^{\tau} \odot y^{\sigma})$ by restricting to visible events. □

Finally, we relate immediate causality in the interaction and composition:

Lemma 6.2.21. *Consider $p, p' \in \tau \odot \sigma$. If $p \rightarrowtriangle_{\tau \odot \sigma} p'$, then we have*

$$p \rightarrowtriangle_{\tau \circledast \sigma} q_1 \rightarrowtriangle_{\tau \circledast \sigma} \cdots \rightarrowtriangle_{\tau \circledast \sigma} q_n \rightarrowtriangle_{\tau \circledast \sigma} p'$$

for some $q_1, \ldots, q_n \in \tau \circledast \sigma$ synchronized.

Proof. Straightforward by definition. □

We wrap up with the main proposition for composition of prestrategies:

Proposition 6.2.22. *Consider $\sigma : A \vdash B$ and $\tau : B \vdash C$ prestrategies.*
Then there is a prestrategy $\tau \odot \sigma$, unique up to iso, s.t. there is an order-iso:

$$(- \odot -) : \{(x^{\tau}, x^{\sigma}) \in \mathscr{C}(\tau) \times \mathscr{C}(\sigma) \mid x^{\sigma}, x^{\tau} \text{ min. caus. comp.}\} \simeq \mathscr{C}(\tau \odot \sigma)$$

s.t. for all $x^{\sigma} \in \mathscr{C}(\sigma), x^{\tau} \in \mathscr{C}(\tau)$ min. caus. comp., $\partial_{\tau \odot \sigma}(x^{\tau} \odot x^{\sigma}) = x^{\sigma}_{A} \vdash x^{\tau}_{C}$.

$$
\begin{array}{llllll}
(\partial_\sigma \parallel C \parallel D)(m) & \lhd_\sigma & (\partial_\sigma \parallel C \parallel D)(m') & \text{where} & m <_{\sigma\parallel C\parallel D} m' & (m,m' \in x^\sigma \parallel x_C \parallel x_D), \\
(A \parallel \partial_\tau \parallel D)(m) & \lhd_\tau & (A \parallel \partial_\tau \parallel D)(m') & \text{where} & m <_{A\parallel\tau\parallel D} m' & (m,m' \in x_A \parallel x^\tau \parallel x_D), \\
(A \parallel B \parallel \partial_\delta)(m) & \lhd_\delta & (A \parallel B \parallel \partial_\delta)(m') & \text{where} & m <_{A\parallel B\parallel\delta} m' & (m,m' \in x_A \parallel x_B \parallel x^\delta),
\end{array}
$$

Figure 6.5: Ternary causal compatibility

Proof. Existence. The event structure, along with the display map on events, are defined in Definition 6.2.16, with order-isomorphism defined in Proposition 6.2.18. We must check that $\partial_{\tau\odot\sigma}$ is a map of event structures. By Proposition 6.2.13, $\partial_{\tau\circledast\sigma}(x^\tau \circledast x^\sigma) = x^\sigma_A \parallel x_B \parallel x^\tau_C$, hence by definition of $\partial_{\tau\odot\sigma}$, $\partial_{\tau\odot\sigma}(x^\tau \odot x^\sigma) = x^\sigma_A \vdash x^\tau_C$. Now as $\partial_{\tau\circledast\sigma}$ is a map of event structures, it induces a bijection between $x^\tau \circledast x^\sigma$ and $x^\sigma_A \parallel x_B \parallel x^\tau_C$, which by definition of $\tau \odot \sigma$ restricts to a bijection between $x^\tau \odot x^\sigma$ and $x^\sigma_A \vdash x^\tau_C$, so

$$\mathsf{card}(x^\tau \odot x^\sigma) = \mathsf{card}(x^\sigma_A) + \mathsf{card}(x^\tau_C)$$

ensuring that $\partial_{\tau\odot\sigma}$ preserves cardinal. By Lemma 6.2.20 it follows that it also preserves unions, and hence forms a map of event structures by Lemma 6.1.11.

Uniqueness up to strong iso is straightforward by Lemma 6.1.12. □

6.2.3 The associator

We have defined the composition of prestrategies; not yet that of strategies, as we have not yet established that composition preserves receptivity and courtesy. Before doing that we shall study *associativity* of composition, as it turns out that it holds independently of the additional conditions for strategies. Thus fix now prestrategies $\sigma : A \vdash B$, $\tau : B \vdash C$ and $\delta : C \vdash D$. We aim to show that composition is associative up to isomorphism, *i.e.* that there is an isomorphism of prestrategies, called the *associator*:

$$a_{\sigma,\tau,\delta} : (\delta \odot \tau) \odot \sigma \cong \delta \odot (\tau \odot \sigma)\,.$$

Proposition 6.2.22 makes associativity seem almost trivial: as configurations of the composition are identified to certain pairs of configurations, mapping $(\delta \odot \tau) \odot \sigma$ to $\delta\odot(\tau\odot\sigma)$ is only a matter of sending $(x^\delta \odot x^\tau) \odot x^\sigma$ to $x^\delta \odot (x^\tau \odot x^\sigma)$, *i.e.* by associativity of the set-theoretic cartesian product – as for associativity for the composition of spans. But as Proposition 6.2.22 makes formal, composition of strategies is composition of spans, *plus deadlocks*, captured by the *causal compatibility* constraint; and we must also prove that causal compatibility is compatible with associativity.

For this, it is cleaner to introduce *causally compatible triples*:

Definition 6.2.23. *Consider $x^\sigma \in \mathscr{C}(\sigma)$, $x^\tau \in \mathscr{C}(\tau)$ and $x^\delta \in \mathscr{C}(\delta)$. We set conditions:*

matching:	*if $x^\sigma_B = x^\tau_B = x_B$ and $x^\tau_C = x^\delta_C = x_C$,*
causally compatible:	*if they are matching and the relation $\lhd = \lhd_\sigma \cup \lhd_\tau \cup \lhd_\delta$ on $x_A \parallel x_B \parallel x_C \parallel x_D$ defined in Figure 6.5, is acyclic.*
minimal:	*if events maximal for $\lhd^*$ occur in A or D.*

where $\partial_\sigma x^\sigma = x_A \parallel x_B^\sigma$, $\partial_\tau x^\tau = x_B^\tau \parallel x_C^\tau$, *and* $\partial_\delta x^\delta = x_C^\delta \parallel x_D$.

Crucially, ternary causal compatibility is equivalent to the two skewed versions:

Lemma 6.2.24. *For* $x^\sigma \in \mathscr{C}(\sigma), x^\tau \in \mathscr{C}(\tau)$, $x^\delta \in \mathscr{C}(\delta)$, *the following are equivalent:*

(1) $x^\sigma, x^\tau, x^\delta$ *are minimal causally compatible,*
(2) x^σ, x^τ *are min. caus. comp. and* $x^\tau \odot x^\sigma, x^\delta$ *are min. caus. comp.,*
(3) x^τ, x^δ *are min. caus. comp. and* $x^\sigma, x^\delta \odot x^\tau$ *are min. caus. comp.*

Proof. (1) ⇒ (2). For x^σ, x^τ causally compatible, a cycle in $\varphi[x^\sigma, x^\tau]$ routinely corresponds to a cycle in $x_A \parallel x_B \parallel x_C$ for the relation $\lhd_\sigma \cup \lhd_\tau$ defined as:

$$\begin{array}{llllll} (\partial_\sigma \parallel C)(m) & \lhd_\sigma & (\partial_\sigma \parallel C)(m') & \text{where} & m <_{\sigma\parallel C} m' & (m, m' \in x^\sigma \parallel x_C), \\ (A \parallel \partial_\tau)(m) & \lhd_\tau & (A \parallel \partial_\tau)(m') & \text{where} & m <_{A\parallel\tau} m' & (m, m' \in x_A \parallel x^\tau) \end{array}$$

which immediately transports to a cycle in $\lhd$ on $x_A \parallel x_B \parallel x_C \parallel x_D$, contradiction. For minimality, consider m maximal in $x_A \parallel x_B \parallel x_C$ for $\lhd_\sigma \cup \lhd_\tau$; assume, seeking a contradiction, that it is in B. Then this event considered in $x_A \parallel x_B \parallel x_C \parallel x_D$, still denoted by m by abuse of notation, cannot be maximal for $\lhd$ by minimality. Thus there is $m \lhd n$ for n in $x_A \parallel x_B \parallel x_C \parallel x_D$. But by construction of $\lhd$, n must be in A, B, C and m $(\lhd_\sigma \cup \lhd_\tau)$ n in $x_A \parallel x_B \parallel x_C$, contradiction. Hence, x^σ, x^τ minimal.

For $x^\tau \odot x^\sigma, x^\delta$ causally compatible, a cycle in $\varphi[x^\tau \odot x^\sigma, x^\delta]$ corresponds to a cycle in $x_A \parallel x_C \parallel x_D$ for the relation $\lhd_{\tau\odot\sigma} \cup \lhd_\delta$ obtained as above. But this immediately gives a cycle in $x_A \parallel x_B \parallel x_C \parallel x_D$ for $(\lhd_\sigma \cup \lhd_\tau)^* \cup \lhd_\delta$, which gives a cycle in $x_A \parallel x_B \parallel x_C \parallel x_D$ for $\lhd$, contradiction. For minimality, an event m maximal for $\lhd_{\sigma,\tau}$ and $\lhd_\delta$ cannot be in C: otherwise there is n in A, D such that $m \lhd^* n$, but then it is straightforward that m $(\lhd_{\sigma,\tau} \cup \lhd_\delta)^*$ n, contradicting its maximality.

(2) ⇒ (1). Consider a cycle $m_1 \lhd \cdots \lhd m_k \lhd m_1$. If the cycle remains in A, B, C, we get a contradiction to x^σ, x^τ causally compatible. Otherwise, removing all moves in B, we get a cycle in $(\lhd_{\sigma,\tau} \cup \lhd_\delta)$ hence a contradiction to $x^\tau \odot x^\sigma, x^\delta$ causally compatible. For minimality, consider m maximal for $\lhd^*$. By minimality of x^σ, x^τ it is not in B, and by minimality of $x^\tau \odot x^\sigma, x^\delta$ it cannot be in C; hence it is in A or D.

(1) ⇒ (3), (3) ⇒ (1). Symmetric to the two implications above. □

Altogether, this gives us all the ingredients required for the associator:

Proposition 6.2.25. *Consider* $\sigma : A \vdash B, \tau : B \vdash C$ *and* $\delta : C \vdash D$ *prestrategies. Then there is a unique isomorphism, the* ***associator****:*

$$a_{\sigma,\tau,\delta} : (\delta \odot \tau) \odot \sigma \cong \delta \odot (\tau \odot \sigma),$$

s.t. for all $x^\sigma, x^\tau, x^\delta$ *min. caus. comp.,* $a_{\sigma,\tau,\delta}((x^\delta \odot x^\tau) \odot x^\sigma) = x^\delta \odot (x^\tau \odot x^\sigma)$.

Proof. The two clauses serve as the definition of the action of $a_{\delta,\tau,\sigma}$ on configurations, via Proposition 6.2.22 and Lemma 6.2.24. Compatibility with display maps is from the description of the display map of composition in Proposition 6.2.22. To lift this to a map, we observe that preservation of cardinal is clear by local injectivity and preservation of display maps, and preservation of unions follows from Lemma 6.2.20. Finally, the same arguments holds for the inverse, therefore forming an isomorphism. □

Composition of prestrategies is associative, but there does not seem to be an identity turning it into a (bi)category – the obvious candidate, the *copycat strategy* that we shall see later on, is only neutral for composition with respect to *strategies*.

6.3 Composition of Strategies

Now we focus on *strategies*, *i.e.* those prestrategies satisfying the additional conditions of Definition 6.1.6. In this section, we aim in particular to prove Proposition 6.2.1.

6.3.1 Composition of Strategies

Thus, we now turn to the composition of strategies. Our first objective is to establish that they are indeed stable under composition.

Proposition 6.3.1. *Consider $\sigma : A \vdash B$ and $\tau : B \vdash C$ strategies.*
Then, $\tau \odot \sigma : A \vdash C$ is a strategy.

Proof. By Proposition 6.2.22, we have $\tau \odot \sigma : A \vdash C$ a prestrategy.

Courteous. Let $p \rightarrowtriangle_{\tau\odot\sigma} p'$ s.t. $\mathrm{pol}(p) = +$ or $\mathrm{pol}(p') = -$. By Lemma 6.2.21,

$$p \rightarrowtriangle_{\tau\circledast\sigma} q_1 \rightarrowtriangle_{\tau\circledast\sigma} \cdots \rightarrowtriangle_{\tau\circledast\sigma} q_n \rightarrowtriangle_{\tau\circledast\sigma} p'$$

for some $q_1, \ldots, q_n \in \tau \circledast \sigma$ synchronized.

We show that in fact $n = 0$ and $p \rightarrowtriangle_{\tau\circledast\sigma} p'$. If $\mathrm{pol}(p) = +$, then $\mathrm{pol}_{\tau\circledast\sigma}(q_1) \in \{\ell, r\}$ – assume *w.l.o.g.* that it is ℓ. Then by Lemma 6.2.15, $p_\sigma, (q_1)_\sigma$ are defined and $p_\sigma \rightarrowtriangle_\sigma (q_1)_\sigma$. But by courtesy, this entails $\partial_\sigma(p_\sigma) \rightarrowtriangle_{A\vdash B} \partial_\sigma((q_1)_\sigma)$, impossible for q_1 synchronized. Hence $n = 0$, and $p \rightarrowtriangle_{\tau\circledast\sigma} p'$. If $\mathrm{pol}(p') = -$, then again by Lemma 6.2.15, $\partial_{\tau\circledast\sigma}(q_n) \rightarrowtriangle_{A\|B\|C} \partial_{\tau\circledast\sigma}(p')$, contradicting q_n synchronized, so $p \rightarrowtriangle_{\tau\circledast\sigma} p'$.

Now, by Lemma 6.2.14, $p_\sigma \rightarrowtriangle_\sigma p'_\sigma$ or $p_\tau \rightarrowtriangle_\sigma p'_\tau$, say *w.l.o.g.* the former. But then $\partial_\sigma(p_\sigma) \rightarrowtriangle_{A\vdash B} \partial_\sigma(p'_\sigma)$, from which $\partial_{\tau\odot\sigma}(p) \rightarrowtriangle_{A\vdash C} \partial_{\tau\odot\sigma}(p')$ follows.

Receptive. Consider $x^\tau \odot x^\sigma \in \mathscr{C}(\tau \odot \sigma)$, and $\partial_{\tau\odot\sigma}(x^\tau \odot x^\sigma) = x^\sigma_A \parallel x^\tau_C \vdash_{A\vdash C} c^-$, assuming *w.l.o.g.* it is in C. For *existence*, by receptivity of τ we have $x^\tau \vdash_\tau t$ such that $\partial_\tau(t) = c$. Writing $y^\tau = x^\tau \uplus \{t\}$, it is direct that x^σ, y^τ is minimal causally compatible, and $x^\tau \odot x^\sigma \mathrel{-\!\!\subset} y^\tau \odot x^\sigma$ provides the required extension, via Proposition 6.2.22. For *uniqueness*, assume $x^\tau \odot x^\sigma \mathrel{-\!\!\subset} z^\tau \odot z^\sigma$ via some event p such that $\partial_{\tau\odot\sigma}(p) = c^-$. So,

$$x^\tau \circledast x^\sigma \subseteq u^\tau \circledast u^\sigma \overset{p}{\mathrel{-\!\!\subset}} z^\tau \circledast z^\sigma$$

where $x^\tau \circledast x^\sigma \subseteq u^\tau \circledast u^\sigma$ adds a set X of synchronized events, such that for all $q \in X$, $q \leq_{\tau\circledast\sigma} p$. But for any $q \rightarrowtriangle_{\tau\circledast\sigma} p$, we have q in C by Lemma 6.2.15, so X is empty. As p is in C, this entails $x^\sigma = u^\sigma = z^\sigma$ and $x^\tau = u^\tau \mathrel{-\!\!\subset} z^\tau$. So $z^\tau = x^\tau \uplus \{t'\}$ such that $x^\tau \vdash_\tau t'$ and $\partial_\tau(t') = c$, so $t = t'$ by *receptive* for τ; hence $y^\tau \odot x^\tau = z^\tau \odot z^\sigma$. □

We have a well-defined composition operation for *strategies*, but we have not yet arrived at the claimed Proposition 6.2.1. For that, it still remains to investigate the effect of ignoring the trailing Opponent moves, restricting to *+-covered configurations*.

6.3.2 Strategies and +-covered configurations

In traditional alternating game semantics, strategies can be defined in two different ways: as done in Definition 3.1.2 by taking plays finishing by moves of arbitrary polarity, but requiring a *receptivity* property for strategies; or – as is more standard – by only considering even-length plays in strategies, that is, plays ending with Player moves. For concurrent strategies, a similar effect can be obtained via *+-covered configurations*:

Definition 6.3.2. *Consider $\sigma : A$ a (pre)strategy on game A. A configuration $x \in \mathscr{C}(\sigma)$ is **+-covered** if for any $s \in x$ such that s is $\leq_\sigma$-maximal in x, $\mathrm{pol}_\sigma(s) = +$.*

We write $\mathscr{C}^+(\sigma)$ for the +-maximal configurations of σ.

We shall see that it suffices to compare strategies on their +-covered configurations. To show this, it is convenient to rely not on the original definition of strategies in Definition 6.1.6, but on the following more "big-step" characterization:

Proposition 6.3.3. *Consider $\sigma : A$ a prestrategy on game A. Then it is a strategy iff*

$$\begin{array}{rl} -\text{-discrete opfibration:} & \text{for all } x^\sigma \in \mathscr{C}(\sigma), \text{ for all } \partial_\sigma(x^\sigma) \subseteq^- y_A \in \mathscr{C}(A), \\ & \text{there exists a unique } x^\sigma \subseteq y^\sigma \in \mathscr{C}(\sigma) \text{ s.t. } \partial_\sigma(y^\sigma) = y_A. \\ +\text{-discrete fibration:} & \text{for all } x^\sigma \in \mathscr{C}(\sigma), \text{ for all } \partial_\sigma(x^\sigma) \supseteq^+ y_A \in \mathscr{C}(A), \\ & \text{there exists a unique } x^\sigma \supseteq y^\sigma \in \mathscr{C}(\sigma) \text{ s.t. } \partial_\sigma(y^\sigma) = y_A. \end{array}$$

where $x \subseteq^p y$ (for $p \in \{-,+\}$) means $x \subseteq y$ and $\mathrm{pol}_A(y \setminus x) \subseteq \{p\}$.

Proof. −-discrete opfibration. Existence is clear by repeated applications of *receptive*. For *uniqueness*, consider $x^\sigma \subseteq y^\sigma, z^\sigma \in \mathscr{C}(\sigma)$ s.t. $\partial_\sigma(y^\sigma) = \partial_\sigma(z^\sigma) = y_A$. Take $m_1 \in y^\sigma \setminus x^\sigma$ and $m_2 \in z^\sigma \setminus x^\sigma$ s.t. $\partial_\sigma(m_1) = \partial_\sigma(m_2)$, *w.l.o.g.* assume that for any $m_1' <_\sigma m_1$ and $m_2' <_\sigma m_2$, $\partial_\sigma(m_1') = \partial_\sigma(m_2')$ implies $m_1' = m_2'$. By *courteous*, m_1 and m_2 have the same immediate causal dependencies, so $m_1 = m_2$ by *receptive*.

+-discrete fibration. Existence follows by repeated applications of *courtesy*, which entails that events in $x^\sigma \in \mathscr{C}(\sigma)$ displaying to maximal positive events in $\partial_\sigma(x^\sigma)$ must be maximal in x^σ. *Uniqueness* is clear by *locally injective*.

In the sequel, we shall only use this direction. So for the sake of economy, for the other direction we simply refer to [Castellan et al., 2017a, Lemma 3.13]. □

This was first noticed by Winskel in [Winskel, 2013b], in seeking a connection between concurrent strategies as profunctors – indeed, by the lemma above, it also follows that strategies are also characterized by the discrete fibration property for the composite partial order $\sqsubseteq_A = \supseteq^- \subseteq^+$, the so-called "Scott order", which informs a link with profunctors. In this monograph, we shall merely use the −-discrete opfibration property. Relying on that, we refine the mapification lemma in the following way[4]:

Lemma 6.3.4. *Consider $\sigma, \tau : A$ two strategies. Assume there is a function*

$$f : \mathscr{C}^+(\sigma) \to \mathscr{C}^+(\tau)$$

compatible with display maps and preserving unions. Then, there is a unique morphism of strategies $\widehat{f} : \sigma \Rightarrow \tau$ such that for all $x \in \mathscr{C}^+(\sigma)$, $\widehat{f}\, x = f\, x$.

[4]Thanks to Victor Blanchi for discovering this nice lemma!

Proof. Uniqueness. Consider g, h two such extensions, and consider $x \in \mathscr{C}(\boldsymbol{\sigma})$. Removing trailing events from x, there is a unique $y \in \mathscr{C}^+(\boldsymbol{\sigma})$ such that $y \subseteq^- x$. By hypothesis, $g(y) = h(y) = f(y)$. Since g, h are maps of event structures and preserve display maps, they must preserve inclusion and polarities of these inclusions, so that

$$f(y) \subseteq^- g(x), h(x)$$

but $\partial_{\boldsymbol{\tau}} g(x) = \partial_{\boldsymbol{\tau}} h(x)$ by pres. of display maps, so $g(x) = h(x)$ by $-$*-discrete opfibration* of Proposition 6.3.3. So $g(x) = h(x)$ for all $x \in \mathscr{C}(\boldsymbol{\sigma})$, so $g = h$ by Lemma 6.1.12.

Existence. We first extend f to all configurations: if $x \in \mathscr{C}(\boldsymbol{\sigma})$, as above we consider the unique $y \in \mathscr{C}^+(\boldsymbol{\sigma})$ such that $y \subseteq^- x$ obtained by removing trailing negative events in x; as f preserves display maps we have $\partial_{\boldsymbol{\tau}} f(y) = \partial_{\boldsymbol{\sigma}} y \subseteq^- \partial_{\boldsymbol{\sigma}} x$. By $-$*-discrete opfibration* of Proposition 6.3.3, there is a unique $z \in \mathscr{C}(\boldsymbol{\tau})$ such that $f(y) \subseteq^- z$ and $\partial_{\boldsymbol{\tau}}(z) = \partial_{\boldsymbol{\sigma}}(x)$; we set $f(x) = z$. So defined, $f : \mathscr{C}(\boldsymbol{\sigma}) \to \mathscr{C}(\boldsymbol{\tau})$ automatically preserves display maps and hence preserves cardinal (as $\partial_{\boldsymbol{\sigma}}$ and $\partial_{\boldsymbol{\tau}}$, as maps of event structures, are locally injective). By uniqueness in $-$*-discrete opfibration* of Proposition 6.3.3, it is easy to prove that f preserves unions. Hence by Lemma 6.1.11, it extends to a map of event structures in a unique way, concluding the proof.

Rigidity. We prove that any map of event structures $h : \boldsymbol{\sigma} \to \boldsymbol{\tau}$ such that $\partial_{\boldsymbol{\tau}} \circ h = \partial_{\boldsymbol{\sigma}}$ and that sends $+$-covered configurations to $+$-covered configurations, is rigid. We actually show that if $s_1 \rightarrowtriangle_{\boldsymbol{\sigma}} s_2$, then $h(s_1) <_{\boldsymbol{\sigma}} h(s_2)$, which entails rigidity. If $\mathrm{pol}(s_1) = +$ or $\mathrm{pol}(s_2) = -$, then $\partial_{\boldsymbol{\sigma}} s_1 \rightarrowtriangle_A \partial_{\boldsymbol{\sigma}} s_2$ by *courteous*, and thus $f(s_1) \leq_{\boldsymbol{\tau}} f(s_2)$ by Lemma 6.1.8 – so that $f(s_1) <_{\boldsymbol{\tau}} f(s_2)$ by local injectivity. It remains to consider the case where $\mathrm{pol}(s_1) = -$ and $\mathrm{pol}(s_2) = +$. In that case, as $[s_2]_{\boldsymbol{\sigma}} \in \mathscr{C}^+(\boldsymbol{\sigma})$, we have $h([s_2]_{\boldsymbol{\sigma}}) \in \mathscr{C}^+(\boldsymbol{\tau})$ as well. Now, observe that s_1 is maximal in $[s_2)_{\boldsymbol{\sigma}}$, otherwise contradicting $s_1 \rightarrowtriangle_{\boldsymbol{\sigma}} s_2$. Hence writing $x = [s_2]_{\boldsymbol{\sigma}} \setminus \{s_1, s_2\}$, $y = [s_2]_{\boldsymbol{\sigma}} \setminus \{s_2\}$, we have $x \prec\!\!\subset y \prec\!\!\subset [s_2]_{\boldsymbol{\sigma}}$ in $\mathscr{C}(\boldsymbol{\sigma})$. This is sent to $\boldsymbol{\tau}$ via h to the covering sequence in $\mathscr{C}(\boldsymbol{\tau})$

$$h(x) \overset{h(s_1)}{\prec\!\!\subset} h(y) \overset{h(s_2)}{\prec\!\!\subset} h([s_2]_{\boldsymbol{\sigma}}),$$

which shows in particular that only $h(s_2)$ can possibly depend on $h(s_1)$ within $h([s_2]_{\boldsymbol{\sigma}})$. Hence if $\neg(h(s_1) <_{\boldsymbol{\tau}} h(s_2))$, it immediately follows that $h(s_1)$ is maximal in $h([s_2]_{\boldsymbol{\sigma}})$. But $h(s_1)$ is negative, contradicting that $h([s_2]_{\boldsymbol{\sigma}})$ is $+$-covered; thus $h(s_1) <_{\boldsymbol{\sigma}} h(s_2)$. □

It is striking that the *rigidity* condition on morphisms is automatic, and follows from the fact that the map of event structures preserves $+$-covered configurations!

This lemma will be used often, in two particular situations: firstly, to show that two morphisms of strategies coincide, it suffices to show that they coincide on $+$-covered configurations; secondly, in order to prove $\boldsymbol{\sigma}$ and $\boldsymbol{\tau}$ isomorphic, it suffices to produce an order-iso between $\mathscr{C}^+(\boldsymbol{\sigma})$ and $\mathscr{C}^+(\boldsymbol{\tau})$ compatible with display maps (observing that the supremum of two $+$-covered configurations in $\mathscr{C}^+(\boldsymbol{\sigma})$, if it exists, is their union).

6.3.3 Composition and +-coveredness

Clearly $+$-covered configurations are important, in that morphisms between strategies are entirely determined by their action on $+$-covered configurations. Next we show that

additionally, +-covered configurations interact very well with composition. Let us fix for now $\sigma : A \vdash B$ and $\tau : B \vdash C$ two strategies, and investigate their composition.

The first step is to remark that +-coverdness makes *minimality* redundant:

Lemma 6.3.5. *Consider $x^\sigma \in \mathscr{C}^+(\sigma)$ and $x^\tau \in \mathscr{C}^+(\tau)$ causally compatible.*
Then, x^σ and x^τ are minimal *causally compatible.*

Proof. Seeking a contradiction, assume there is a maximal event of $\varphi[x^\sigma, x^\tau]$ in B, necessarily of the form $((1,s),(2,t))$ for $s \in x^\sigma$ and $t \in x^\tau$ necessarily both maximal. But necessarily one of them is negative, contradicting +-coveredness of x^σ and x^τ. □

Besides, the resulting configurations of the composition are automatically +-covered:

Lemma 6.3.6. *Consider $x^\sigma \in \mathscr{C}^+(\sigma)$ and $x^\tau \in \mathscr{C}^+(\tau)$ causally compatible.*
Then, $x^\tau \odot x^\sigma \in \mathscr{C}^+(\tau \odot \sigma)$.

Proof. First, x^σ, x^τ is minimal causally compatible by Lemma 6.3.5.

Consider $m \in x^\tau \odot x^\sigma$ maximal. This means that m is also maximal in $x^\tau \circledast x^\sigma = [x^\tau \odot x^\sigma]_{\tau\circledast\sigma}$, so corresponds to some pair maximal in $\varphi[x^\sigma, x^\tau]$ via Lemma 6.2.12. Since x^σ, x^τ is minimal causally compatible, this pair is in A or C – say *w.l.o.g.* in C, so it has the form $((2,c),(2,t))$ for $t \in x^\tau$. Necessarily, t is maximal in x^τ, so is positive since $x^\sigma \in \mathscr{C}^+(\tau)$. Hence, m is positive as well. □

So causally compatible +-covered $x^\sigma \in \mathscr{C}^+(\sigma)$ and $x^\tau \in \mathscr{C}^+(\tau)$ are minimal, and their composition yields $x^\tau \odot x^\sigma \in \mathscr{C}^+(\tau \odot \sigma)$ +-covered. We prove the converse:

Lemma 6.3.7. *Consider $x^\sigma \in \mathscr{C}(\sigma)$ and $x^\tau \in \mathscr{C}(\tau)$ minimal causally compatible.*
If $x^\tau \odot x^\sigma \in \mathscr{C}^+(\tau \odot \sigma)$ is +-covered, so are $x^\sigma \in \mathscr{C}^+(\sigma)$ and $x^\tau \in \mathscr{C}^+(\tau)$.

Proof. Consider $s \in x^\sigma$ maximal. Necessarily, there is $p \in x^\tau \circledast x^\sigma$ such that $p_\sigma = s$. If p is maximal in $x^\tau \circledast x^\sigma$, by Lemma 6.2.19 we have $x^\tau \circledast x^\sigma = [x^\tau \odot x^\sigma]_{\tau\circledast\sigma}$, so p must be visible and maximal in $x^\tau \odot x^\sigma$, hence positive by hypothesis, and s is positive.

Otherwise, there is some $p \rightarrowtriangle_{\tau\circledast\sigma} q$. By Lemma 6.2.14, $p_\sigma \rightarrowtriangle_\sigma q_\sigma$ or $p_\tau \rightarrowtriangle_\tau q_\tau$. The former contradicts maximality of $p_\sigma = s$, so $p_\tau \rightarrowtriangle_\tau q_\tau$. If s is positive, we are done. Otherwise, p_τ is positive, so q_τ is also in B by *courteous*, and $\partial_\tau(p_\tau) <_{B\vdash C} \partial_\tau(q_\tau)$. Then q_σ is also defined, and by Lemma 6.1.8, $p_\sigma < q_\sigma$, contradicting maximality of s. The symmetric reasoning shows any $t \in x^\tau$ maximal in x^τ is positive. □

We can now finally deduce our characterization of the composition of strategies:

Proposition 6.2.1. *Consider A, B, C games, and $\sigma : A \vdash B$ and $\tau : B \vdash C$ strategies.*
Then there is a strategy $\tau \odot \sigma : A \vdash C$, unique up to iso, s.t. there are order-isos:

$$(- \odot -) \;:\; \{(x^\tau, x^\sigma) \in \mathscr{C}^+(\tau) \times \mathscr{C}^+(\sigma) \mid \textit{causally compatible}\} \;\simeq\; \mathscr{C}^+(\tau \odot \sigma)$$

such that for $x^\sigma \in \mathscr{C}^+(\sigma)$ and $x^\tau \in \mathscr{C}^+(\tau)$ causally compatible,

$$\partial_{\tau\odot\sigma}(x^\tau \odot x^\sigma) = x^\sigma_A \vdash x^\tau_C$$

where we write $\partial_\sigma x^\sigma = x^\sigma_A \vdash x^\sigma_B$ and $\partial_\tau x^\tau = x^\tau_B \vdash x^\tau_C$.

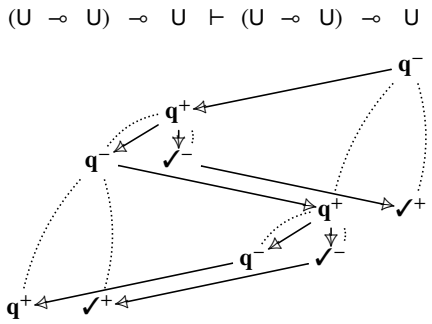


Figure 6.6: The copycat strategy on (U ⊸ U) ⊸ U

Proof. Existence. The composition exists by Proposition 6.2.22, and is a strategy by Proposition 6.3.1. The isos are restrictions of those of Proposition 6.2.22. By Lemma 6.3.5, causally compatible $x^{\sigma} \in \mathscr{C}^+(\sigma)$ and $x^{\tau} \in \mathscr{C}^+(\tau)$ are automatically minimal, and by Lemma 6.3.6, $x^{\tau} \odot x^{\sigma} \in \mathscr{C}^+(\tau \odot \sigma)$ is +-covered. Reciprocally, if $x^{\tau} \odot x^{\sigma} \in \mathscr{C}^+(\tau \odot \sigma)$ is +-covered, then by Lemma 6.3.7, so are $x^{\sigma} \in \mathscr{C}^+(\sigma)$ and $x^{\tau} \in \mathscr{C}^+(\tau)$.

Uniqueness. By Lemma 6.3.4. □

This concludes our study of composition of prestrategies and strategies. In the next section, we show that games and strategies may be organized into a bicategory **CG**.

6.4 The Bicategory CG

The bicategory will have: as *objects*, games, as *morphisms* from A to B, the strategies on $A \vdash B$, and as 2-*cells*, the morphisms of strategies. We have already seen above the composition of strategies and the *unitors*, 2-cells witnessing associativity of composition. It remains to construct *identities*, 2-cells witness identity laws, definition of horizontal composition of 2-cells; after which we must establish the coherence laws.

6.4.1 Copycat

First we define *copycat strategies*, which will provide identities.

Definition. Copycat, illustrated in Figure 6.6, is an *asynchronous forwarder*: it sends each negative event on $A \vdash A$ to the matching positive event on the other side.

We start with the components of copycat, postponing the verification of conditions.

Definition 6.4.1. *For any game A, we set $\mathbf{ɑ}_A : A \vdash A$ with components:*

$$\begin{array}{rrcl}
\text{events:} & |\mathbf{ɑ}_A| & = & |A| + |A| \\
\text{causality:} & \leq_{\mathbf{ɑ}_A} & = & (\{((1,a),(1,a')) \mid a \leq_A a'\}\uplus \\
 & & & \{((2,a),(2,a')) \mid a \leq_A a'\}\uplus \\
 & & & \{((1,a),(2,a)) \mid \mathrm{pol}_A(a) = +\}\uplus \\
 & & & \{((2,a),(1,a)) \mid \mathrm{pol}_A(a) = -\})^* \\
\text{conflict:} & \#_{\mathbf{ɑ}_A} & = & \{(e_1,e_2) \mid \exists e_1' \leq_{\mathbf{ɑ}_A} e_2, e_2' \leq_{\mathbf{ɑ}_A} e_2,\ e_1' \#_{A\vdash A} e_2'\}
\end{array}$$

where $(-)^*$ is the transitive closure; and $\partial_{\mathbf{ɑ}_A} : |\mathbf{ɑ}_A| \to |A \vdash A|$ the identity function.

Copycat is often a challenging landmark in setting up a game semantics, and here it is no exception: it is not obvious at all that it satisfies all the required conditions! We shall slowly examine various aspects of this definition, for a fixed game A.

The event structure $\mathbf{ɑ}_A$. We first focus on the basic structure of $\mathbf{ɑ}_A$.

Lemma 6.4.2. *Consider A a game. Then, $(|\mathbf{ɑ}_A|, \leq_A, \#_A)$ is an event structure. Moreover, $\mathscr{C}(\mathbf{ɑ}_A) = \{x \parallel y \mid x, y \in \mathscr{C}(A),\ x \cap y \subseteq^+ x \ \&\ x \cap y \subseteq^- y\}$.*

Proof. Immediate verifications, omitted. □

It will be useful to understand exactly what are the immediate causal links of $\mathbf{ɑ}_A$:

Lemma 6.4.3. *Consider A a game, and $(i,a),(j,a') \in \mathbf{ɑ}_A$. Then $(i,a) \rightarrowtriangle_{\mathbf{ɑ}_A} (j,a')$ iff*

(1) *$i = j$, $a \rightarrowtriangle_A a'$, and ($\mathrm{pol}_{\mathbf{ɑ}_A}(i,a) = +$ or $\mathrm{pol}_{\mathbf{ɑ}_A}(j,a') = -$), or*
(2) *$i \neq j$, $a = a'$, $\mathrm{pol}_{\mathbf{ɑ}_A}(i,a) = -$ and $\mathrm{pol}_{\mathbf{ɑ}_A}(j,a') = +$.*

This is a direct verification. We also include a direct consequence of this:

Lemma 6.4.4. *Consider A a game. Then, $\mathscr{C}^+(\mathbf{ɑ}_A) = \{x \parallel x \mid x \in \mathscr{C}(A)\}$.*

Proof. $\supseteq$. Clear by Lemma 6.4.2.

$\subseteq$. Consider $x \parallel y \in \mathscr{C}^+(\mathbf{ɑ}_A)$; by Lemma 6.4.2, we have $x, y \in \mathscr{C}(A)$ with $x \cap y \subseteq^+ x$ and $x \cap y \subseteq^- y$. Suppose one of these inclusions is strict; for instance with $a \in y$ such that $a \notin x$ (the other case is symmetric). Necessarily, $\mathrm{pol}_A(a) = -$. If a is maximal in y, then $(2,a)$ is maximal in $x \parallel y$ by Lemma 6.4.3 since $a \notin x$. But $\mathrm{pol}_{A\vdash A}(2,a) = -$, contradicting that $x \parallel y$ is $+$-covered. Otherwise, there is $a' \in y$ such that $a \rightarrowtriangle_A a'$. Now if $\mathrm{pol}_A(a') = +$, we must have $a' \in x$, so $a \in x$ as $x \in \mathscr{C}(A)$, contradiction. So, $\mathrm{pol}_A(a') = -$. By iterating this reasoning, we find $a'' \in y \setminus x$ negative and maximal for A; and by the reasoning above $(2,a'')$ is negative maximal in $x \parallel y$, contradiction. □

This lemma above will play an important role in this monograph: it shows that when restricting to $+$-covered configurations (which is always possible when comparing strategies, by Lemma 6.3.4), copycat behaves as the identity relation. For $x \in \mathscr{C}(A)$, we write $\mathbf{ɑ}_x = x \vdash x \in \mathscr{C}^+(\mathbf{ɑ}_A)$ for the corresponding $+$-covered configuration.

Next, we show that copycat indeed is a well-formed strategy.

Proposition 6.4.5. *For any game A, we have $\mathbb{c}_A : A \vdash A$ a strategy.*

Proof. By Lemma 6.4.2, $\mathbb{c}_A$ is an event structure. It is direct that $\partial_{\mathbb{c}_A}$ is a map of event structures. *Courteous* follows from Lemma 6.4.3, and *receptive* from Lemma 6.4.2. □

In the sequel, we shall mostly reason on copycat via the characterization of its +-covered configurations in Lemma 6.4.4, showing that it acts as an identity relation.

6.4.2 Copycat and Strategies

Now, we show that copycat indeed is neutral for composition with respect to strategies.

The unitors. To define the unitor isomorphisms, from our existing tools, the only thing left to prove is that synchronizations with copycat never induce any deadlock. For this – and for later on in this monograph – we introduce the following lemma:

Lemma 6.4.6. *Take $\sigma : A \vdash B, \tau : B \vdash C$, and $x^\sigma \in \mathscr{C}(\sigma), x^\tau \in \mathscr{C}(\tau)$ matching.*
Consider the relation $\lhd_\sigma \cup \lhd_\tau$ on $x_A \parallel x_B \parallel x_C$ defined with

$$\begin{array}{llllll} (\partial_\sigma \parallel C)(m) & \lhd_\sigma & (\partial_\sigma \parallel C)(m') & \text{where} & m <_{\sigma\parallel C} m' & (m, m' \in x^\sigma \parallel x_C), \\ (A \parallel \partial_\tau)(m) & \lhd_\tau & (A \parallel \partial_\tau)(m') & \text{where} & m <_{A\parallel\tau} m' & (m, m' \in x_A \parallel x^\tau) \end{array}$$

writing $\partial_\sigma(x^\sigma) = x_A \parallel x_B$ and $\partial_\tau(x^\tau) = x_B \parallel x_C$.
Then, x^σ, x^τ are causally compatible iff $\lhd_\sigma \cup \lhd_\tau$ has no cycle within x_B.

Proof. If. First, notice that $\lhd$ has no direct link between x_A and x_C. The crux of the argument is that if $m \lhd m'$ with one of them in A, then $m \lhd_\sigma m'$: indeed if they are both in A, then by definition either $m \lhd_\sigma m'$ and we are done, or $m <_{A\parallel B\parallel C} m'$ and $m \lhd_\sigma m'$ follows from Lemma 6.1.8. Likewise, if $m \lhd m'$ with one of them in C, then $m \lhd_\tau m'$.

We proceed with the proof. First, it is a simple reformulation of the definition that x^σ, x^τ are causally compatible iff $\lhd = \lhd_\sigma \cup \lhd_\tau$ is acyclic. If there is a cycle in $\lhd$, it must pass through B: otherwise it still entirely in A or C. But by the observation above, this yields a cycle in $\lhd_\sigma$ or $\lhd_\tau$, contradiction. So the cycle passes through B. Then, for

$$m^B \lhd n_1^A \lhd \cdots \lhd n_k^A \lhd (m')^B ,$$

again by the observation above all those links are in $\lhd_\sigma$, hence $m^B \lhd_\sigma (m')^B$. Likewise, all sections of the cycle in C may be removed symmetrically, yielding a cycle in x_B. □

We now focus on the interactions of a strategy with copycat. In particular, the core of the next lemma is that no deadlocks can arise from the interaction of a strategy with copycat, so that all matching pairs of +-covered configurations are causally compatible.

Lemma 6.4.7. *Consider $\sigma : A \vdash B$ a strategy, and $x^\sigma \in \mathscr{C}^+(\sigma)$, $x^{\mathbb{c}_B} \in \mathscr{C}^+(\mathbb{c}_B)$.*
Then, $x^\sigma, x^{\mathbb{c}_B}$ are causally compatible iff $x^{\mathbb{c}_B} = \mathbb{c}_{x_B^\sigma}$, for $\partial_\sigma(x^\sigma) = x_A^\sigma \parallel x_B^\sigma$.

Proof. If. Write $\partial_\sigma(x^\sigma) = x_A \parallel x_B$, and seeking a contradiction, consider a cycle in $x_A \parallel x_B^\ell \parallel x_B^r$ – using ℓ, r for disambiguation – in the relation $\lhd_\sigma \cup \lhd_{\mathfrak{C}_B}$ defined as:

$$\begin{array}{llllll} (\partial_\sigma \parallel B)(m) & \lhd_\sigma & (\partial_\sigma \parallel B)(m') & \text{where} & m <_{\sigma \parallel B} m' & (m, m' \in x^\sigma \parallel x_B^r) \\ (A \parallel \partial_{\mathfrak{C}_B})(m) & \lhd_{\mathfrak{C}_B} & (A \parallel \partial_{\mathfrak{C}_B})(m') & \text{where} & m <_{A \parallel \mathfrak{C}_B} m' & (m, m' \in x_A \parallel x_B^\ell \parallel x_B^r), \end{array}$$

by Lemma 6.4.6 we can assume that the cycle is entirely in x_B^ℓ. Now, it is straightforward from the definition of $\mathfrak{C}_B$ that if $m \lhd_{\mathfrak{C}_B} m'$ with m, m' in x_B^ℓ, then $m <_{A \parallel B \parallel B} m'$ so that $m \lhd_\sigma m'$ by Lemma 6.1.8. Thus, this yields a cycle in $\lhd_\sigma$, contradiction.

Only if. Immediate from the definition and Lemma 6.4.4. □

Clearly, the symmetric lemma holds, with a symmetric proof:

Lemma 6.4.8. *Consider $\sigma : A \vdash B$ a strategy, and $x^{\mathfrak{C}_A} \in \mathscr{C}^+(\mathfrak{C}_A)$, $x^\sigma \in \mathscr{C}^+(\sigma)$.*
Then, $x^{\mathfrak{C}_A}, x^\sigma$ are causally compatible iff $x^{\mathfrak{C}_A} = \mathfrak{C}_{x_A^\sigma}$, for $\partial_\sigma(x^\sigma) = x_A^\sigma \parallel x_B^\sigma$.

This completes the ingredients needed to define the *unitors*:

Proposition 6.4.9. *Consider $\sigma : A \vdash B$ a strategy.*
*Then there are unique isomorphisms of strategies, the **unitors**:*

$$l_\sigma : \mathfrak{C}_B \odot \sigma \cong \sigma \qquad\qquad r_\sigma : \sigma \odot \mathfrak{C}_A \cong \sigma$$

such that for all $x^\sigma \in \mathscr{C}^+(\sigma)$, we have $l_\sigma(\mathfrak{C}_{x_B^\sigma} \odot x^\sigma) = x^\sigma$ and $r_\sigma(x^\sigma \odot \mathfrak{C}_{x_A^\sigma}) = x^\sigma$.

Proof. Configurations and symmetries of $\mathfrak{C}_B \odot \sigma$ and $\sigma \odot \mathfrak{C}_A$ have this shape by Lemmas 6.4.7 and 6.4.8. These assignments on configurations are order-isos and compatible with display maps, thus extend to an isomorphism of strategies by Lemma 6.3.4. □

6.4.3 Horizontal Composition and Bicategorical Structure

The missing piece is the horizontal composition of 2-cells, handled by:

Proposition 6.4.10. *Consider $\sigma, \sigma' : A \vdash B$ and $\tau, \tau' : B \vdash C$ strategies, and $f : \sigma \Rightarrow \sigma'$, $g : \tau \Rightarrow \tau'$ morphisms of strategies.*
*There is a unique morphism of strategies, the **horizontal composition** of f and g,*

$$g \odot f : \tau \odot \sigma \Rightarrow \tau' \odot \sigma',$$

such that $(g \odot f)(x^\tau \odot x^\sigma) = g(x^\tau) \odot f(x^\sigma)$ for all $x^\tau \odot x^\sigma \in \mathscr{C}^+(\tau \odot \sigma)$.

Proof. Uniqueness. Obvious from Lemma 6.3.4.

Existence. Consider $x^\tau \odot x^\sigma \in \mathscr{C}^+(\tau \odot \sigma)$. As f and g preserve display maps, $f(x^\sigma)$ and $g(x^\tau)$ are matching. As f and g are rigid, and by Lemma 6.1.8, they induce order-isos $f : x^\sigma \cong f(x^\sigma)$ and $g : x^\tau \cong g(x^\tau)$, so $f(x^\sigma)$ and $g(x^\tau)$ are still $+$-covered, and causally compatible. So the action on $+$-covered configurations $x^\tau \odot x^\sigma \in \mathscr{C}^+(\tau \odot \sigma)$ given above is well-defined. It obviously preserves display maps and preserves unions by Lemma 6.2.20, hence it lifts to a morphism of strategies by Lemma 6.2.20. □

We are now in position to conclude with the bicategorical structure:

Theorem 6.4.11. *There is* **CG***, a bicategory with:* objects*, all games,* morphisms *from A to B, strategies on $A \vdash B$, and* 2-cells*, morphisms of strategies.*

Proof. Identities are copycat strategies, composition is composition of strategies. Associators are defined in Section 6.2.3, unitors in Section 6.4.2. The laws are obtained by checking them pointwise on $+$-covered configurations and concluding by Lemma 6.3.4; for instance, naturality of the left unitor is established very simply, via:

$$\begin{aligned} l_{\sigma} \circ (\mathrm{id} \odot f)(\mathbb{c}_{x_B^{\sigma}} \odot x^{\sigma}) &= l_{\sigma}(\mathbb{c}_{x_B^{\sigma}} \odot f(x^{\sigma})) \\ &= f(x^{\sigma}) \\ &= f \circ l_{\sigma}(\mathbb{c}_{x_B^{\sigma}} \odot x^{\sigma}) \end{aligned}$$

for $f : \sigma \Rightarrow \sigma' : A \vdash B$ a morphism of strategies, and $x^{\sigma} \in \mathscr{C}^{+}(\sigma)$. □

There is, of course, much more to the structure of **CG**. In particular it is a compact closed bicategory, and may be used as a model for a linear programming language. But we shall not investigate this structure further, and will do it instead for **TCG**.

6.4.4 Characterization of Strategies

Before adding symmetry into the picture, we prove an additional property on copycat. Namely, it allows a neat characterization of *strategies* among prestrategies:

Proposition 6.4.12. *Consider $\sigma : A \vdash B$ a prestrategy.*
Then, σ is a strategy iff $\mathbb{c}_B \odot \sigma \odot \mathbb{c}_A \cong \sigma$.

Proof. If. It suffices to prove that $\mathbb{c}_B \odot \sigma \odot \mathbb{c}_A$ is automatically *receptive* and *courteous*.

For *courteous*, consider $p \rightarrowtriangle_{\mathbb{c}_B \odot \sigma \odot \mathbb{c}_A} p'$ such that $\mathrm{pol}(p) = +$ or $\mathrm{pol}(p') = -$, say first it is the former. Necessarily p occurs in A or B, say *w.l.o.g.* that it is in B. Up to iso we consider $p \rightarrowtriangle_{\mathbb{c}_B \odot \sigma'} p'$, for $\sigma' = \sigma \odot \mathbb{c}_A$. By Lemma 6.2.21 we have

$$p \rightarrowtriangle_{\mathbb{c}_B \circledast \sigma'} q_1 \rightarrowtriangle_{\mathbb{c}_B \circledast \sigma'} \cdots \rightarrowtriangle_{\mathbb{c}_B \circledast \sigma'} q_n \rightarrowtriangle_{\mathbb{c}_B \circledast \sigma'} p'$$

with $q_1, \ldots, q_n$ synchronized. By Lemma 6.2.14, we must have $p_{\mathbb{c}_B} \rightarrowtriangle_{\mathbb{c}_B} (q_1)_{\mathbb{c}_B}$ – as p_{σ} does not exist. But as p is positive, this implies $\partial_{\mathbb{c}_B}(p_{\mathbb{c}_B}) \rightarrowtriangle_{B \vdash B} \partial_{\mathbb{c}_B}((q_1)_{\mathbb{c}_B})$, contradicting q_1 synchronized, so $p \rightarrowtriangle_{\mathbb{c}_B \circledast \sigma'} p'$ and $p_{\mathbb{c}_B} \rightarrowtriangle_{\mathbb{c}_B} p'_{\mathbb{c}_B}$, which entails $\partial_{\mathbb{c}_B}(p_{\mathbb{c}_B}) \rightarrowtriangle_{B \vdash B} \partial_{\mathbb{c}_B}(p'_{\mathbb{c}_B})$ so that $\partial_{\mathbb{c}_B \odot \sigma'}(p) \rightarrowtriangle_{A \vdash B} \partial_{\mathbb{c}_B \odot \sigma'}(p')$ as required. All the other cases are symmetric, so that $\partial_{\mathbb{c}_B \odot \sigma \odot \mathbb{c}_A}(p) \rightarrowtriangle_{A \vdash B} \partial_{\mathbb{c}_B \odot \sigma \mathbb{c}_A}(p')$ as required.

For *receptive*, consider $z \in \mathscr{C}(\mathbb{c}_B \odot \sigma \odot \mathbb{c}_A)$ with $\partial_{\mathbb{c}_B \odot \sigma \odot \mathbb{c}_A}(z) \vdash_{A \vdash B} b^{-}$, *w.l.o.g.* assume b occurs in B, and consider again z, which by Proposition 6.2.22 we may write as $z = (x_B \parallel y_B) \odot x^{\sigma'} \in \mathscr{C}(\mathbb{c}_B \odot \sigma')$ with $\sigma' = \sigma \odot \mathbb{c}_A$, with $\partial_{\mathbb{c}_B \odot \sigma'}(z) = x_A^{\sigma'} \parallel y_B$. So by hypothesis, we have $y_B \vdash_B b^{-}$. Then it is immediate that $x_B \parallel (y_B \cup \{b\})$ are still minimal causally compatible, so that by Proposition 6.2.22 we have

$$(x_B \parallel y_B) \odot x^{\sigma'} \quad \overset{p}{\subset\!\!\!-} \quad (x_B \parallel (y_B \cup \{b\})) \odot x^{\sigma'}$$

for some $(x_B \parallel y_B) \odot x^{\sigma'} \vdash_{\mathbb{C}_B \odot \sigma'} p$ such that $\partial_{\mathbb{C}_B \odot \sigma'}(p) = b$. Uniqueness is straightforward via Proposition 6.2.22 and definition of $\partial_{\mathbb{C}_B \odot \sigma'}$.

Receptive and *courteous* are clearly invariant under strong iso, so $\boldsymbol{\sigma}$ is a strategy.

Only if. By bicategorical laws for **CG**. □

Thus strategies are exactly those prestrategies invariant under composition with copycat, *i.e.* robust when accessed under asynchronous delay. Conceptually, this vindicates the definition of strategies. Technically, it occasionally provides a convenient way to prove that certain prestrategies are actually strategies. This result appeared as the main theorem in Rideau and Winskel's original paper on concurrent strategies [Rideau and Winskel, 2011]. A result in the same spirit already appeared in Ghica and Murawski's interleaving concurrent games model [Ghica and Murawski, 2008].

6.5 Conclusions and Historical Notes

Concurrent games were initiated by Abramsky and Melliès' seminal fully complete model for MALL [Abramsky and Melliès, 1999]. There, strategies are certain stable closure operators on dI-domains: intuitively, a strategy-as-closure-operator closes the current position under all available Player moves, played in some unspecified order. Concurrent games were originally presented as a way to circumvent the so-called *Blass problem*, *i.e.* the non-associativity of composition in Blass' model of linear logic, diagnosed in [Abramsky, 2003] as due to an excess of sequentiality.

Asynchronous Games. Following this, Melliès led a research programme on *asynchronous games* [Melliès, 2003, Melliès, 2004a, Melliès, 2004b, Melliès, 2005]. Melliès' goal was to reproduce and extend the success of concurrent games in a language closer to more standard (sequential) game semantics; as well as put the spotlight on the *causal structures* that lurk behind the traditional notion of *innocence*. Asynchronous games are based on Mazurkiewicz traces: plays are *alternating* but related by *homotopy tiles* expressing causal independence between parts of computation. Plays up to homotopy give rise to a notion of *positions* compatible with the positions of the original concurrent games. Based on alternating asynchronous games, Melliès was able to provide a fully complete model of full propositional linear logic [Melliès, 2005], showing that what made this possible already for closure operators was not in fact concurrency, but *positionality* – see [Clairambault, 2023] for a recent retelling of this story.

Later, Melliès and Mimram developed basic *non-alternating* asynchronous games [Melliès and Mimram, 2007]. While non-alternating asynchronous strategies are nondeterministic as sets of plays, they satisfy conditions ensuring that they are essentially deterministic up to the choice of the scheduler – in particular, it follows a posteriori from the conditions that the set of possible executions is the set of linearizations of a *causality* partial order: provided the external Opponent plays in the same way, all execution paths will ultimately reach the same state.

Strategies as Partial Orders. Independently, a notion of *strategies-as-partial-orders* emerged from Girard's Ludics [Girard, 2001]. Indeed, *designs* (strategies in Ludics) may be seen as a linear version of Hyland-Ong strategies [Faggian and Hyland, 2002], presented following their *causal* structure rather than the traditional chronologically ordered plays. This inspired *Ludics Nets*, or *L-nets* [Faggian and Maurel, 2005]: a notion of *strategies-as-graph*, aiming at a concurrent generalization of designs (see also [Curien and Faggian, 2005] for links with more standard strategies via proof nets). Inspired by this line of work and ideas from Hyland, Faggian and Piccolo introduced a category of *strategies-as-partial-orders* [Faggian and Piccolo, 2009], which lands close to Mimram and Melliès' non-alternating asychronous strategies – but whereas in Mimram and Melliès' work the causal partial order is derived, here it is primitive.

Strategies as Event Structures. Rideau and Winskel then introduced the bicategory **CG** [Rideau and Winskel, 2011] that we developed in this chapter: as for Faggian and Piccolo, strategies in **CG** are explicitly partially ordered, but they additionally have a fibred structure letting them harmonously handle non-determinism with explicit branching information – this is analogous to other fibred approaches to non-determinism in game semantics [Eberhart et al., 2017, Jacq and Melliès, 2018]. At the same time, Winskel was starting a research project "ECSYM", seeking a concurrent generalization of domain theory – the main inspirations were the non-deterministic extension of domain theory given by spans of event structures [Saunders-Evans and Winskel, 2006] and *event structures with symmetry* [Winskel, 2007]. But ECSYM soon focused on games and strategies and was the cradle for the first steps of the present developments.

From now on, unless specified otherwise and until the end of this document, we shall use the term *concurrent games* to refer specifically to this latter framework **CG** – keeping in mind all its precursors and the intellectual history behind it.

Chapter 7

Thin Concurrent Games

The main limitation of **CG** is that it is inherently restricted to speaking about systems with linear resource usage. Indeed, games as in Definition 6.1.3 do not support the *bang* construction from linear logic. Of course, one can define in **CG** a construction on games taking countably many copies as illustrated in (6.3), but this operation is incomplete: analogously to categorifications of the relational model (generalized species of structure [Fiore et al., 2008] or template games [Melliès, 2019b]) where *states* form a groupoid, we need to add additional structure to games, expressing which events are "morally the same" by specifying the admissible permutations between events.

Intuitively, the roadmap for building *concurrent games with symmetry* is to follow the same recipe as for **CG**, but replacing *event structures* (and their maps) with Winskel's *event structures with symmetry* [Winskel, 2007] – which add to event structures precisely the ability to express these missing permutations. But while developing this, it turned out far more subtle than initially anticipated. With respect to this anticipated roadmap, the main surprise of **TCG** is a rediscovery of phenomenon first noticed by Melliès in the setting of asynchronous games: that a *causal*, *deterministic* account of uniformity required to split symmetries into two, the *positive* symmetries due to Player, and the *negative* symmetries due to Opponent [Melliès, 2003].

In this chapter, we will review a few of the subtleties and arising design choices. Then we shall focus on constructing *Thin Concurrent Games*, under the form of a bicategory **TCG**. We must hence ask the reader to brace for another rather technical journey. Like **CG**, the construction of **TCG** is not easy. But like **CG**, we did our best to engineer its construction in such a way that one can use it later on while leaving most combinatorial details and low-level definitions nicely encapsulated.

7.1 Symmetry in Games and Strategies

In this section, we introduce the basic definitions for games and strategies in the presence of symmetry, leaving for later the compositional structure. Before that, we start

with a few preliminaries on event structures with symmetry.

7.1.1 Event Structures with Symmetry

In order to motivate the definition of event structures with symmetry, we start with the key construction that *introduces* symmetry in games arising from the interpretation of types or linear logic formulas: the *bang* – making formal the illustration of (6.3).

Definition 7.1.1. *Consider A a game. Then, we define the* ***bang*** $!A$ *with:*

$$\begin{array}{rrcl} \text{events:} & |!A| & = & \mathbb{N} \times |A| \\ \text{causality:} & (i,a_1) \leq_{!A} (j,a_2) & \Leftrightarrow & i = j \wedge a_1 \leq_A a_2 \\ \text{conflict:} & (i,a_1) \#_{!A} (j,a_2) & \Leftrightarrow & i = j \wedge a_1 \#_A a_2 \\ \text{polarities:} & \mathrm{pol}_{!A}(i,a) & = & \mathrm{pol}_A(a)\,. \end{array}$$

Following the methodology of linear logic, the construction $!A$ injects a non-linear behaviour into an otherwise linear setting – one may define the usual, non-linear arrow via $A \to B = !A \multimap B$. To achieve that, the definition above follows the intuition from AJM games (see Section 3.4) that $!A$ consists in countably many independent copies of the game A, kept apart via integers referred to as *copy indices*.

It will be useful to adopt specific notations for configurations of $!A$. First:

Definition 7.1.2. *Consider A a game. We write* $\mathrm{Fam}(A)$ *for the set of all families $(x_i)_{i\in I}$ where $I \subseteq_f \mathbb{N}$ is a finite set of integers, and $x_i \in \mathscr{C}(A)$ with $x_i \neq \emptyset$ for all $i \in I$.*
We call elements of $\mathrm{Fam}(A)$ *simply* ***families*** *on A.*

The set $\mathrm{Fam}(A)$ is partially ordered by $(x_i)_{i\in I} \leq (y_j)_{j\in J}$ iff $I \subseteq J$ and $x_i \subseteq y_i$ for all $i \in I$. Families on A are useful in that they correspond to configurations of $!A$ – the proof of this proposition is straightforward, but observe that it crucially relies on the hypothesis that the configurations appearing in families are *non-empty*.

Proposition 7.1.3. *Consider A a game. Then, the function*

$$\begin{array}{rcl} \mathrm{Fam}(A) & \to & \mathscr{C}(!A) \\ (x_i)_{i\in I} & \mapsto & \sum_{i\in I} x_i = \biguplus_{i\in I} \{i\} \times x_i \end{array}$$

is an order-isomorphism – we often write $[x_i \mid i \in I] \in \mathscr{C}(!A)$ for $(x_i)_{i\in I} \in \mathrm{Fam}(A)$.

Copy indices are special with respect to other constructions producing new moves. They keep distinct copies apart, but in principle the behaviour of strategies should not depend on the specific integers used: their exact value should not matter, which we express with our convention introduced in Section 3.4 to often write them down in grey. But we need a formal way to express that they are indeed interchangeable:

Definition 7.1.4. *Consider A a game, and $x, y \in \mathscr{C}(!A)$. A* ***plain reindexing*** $\theta : x \cong_A y$ *is a bijection $\theta : x \simeq y$ such that there is $\pi : I \simeq J$ a permutation satisfying*

$$\theta(i,a) = (\pi(i), a)$$

for all $(i,a) \in !A$.

A plain reindexing links two configurations $x, y \in \mathscr{C}(!A)$ which only differ via copy indices for the bang. We need more than an equivalence relation: to express that strategies are "invariant under reindexings", we need an explicit bijection rather than the mere information that two configurations can be reindexed to each other. Indeed, while in AJM games an equivalence $s \cong_A t$ induces, chronologically, a 1-to-1 matching between the events of s and those of t, an equivalence relation does not suffice anymore for configurations, which are only partially ordered – hence explicit bijections are required.

Plain reindexings only change copy indices coming from the external !, but in general reindexings may affect copy indices coming from deep within the type. Thus a type should not be interpreted only with a game, but with a "game with a set of allowed reindexings" – but to what mathematical structure does that correspond?

Isomorphism families. It is an *isomorphism family* on an event structure:

Definition 7.1.5. *An **isomorphism family** on event structure E is a set $\mathcal{S}(E)$ of bijections between configurations of E, satisfying the additional conditions:*

groupoid:	*$\mathcal{S}(E)$ contains identities; is closed under composition and inverse.*
restriction:	*for all $\theta : x \simeq y \in \mathcal{S}(E)$ and $x \supseteq x' \in \mathscr{C}(E)$,*
	there is a (necessarily) unique $\theta \supseteq \theta' \in \mathcal{S}(E)$ s.t. $\theta' : x' \simeq y'$.
extension:	*for all $\theta : x \simeq y \in \mathcal{S}(E)$, $x \subseteq x' \in \mathscr{C}(E)$,*
	there is a (not necessarily unique) $\theta \subseteq \theta' \in \mathcal{S}(E)$ s.t. $\theta' : x' \simeq y'$.

*The pair $(E, \mathcal{S}(E))$ is called an **event structure with symmetry (ess)**.*

We regard isomorphism families as *proof-relevant* equivalence relations: they convey which configurations are interchangeable, witnessed by an explicit bijection.

If E is an ess, we call the elements of $\mathcal{S}(E)$ **symmetries**. We write $\theta : x \cong_E y$ to mean that $\theta : x \simeq y$ is a bijection such that $\theta \in \mathcal{S}(E)$, and write $x = \mathsf{dom}(\theta)$ and $y = \mathsf{cod}(\theta)$. We also write $x \cong_E y$ to mean that there is a symmetry θ such that $\theta : x \cong_E y$, this yields an equivalence relation on configurations.

It is an easy exercise to prove that symmetries are automatically order-isomorphisms:

Lemma 7.1.6. *Consider E an ess, and $\theta : x \cong_E y$ a symmetry.*
Then, for all $e_1, e_2 \in x$, we have $e_1 \leq_E e_2$ iff $\theta(e_1) \leq_E \theta(e_2)$.

Proof. Left as an exercise to the reader. □

Maps with symmetry. If event structures are equipped with an isomorphism family, we must adjust the notion of maps of event structures so that symmetry is preserved:

Definition 7.1.7. *Consider E and F two ess, and $f : E \to F$ a map of event structures.*
*We say that f **preserves symmetry** if for all $\theta \in \mathcal{S}(E)$, we have $f\theta \in \mathcal{S}(F)$, where*

$$f\theta = \{(fe, fe') \mid (e, e') \in \theta\}\,.$$

*Then f is a **map of event structures with symmetry** – we write* **ESS** *for the category.*

It is immediate that $f\theta$ may also be expressed as the composition

$$fx \overset{f^{-1}}{\simeq} x \overset{\theta}{\simeq} y \overset{f}{\simeq} fy$$

obtained by exploiting local injectivity of f; the above then asks that this composition of bijections should be in $\mathcal{S}(F)$ – this observation will be used silently from now on.

In *concurrent games with symmetry*, both games and strategies will be certain event structures with symmetry, and the *display map* of a strategy will have to preserve symmetry. As for plain maps of event structures, maps of event structures with symmetry may be characterized through their action on configurations *and symmetries*:

Lemma 7.1.8. *Consider E and F two ess, and $f : E \to F$ a map of event structures. Then f preserves symmetry iff there is a (necessarily unique) function*

$$\widetilde{f} : \mathcal{S}(E) \to \mathcal{S}(F)$$

commuting with dom *and* cod, i.e. $\mathsf{dom} \circ \widetilde{f} = f \circ \mathsf{dom}$ *and* $\mathsf{cod} \circ \widetilde{f} = f \circ \mathsf{cod}$.

Proof. Consider $\theta : x \cong_E y$. By Lemma 7.1.6, θ is an order-iso, so we get

$$\emptyset = \theta_0 \overset{(e_1,e_1')}{\lessdot} \theta_1 \overset{(e_2,e_2')}{\lessdot} \theta_2 \lessdot \ldots \lessdot \theta_{n-1} \overset{(e_n,e_n')}{\lessdot} \theta_n = \theta \tag{7.1}$$

by following any covering chain of x or y. By commutation with dom and cod, it is immediate by induction on i that for all $0 \leq i \leq n$, $\widetilde{f}\,\theta_i = f\,\theta_i$; in particular $\widetilde{f}\,\theta = f\,\theta$. This shows uniqueness of $\widetilde{f}$ as it must be the action of f on symmetries as specified in Definition 7.1.7. But this means that f sends symmetries to symmetries, thus it preserves symmetry as required. □

Altogether, we get a symmetry-aware version of the mapification lemma:

Lemma 7.1.9. *Consider E, F two ess, and a function $f : \mathscr{C}(E) \to \mathscr{C}(F)$.*
If f preserves unions, cardinality, and there is a function $\widetilde{f} : \mathcal{S}(E) \to \mathcal{S}(F)$ s.t.

$$\mathsf{dom} \circ \widetilde{f} = f \circ \mathsf{dom}\,, \qquad \mathsf{cod} \circ \widetilde{f} = f \circ \mathsf{cod}\,,$$

then there exists a unique map of ess $\widehat{f} : E \to F$ s.t. for all $x \in \mathscr{C}(E)$, $\widehat{f}(x) = f(x)$.

Proof. By Lemma 6.1.11, there is $\widehat{f} : E \to F$ a unique map of event structures. By Lemma 7.1.8 it preserves symmetry, which concludes the proof. □

Observe that by Lemma 7.1.8 the map $\widetilde{f}$, if it exists, is necessarily unique as it must coincide with the action of $\widehat{f}$ (the map of event structures) on symmetries as specified by Definition 7.1.7. In the sequel, when applying Lemma 7.1.9, we shall sometimes accordingly omit the definition of $\widetilde{f}$, when we feel its explicit description is not helpful.

Maps up to Symmetry. For now, symmetry on event structures puts a new obligation on maps, which must now preserve it. We shall now see that it also provides us with a new equivalence relation on maps, which may coincide *up to symmetry* only:

Definition 7.1.10. *Consider E, F ess, and $f, g : E \to F$ maps of ess.*
*We say that f and g are **symmetric**, written $f \sim g$, if the composite bijection*

$$f\,x \quad \overset{f^{-1}}{\simeq} \quad x \quad \overset{g}{\simeq} \quad g\,x$$

written $\langle f, g \rangle_x$ and obtained via local injectivity *of f and g, is in $\mathcal{S}(F)$.*

Symmetry between maps will be extremely important in the sequel: it allows us to identify maps that are the same only up to reindexing, which will be essential in the construction of models of linear logic. For symmetry to play that role, it must of course be preserved by all constructions on maps. In particular:

Proposition 7.1.11. *The category* **ESS***, equipped with symmetry on homsets, is enriched over equivalence relations. In particular, composition preserves $\sim$.*

Proof. Straightforward, using that maps preserve symmetry. □

7.1.2 Saturated Games with Symmetry

There are several ways to extend concurrent games in the presence of symmetry. Though this will not be our final word, we first sketch an approach we call "saturated" or "fat" – mathematically, it is easier, but suffers from significant drawbacks.

Games with symmetry. The motto leading to saturated concurrent games is to simply follow the same recipe as in Section 6.1.2, but building on **ESS** rather than **ES**. Accordingly, we first simply add symmetry to Definition 6.1.3:

Definition 7.1.12. *A **game with symmetry (gws)** is an ess $A = (|A|, \leq_A, \#_A, \mathcal{S}(A))$ with*

$$\mathrm{pol}_A : |A| \to \{-, +\}$$

*a **polarity function** such that for all $\theta : x \cong_A y$ and $a \in x$, $\mathrm{pol}_A(a) = \mathrm{pol}_A(\theta(a))$.*

We are not far removed from traditional, sequential notions of games with symmetry from the literature. In particular, the above is a natural concurrent generalization of AJM games – the link can be easily made by first recovering alternating plays:

Definition 7.1.13. *An **alternating play** on gws A is a sequence $s = s_1 \dots s_n$ which is:*

valid*:*	$\forall 1 \leq i \leq n,\ \{s_1, \dots, s_i\} \in \mathscr{C}(A)$,
non-repetitive*:*	$\forall 1 \leq i, j \leq n,\ s_i = s_j \Rightarrow i = j$,
alternating*:*	$\forall 1 \leq i \leq n-1,\ \mathrm{pol}_A(s_i) \neq \mathrm{pol}_A(s_{i+1})$,
negative*:*	*if* $n \geq 1$, *then* $\mathrm{pol}_A(s_1) = -$.

We write $\downarrow\uparrow$-$\mathsf{Plays}(A)$ for the set of alternating plays on A.

It is clear that if A is a gws, then the tuple $(|A|, \mathrm{pol}_A, \Downarrow\uparrow\text{-}\mathsf{Plays}(A))$ forms a simple game in the sense of Definition 3.1.1. To obtain an AJM game, it remains to define:

Definition 7.1.14. *Let A be a gws and $s, t \in \Downarrow\uparrow\text{-}\mathsf{Plays}(A)$.*

*We say s and t are **symmetric**, written $s \cong_A t$, if s and t have the same length, and*

$$\theta^j_{s,t} = \{(s_i, t_i) \mid 1 \leq i \leq j\} : \{s_1, \ldots, s_j\} \cong_A \{t_1, \ldots, t_j\}$$

is a symmetry in $\mathcal{S}(A)$ for all $1 \leq j \leq n$; writing $s = s_1 \ldots s_n$ and $t = t_1 \ldots t_n$.

As expected, we then have the AJM "unfolding" of a game with symmetry:

Proposition 7.1.15. *For any gws A, $(|A|, \mathrm{pol}_A, \Downarrow\uparrow\text{-}\mathsf{Plays}(A), \cong_A)$ is an AJM game.*

Proof. That $\cong_A$ is an equivalence relation follows directly from the *groupoid* condition on event structures with symmetry. There are three more conditions to check.

For *(1)*, consider $s_1 \ldots s_n \cong_A t_1 \ldots t_p$. By definition, $n = p$ and

$$\theta^n_{s,t} = \{(s_i, t_i) \mid 1 \leq i \leq n\} : \{s_1, \ldots, s_n\} \cong_A \{t_1, \ldots, t_n\}$$

is a symmetry in $\mathcal{S}(A)$. So for all $1 \leq i \leq n$, $\mathrm{pol}_A(s_i) = \mathrm{pol}_A(\theta^n_{s,t}(s_i)) = \mathrm{pol}_A(t_i)$.

Likewise, *(2)* is immediate from the definition.

For *(3)*, consider $s = s_1 \ldots s_n \cong t_1 \ldots t_n = t$ with $sa \in \Downarrow\uparrow\text{-}\mathsf{Plays}(A)$. By definition,

$$\theta^n_{s,t} = \{(s_i, t_i) \mid 1 \leq i \leq n\} : \{s_1, \ldots, s_n\} \cong_A \{t_1, \ldots, t_n\}$$

is a symmetry in $\mathcal{S}(A)$. By hypothesis, $x = \{s_1, \ldots, s_n\} \in \mathcal{C}(A)$ extends to $x' = x \uplus \{a\} \in \mathcal{C}(A)$. Therefore, by condition *extension*, writing $y = \{t_1, \ldots, t_n\}$, there is some $y \subseteq y' \in \mathcal{C}(A)$ and an extension $\theta^n_{s,t} \subseteq \theta' \in \mathcal{S}(A)$ such that $\theta' : x' \cong_A y'$. But θ' is a bijection, so we must have $y' = y \uplus \{\theta'(a)\}$, which we write $b = \theta'(a)$. It is then immediate that $tb \in \Downarrow\uparrow\text{-}\mathsf{Plays}(A)$ with $sa \cong_A tb$ as required. □

For the expert reader, we mention that this unfolding does not respect the usual AJM game constructions as an alternating play on a parallel composition $A \parallel B$ (though we have not yet defined this in the presence of symmetry) might not be locally alternating on A and B – this is analogous to the situation in Hyland-Ong games.

Saturated strategies. Following the same motto as before, a *concurrent prestrategy with symmetry* on gws A should simply add symmetry to Definition 6.1.5, *i.e.* should consist in a tuple $\boldsymbol{\sigma} = (|\boldsymbol{\sigma}|, \leq_{\boldsymbol{\sigma}}, \#_{\boldsymbol{\sigma}}, \mathcal{S}(\boldsymbol{\sigma}), \partial_{\boldsymbol{\sigma}})$ where $(|\boldsymbol{\sigma}|, \leq_{\boldsymbol{\sigma}}, \#_{\boldsymbol{\sigma}}, \mathcal{S}(\boldsymbol{\sigma}))$ is an ess, and

$$\partial_{\boldsymbol{\sigma}} : \boldsymbol{\sigma} \to A$$

preserves symmetry. But while a good first step, this is not yet a working notion of (pre)strategy with symmetry: for instance, it accepts a prestrategy as in Figure 7.1, playing on $!\mathsf{N}$, and which simply outputs the copy index used by Opponent to interrogate the natural number[1]. This breaks *uniformity*: as in AJM games (see Section 3.4), in

[1]Though the $\mathbf{q}^-_i$ are to be symmetric in the game, nothing forces them to be symmetric in $\boldsymbol{\sigma}$.

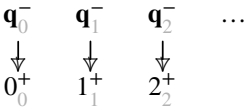


Figure 7.1: An invalid prestrategy with symmetry on !N

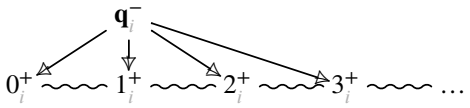
order to satisfy the expected laws of a model of linear logic we must ensure that the behaviour of strategies should not depend on Opponent's choice of copy indices.

In *saturated concurrent games*, uniformity is achieved by a enforcing *isofibration*:

Definition 7.1.16. *Consider A a gws. A **saturated prestrategy** on A is a tuple $\sigma = (|\sigma|, \leq_\sigma, \#_\sigma, \partial_\sigma)$ where $(|\sigma|, \leq_\sigma, \#_\sigma)$ is an ess, $\partial_\sigma : \sigma \to A$ is a map of ess, and s.t.:*

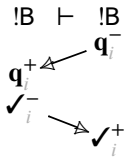
isofibration: *for all $x^\sigma \in \mathscr{C}(\sigma)$, for all $\theta : y_A \cong_A \partial_\sigma x^\sigma$, there is a unique $\varphi : y^\sigma \cong_\sigma x^\sigma$ such that $\partial_\sigma \varphi = \theta$.*

This removes the behaviour of Figure 7.1 – or rather, it forces there to also be



a countable non-deterministic choice letting the strategy pick a random natural number in reaction to any copy index (although the diagram only displays conflicts between contiguous numbers, it is meant that all natural numbers are pairwise conflicting). In effect, *isofibration* enforces uniformity by ensuring that the behaviour of a strategy is invariant under the addition of *noise*, non-deterministically scrambling all copy indices.

The impact of saturation. *Isofibration* is a rather neat condition mathematically speaking, but it is computationally more debatable. To see the issue, consider



a typical configuration of the *copycat strategy* on !U, anticipating on some definitions to come. As in AJM games, in symmetry-free concurrent games, copycat is a simple deterministic strategy merely copying Opponent moves from one side to the other.

But, it does not satisfy *isofibration*: indeed, that would force copycat to also include configurations featuring reindexing, as in Figure 7.2 but with all choices of copy indices. This "saturated copycat" would be a non-deterministic strategy, as illustrated by Figure

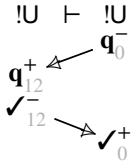


Figure 7.2: Reindexing

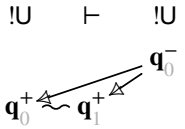


Figure 7.3: Local conflict

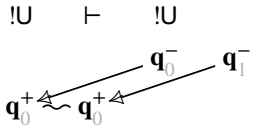


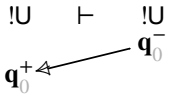
Figure 7.4: Non-local conflict

7.3, as a copy index can be propagated to any copy index on the other side but must be propagated only once. The emerging patterns of conflict can be quite complex: for instance, Figure 7.4 illustrates that by *local injectivity*, two threads independently coming up with the same index must confict, even though they should be computationally independent if the strategies are to satisfy something like single-threadedness.

Saturated concurrent games ensure uniformity by adding non-deterministic noise. All the non-deterministic choices arising in saturated copycat yield symmetric moves, so it is in principle possible to express that the saturated copycat is deterministic "up to symmetry", but it is quite technical and the model loses part of the concrete appeal of **CG**. We now explore an alternative: *thin concurrent games*.

7.1.3 Thin Concurrent Games

We wish to construct an alternative setting for strategies with symmetry, that avoids *isofibration* and instead recognizes the old – deterministic – copycat as uniform. In what sense is the deterministic copycat uniform? Well, as in AJM games: in the diagram



if Opponent reindexes their move to $\mathbf{q}^-_1$, there is a *unique* matching Player reindexing of $\mathbf{q}^+_0$ to $\mathbf{q}^+_1$ so that we get a configuration symmetric to the original. It is this unique Player reindexing that must be captured for *thin* games. The solution is dual to saturation: in saturated games, *all* reindexings are allowed, giving rise to a non-deterministic choice. In contrast in thin games, only *one canonical* reindexing is allowed; conflicts are actual, computationally meaningful non-deterministic choices and not artifacts from symmetry.

But what is an "Opponent reindexing", and a "Player reindexing"? Our first step will be to refine games with symmetry to include chosen symmetries that capture this.

Thin Concurrent Games. We start with the definition:

Definition 7.1.17. *A **thin concurrent game (tcg)** is a gws* $A = (|A|, \leq_A, \#_A, \mathrm{pol}_A, \mathcal{S}(A))$

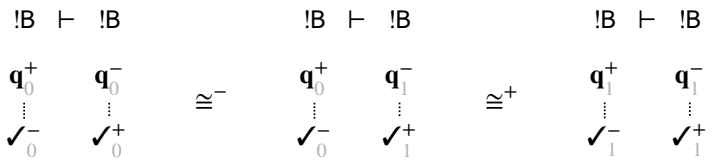


Figure 7.5: Positive and negative symmetries

with isomorphism families $\mathcal{S}_+(A), \mathcal{S}_-(A)$ *s.t.* $\mathcal{S}_+(A) \subseteq \mathcal{S}(A)$, $\mathcal{S}_-(A) \subseteq \mathcal{S}(A)$, *s.t.*

orthogonality: *for all* $\theta \in \mathcal{S}(A)$, *if* $\theta \in \mathcal{S}_+(A) \cap \mathcal{S}_-(A)$, *then* $\theta = \mathrm{id}_x$ *for* $x \in \mathcal{C}(A)$,
$-$-receptivity: *if* $\theta \in \mathcal{S}_-(A)$ *and* $\theta \subseteq^- \theta' \in \mathcal{S}(A)$, *then* $\theta' \in \mathcal{S}_-(A)$,
$+$-receptivity: *if* $\theta \in \mathcal{S}_+(A)$ *and* $\theta \subseteq^+ \theta' \in \mathcal{S}(A)$, *then* $\theta' \in \mathcal{S}_+(A)$,

where $\theta \subseteq^p \theta'$ *means that* $\theta \subseteq \theta'$ *adding only (pairs of) events of polarity* p.

Symmetries in $\mathcal{S}_+(A)$ are called **positive symmetries**, and intuitively correspond to reindexing only positive events. If $\theta : x \cong_A y$ is positive, we also write $\theta : x \cong^+_A y$. Likewise, symmetries in $\mathcal{S}_-(A)$ are the **negative symmetries**, corresponding to reindexing only negative events, and we write $\theta : x \cong^-_A y$ if θ is negative.

Orthogonality means that if a symmetry reindexes *only* Player moves and *only* Opponent moves, it must be an identity symmetry. A negative symmetry that extends in $\mathcal{S}(A)$ with negative events must remain negative, and symmetrically.

We show in Figure 7.5 examples of positive and negative symmetries. It looks like the first should not be negative, as we rename $\checkmark^+_0$ to $\checkmark^+_1$. However, this index is only a repetition of 0 in $\mathbf{q}^-_0$: it originates in the ! on the rhs of $\vdash$, which is attached to $\mathbf{q}^-$.

This split of symmetries between *positive* and *negative* symmetries is the core of thin concurrent games. This simple definition has the following pleasant consequence:

Lemma 7.1.18. *Consider* A *a tcg, and* $\theta : x \cong_A y$ *a symmetry.*
Then there are unique $z \in \mathcal{C}(A)$, $\theta^- : x \cong^-_A z$ *and* $\theta^+ : z \cong^+_A y$ *s.t.* $\theta = \theta^+ \circ \theta^-$.

Proof. Existence. By induction on the size (*i.e.* cardinality) of θ. If $\theta = \emptyset$, it is clear. Take $\theta : x \cong_A y$ factoring as $\theta = \theta^+ \circ \theta^-$ for $\theta^- : x \cong^-_A z$, $\theta^+ : z \cong^+_A y$, and assume

$$\theta \uplus \{(a, b)\} : x \uplus \{a\} \cong_A y \uplus \{b\}\,.$$

Since symmetries preserve polarity, $\mathrm{pol}_A(a) = \mathrm{pol}_A(b)$ – say *w.l.o.g.* it is positive, the other case is symmetric. Then, by *extension* on $\mathcal{S}_-(A)$, there is some extension

$$\theta^- \uplus \{(a, c)\} : x \uplus \{a\} \cong^-_A z \uplus \{c\}$$

for some c a negative event. But then, by *groupoid* laws for $\mathcal{S}(A)$ we have

$$(\theta \uplus \{(a, b)\}) \circ (\theta^- \uplus \{(a, c)\})^{-1} : z \uplus \{c\} \cong_A y \uplus \{b\}$$

which extends θ^+ with $\{(c,b)\}$, so by $+$-*receptivity*, $\theta^+ \uplus \{(c,b)\} \in \mathcal{S}_+(A)$ as required.

Uniqueness. Now, assume there is an alternative factorization $\theta = \varphi^+ \circ \varphi^-$ for

$$\varphi^- : x \cong_A^- z' \qquad\qquad \varphi^+ : z' \cong_A^+ y,$$

then we have $\varphi^- \circ (\theta^-)^{-1} = (\varphi^+)^{-1} \circ \theta^+ : z \cong_A z$; but the former is negative and the latter positive, so those compositions must be the identity by *orthogonality*. □

This is illustrated in Figure 7.5. Intuitively, any reindexing may be obtained by first reindexing Opponent moves, then reindexing Player moves, in a unique way. Obviously, by groupoid laws we can deduce the dual positive-negative factorization.

Thin concurrent strategies. Now, we define *thin concurrent strategies* on a tcg A.

As before, we give the definition in two stages. First, we define *prestrategies*:

Definition 7.1.19. *A **prestrategy** on tcg A comprises an ess $(|\sigma|, \leq_\sigma, \#_\sigma, \mathcal{S}(\sigma))$ with*

$$\partial : \sigma \to A$$

*a map of event structures with symmetry called the **display map**, subject to:*

$\sim$-receptive: *for $\theta : x \cong_\sigma y$, and extensions $x \vdash_\sigma s_1^-$, $\partial(\theta) \vdash_{\mathcal{S}(A)} (\partial(s_1^-), a_2^-)$, there is a unique $s_2^- \in \sigma$ s.t. $\theta \vdash_{\mathcal{S}(\sigma)} (s_1^-, s_2^-)$ and $\partial(s_2^-) = a_2^-$,*

thin: *for $\theta \in \mathcal{S}(\sigma)$, if $\partial_\sigma \theta \in \mathcal{S}_+(A)$, then $\theta = \mathrm{id}_x$ for some $x \in \mathcal{C}(\sigma)$.*

where for $\psi \in \mathcal{S}(A)$, $\psi \vdash_{\mathcal{S}(A)} (a_1, a_2)$ iff $(a_1, a_2) \notin \psi$ and $\psi \uplus \{(a_1, a_2)\} \in \mathcal{S}(A)$.

We call this a *prestrategy*, keeping the same terminology as without symmetry – this should not cause confusion: if the tcg is merely a game (meaning, its symmetries are all identities), these additional conditions vacuously hold. We may still sometimes say *thin (pre)strategy*, to insist on the presence of these additional conditions.

As without symmetry, *prestrategies* are not yet adequate as strategies, but they do capture *uniformity* with respect to symmetry. In particular, $\sim$-*receptivity* forces σ to internally consider as symmetric any negative moves symmetric in the game, *e.g.*

$$\{\mathbf{q}_0^-\} \cong_{!\mathsf{B}}^- \{\mathbf{q}_1^-\} \qquad \Rightarrow \qquad \{\mathbf{q}_0^-\} \cong_\sigma \{\mathbf{q}_1^-\}$$

for $\sigma : !\mathsf{B}$ (provided σ has unique events matching $\mathbf{q}_0^-, \mathbf{q}_1^-$ referred to here by the same name). Then, the *extension* axiom of isomorphism families on $\mathcal{S}(\sigma)$ ensure there is

$$\begin{array}{ccc} \{\mathbf{q}_0^-\} & \cong_\sigma & \{\mathbf{q}_1^-\} \\ \Cap_+ & & \\ x & & \end{array} \qquad \Rightarrow \qquad \begin{array}{ccc} \{\mathbf{q}_0^-\} & \cong_\sigma & \{\mathbf{q}_1^-\} \\ \Cap_+ & & \Cap_+ \\ x & \cong_\sigma & y \end{array}$$

then *thin* ensures that this extension is unique; another such extension y' yields

$$\theta : y \cong_\sigma y'$$

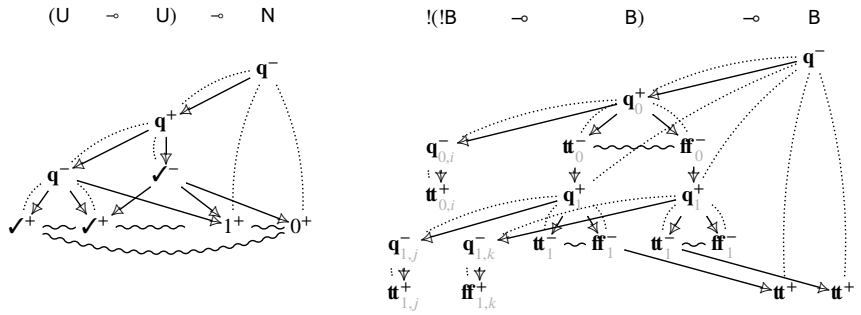


Figure 7.6: Two thin concurrent strategies

obtained by going back and forth via x. But then, displaying to the game yields

$$\mathrm{id}_{\{\mathbf{q}^-_1\}} \quad \subseteq^+ \quad \partial_{\boldsymbol{\sigma}}\theta$$

but $\mathrm{id}_{\{\mathbf{q}^-_1\}} \in \mathcal{S}_+(!\mathsf{B})$ by groupoid laws, so that $\partial_{\boldsymbol{\sigma}}\,\theta \in \mathcal{S}_+(!\mathsf{B})$ by $+$-*receptivity* of positive symmetries. But then by *thin*, θ must be an identity bijection. Together, those conditions $\sim$-*receptivity*, *extension*, *thin* are a powerful way to ensure uniformity of $\boldsymbol{\sigma}$.

Strategies are obtained by adding the same conditions as in Definition 6.1.6:

Definition 7.1.20. *A prestrategy $\boldsymbol{\sigma}$ on tcg A is a **strategy** if it satisfies:*

courteous: *for all $s_1 \rightarrowtriangle_{\boldsymbol{\sigma}} s_2$, if $\mathrm{pol}(s_1) = +$ or $\mathrm{pol}(s_2) = -$ then $\partial_{\boldsymbol{\sigma}}(s_1) \rightarrowtriangle_A \partial_{\boldsymbol{\sigma}}(s_2)$,*
receptive: *for all $x \in \mathscr{C}(\boldsymbol{\sigma})$, for all $\partial_{\boldsymbol{\sigma}}(x) \vdash_A a^-$,*
there is a unique $x \vdash_{\boldsymbol{\sigma}} s^-$ such that $\partial_{\boldsymbol{\sigma}}(s) = a$,

We write $\boldsymbol{\sigma} : A$ to mean that $\boldsymbol{\sigma}$ is a strategy on game A.

As an illustration, we show in Figure 7.6 two examples of concurrent strategies. On the left hand side, the strategy is displayed exhaustively – its symmetries are all reduced to identities. On the right hand side, the representation is symbolic: Opponent may ask arbitrarily many times the moves $\mathbf{q}^-_{0,i}$, $\mathbf{q}^-_{1,j}$ and $\mathbf{q}^-_{1,k}$. Symmetries are not represented in the diagram, but would be all bijections between configurations preserving and reflecting $\rightarrowtriangle$, and only changing copy indices i, j and k. This strategy is that for the term

$$\begin{aligned}\lambda f^{\mathbb{B}\to\mathbb{B}}.\, &\mathbf{if}\ (f\ \mathbf{tt})\\ &(\mathbf{if}\ (f\ \mathbf{tt})\ \bot\ \mathbf{tt})\\ &(\mathbf{if}\ (f\ \mathbf{ff})\ \mathbf{tt}\ \bot)\end{aligned}$$

with respect to the interpretation given in Section 9.1. The left hand side strategy is the affine version of one definable with state, as shall be seen in Section 9.1.

7.1.4 Constructions on Thin Concurrent Games

We start with the basic tcgs from which all others will be defined. The **empty tcg**, written $\mathbf{1}$, has no events. The tcgs matching *ground types* are U, B and N respectively

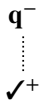


Figure 7.7: U

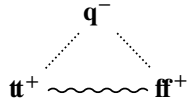


Figure 7.8: B

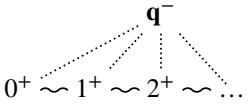


Figure 7.9: N

displayed in Figures 7.7, 7.8 and 7.9 – in all three cases, all symmetries are identities.

Basic constructions. First, we introduce the *simple parallel composition*. In this constructions, tcgs A_1 and A_2 are simply put side by side with all components inherited:

Definition 7.1.21. *Consider two tcgs A_1 and A_2.*
*Then, we define their **simple parallel composition** $A_1 \parallel A_2$ as:*

$$\begin{array}{rrcl}
\text{events:} & |A_1 \parallel A_2| & = & |A_1| + |A_2| \\
\text{causality:} & (i,a) \leq_{A_1 \parallel A_2} (j,a') & \Leftrightarrow & i = j \;\&\; a \leq_{A_i} a' \\
\text{conflict:} & (i,a) \#_{A_1 \parallel A_2} (j,a') & \Leftrightarrow & i = j \;\&\; a \#_{A_i} a' \,, \\
\text{symmetry:} & \theta \in \mathcal{S}(A_1 \parallel A_2) & \Leftrightarrow & \exists \theta_i \in \mathcal{S}(A_i), \theta = \theta_1 \parallel \theta_2 \,, \\
\text{polarities:} & \mathrm{pol}_{A_1 \parallel A_2}(i,a) & = & \mathrm{pol}_{A_i}(a) \\
\text{pos. symmetries:} & \theta_1 \parallel \theta_2 \in \mathcal{S}_+(A_1 \parallel A_2) & \Leftrightarrow & \theta_1 \in \mathcal{S}_+(A_1) \;\&\; \theta_2 \in \mathcal{S}_+(A_2) \\
\text{neg. symmetries:} & \theta_1 \parallel \theta_2 \in \mathcal{S}_-(A_1 \parallel A_2) & \Leftrightarrow & \theta_1 \in \mathcal{S}_-(A_1) \;\&\; \theta_2 \in \mathcal{S}_-(A_2) \,.
\end{array}$$

where, if $\theta_i : x_i \cong_{A_i} y_i$ for $i \in \{1,2\}$, we set $(\theta_1 \parallel \theta_2)(i,e) = (i, \theta_i(e))$.

It is direct by construction that all conditions of tcgs are satisfied. Note that only taking the first four components into account, this definition also applies to plain ess.

Recall that Section 6.1.3 that configurations $x \in \mathscr{C}(A \parallel B)$ are exactly those of the form $x = x_A \parallel x_B = x_A + x_B$ where $x_A \in \mathscr{C}(A)$ and $x_B \in \mathscr{C}(B)$. We now have additional analogous decompositions for *symmetries*:

Lemma 7.1.22. *Consider A and B ess. Then, there are order-isomorphisms*

$$\begin{array}{rcrcl}
(- \parallel -) & : & \mathscr{C}(A) \times \mathscr{C}(B) & \cong & \mathscr{C}(A \parallel B) \\
(- \parallel -) & : & \mathcal{S}(A) \times \mathcal{S}(B) & \cong & \mathcal{S}(A \parallel B) \\
(- \parallel -) & : & \mathcal{S}_-(A) \times \mathcal{S}_-(B) & \cong & \mathcal{S}_-(A \parallel B) \\
(- \parallel -) & : & \mathcal{S}_+(A) \times \mathcal{S}_+(B) & \cong & \mathcal{S}_+(A \parallel B)
\end{array}$$

which commute with dom *and* cod*, in the sense that for all $\theta_A \in \mathcal{S}(A)$ and $\theta_B \in \mathcal{S}(B)$,*

$$\begin{array}{rcl}
\mathsf{dom}(\theta_A \parallel \theta_B) & = & \mathsf{dom}(\theta_A) \parallel \mathsf{dom}(\theta_B) \,, \\
\mathsf{cod}(\theta_A \parallel \theta_B) & = & \mathsf{cod}(\theta_A) \parallel \mathsf{cod}(\theta_B) \,.
\end{array}$$

Proof. Straightforward with $\theta_A \parallel \theta_B$ comprising all $((1,a),(1,a'))$ for $(a,a') \in \theta_A$ and $((2,b),(2,b'))$ for $(b,b') \in \theta_B$; and all sets ordered by componentwise inclusion. □

From now on, we shall routinely decompose symmetries as above. Next, we introduce the other elementary operation on tcgs, the *dual*:

Definition 7.1.23. *Consider A a tcg. Its **dual** $A^\perp$ has all components as for A except:*

$$\begin{array}{rcll} \mathrm{pol}_{A^\perp}(a) & = & -\mathrm{pol}_A(a), & (a \in A) \\ \mathcal{S}_+(A^\perp) & = & \mathcal{S}_-(A), & \\ \mathcal{S}_-(A^\perp) & = & \mathcal{S}_+(A). & \end{array}$$

It is clear by construction that if A is a tcg, then so is $A^\perp$. As for plain games, if A and B are tcgs then we set $A \vdash B$ to be $A^\perp \parallel B$. Clearly, the order-isomorphisms of Lemma 7.1.22 adapt smoothly to this construction, and let us *e.g.* write any symmetry on $A \vdash B$ as $\theta_A \vdash \theta_B : x_A \vdash x_B \cong_{A \vdash B} y_A \vdash y_B$ for $\theta_A : x_A \cong_A y_A$ and $\theta_B : x_B \cong_B y_B$.

Constructions for PCF types. In this chapter, we shall not yet detail the interpretation of programming languages. Nevertheless, we introduce here the two additional constructions required to interpret PCF types, so that as to make precise the games in the examples appearing so far. For the remaining constructions on tcgs, see Chapter 8.

The first construction is the *linear arrow* applied to tcgs with a structural constraint:

Definition 7.1.24. *We define the following conditions on a thin concurrent game A:*

$$\begin{array}{rl} \text{negative:} & \text{for all } a \in \min(A), \text{ we have } \mathrm{pol}_A(a) = -, \\ \text{well-opened:} & A \text{ is negative, and there is exactly one } a \in \min(A). \end{array}$$

where $\min(A)$ *is the set of minimal events in* A.

As it turns out, the interpretation of PCF types only yields well-opened tcgs, whose unique minimal events is the *initial* move by Opponent, prompting computation.

To interpret the arrow type, we first define the *linear arrow*. Intuitively, in game semantics, strategies *from A to B* are defined to play on A and B simultaneously, with however the polarity on A reversed to reflect its contravariance. This suggests using $A \vdash B$ as arrow type – and we shall see indeed that strategies from A to B do play on this compound game. But as in the traditional games introduced in Part I, types should yield *negative* games (at least as long as we are considering call-by-name languages); while $A \vdash B$ is not negative unless A is empty. So we set:

Definition 7.1.25. *Consider A a negative tcg, and B a well-opened tcg.*
*Their **linear arrow** $A \multimap B$ has all components set as $A \vdash B$ except for:*

$$\text{causality:} \quad \leq_{A \multimap B} \;=\; \leq_{A \vdash B} \uplus \{((2, b_0), (1, a)) \mid a \in A\}$$

writing $\min(B) = \{b_0\}$*, yielding a well-opened tcg.*

This corrects the non-negativity of $A \vdash B$, by forcing the missing dependency.

Finally, as for AJM games, the last missing component of the arrow tcg is the *exponential* operation from linear logic; which finally has a non-trivial action on symmetry:

Definition 7.1.26. *Consider A a negative tcg. Then, we define the* ***bang*** $!A$ *with:*

$$
\begin{array}{rrcl}
\text{events:} & |!A| & = & \mathbb{N} \times |A| \\
\text{causality:} & (i, a_1) \leq_{!A} (j, a_2) & \Leftrightarrow & i = j \wedge a_1 \leq_A a_2 \\
\text{conflict:} & (i, a_1) \#_{!A} (j, a_2) & \Leftrightarrow & i = j \wedge a_1 \#_A a_2 \\
\text{symmetries:} & \theta \in \mathcal{S}(!A) & \Leftrightarrow & \exists \pi : \mathbb{N} \simeq \mathbb{N}, \quad \exists (\theta_n)_{n \in \mathbb{N}} \in \mathcal{S}(A)^{\mathbb{N}} \\
 & & & \forall (i, a) \in \mathsf{dom}(\theta), \theta(i, a) = (\pi(i), \theta_i(a)) \\
\text{polarities:} & \mathrm{pol}_{!A}(i, a) & = & \mathrm{pol}_A(a) \\
\text{pos. symmetries:} & \theta \in \mathcal{S}_+(!A) & \Leftrightarrow & \exists (\theta_n)_{n \in \mathbb{N}} \in \mathcal{S}_+(A)^{\mathbb{N}}, \\
 & & & \forall (i, a) \in \mathsf{dom}(\theta), \theta(i, a) = (i, \theta_i(a)) \\
\text{neg. symmetries:} & \theta \in \mathcal{S}_-(!A) & \Leftrightarrow & \exists \pi : \mathbb{N} \simeq \mathbb{N}, \exists (\theta_n)_{n \in \mathbb{N}} \in \mathcal{S}_-(A)^{\mathbb{N}}, \\
 & & & \forall (i, a) \in \mathsf{dom}(\theta), \theta(i, a) = (\pi(i), \theta_i(a))
\end{array}
$$

This yields a negative tcg $!A$*.*

Positive and negative symmetries for the bang were illustrated in Figure 7.5. We skip the routine verification that this defines a tcg, but it is worth observing that this depends on *negative*. If $\ominus\oplus$ is a game with two independent events with the indicated polarities, then $!(\ominus\oplus)$ is *not* a tcg: the positive symmetry $\oplus_0 \cong^+ \oplus_0$ extends with $(\oplus_1, \oplus_2)$ to a valid symmetry, so by $+$*-receptivity* it should be a positive symmetry – but it is not[2].

Here, we keep our convention to write copy indices in grey. With only the first four clauses, this also applies to define the bang $!E$ of a plain ess E – notice then the similarity between the definition of symmetries and Definition 3.4.3. But for a tcg, the definitions of positive and negative symmetries are asymmetric: as the bang applies to negative tcgs, the permutation π must be understood as reindexing Opponent moves – accordingly, it is arbitrary for $\mathcal{S}_-(!A)$ but restricted to the identity for $\mathcal{S}_+(!A)$.

Altogether, this lets us interpret any PCF type as a well-opened tcg, with $[\![\mathbb{U}]\!] = \mathsf{U}$, $[\![\mathbb{B}]\!] = \mathsf{B}$, $[\![\mathbb{N}]\!] = \mathsf{N}$, and $[\![A \to B]\!] = ![\![A]\!] \multimap [\![B]\!]$ – we stop there for now.

7.2 Mediating Between Strategies

In Chapter 6, we have seen that (plain) concurrent games and strategies naturally organized themselves into a *bicategory*, where strategies are related by certain *morphisms* (Definition 6.1.7) – the corresponding *isomorphisms* provide an adequate equivalence on strategies. As in **CG**, strategies on thin concurrent games are to be compared via adequate notions of morphisms. But unlike in **CG**, there are quite a few candidates for those morphisms and generated equivalences; hence we start by exploring the options.

7.2.1 A Zoology of Morphisms and Equivalences

Morphisms between strategies. It is natural to first relate strategies with the same morphisms, importing Definition 6.1.7, simply replacing morphisms of event structures

[2]The bang in thin concurrent games behaves well only in situations where one deals with *polarized* games, *i.e.* all games are either positive or negative: no games have minimal events of mixed polarity. In the rare situations where one needs a bang on non-polarized games [Baillot et al., 1997a, Melliès, 2019b], saturated concurrent games should be used instead – in that case the reader is referred to [Castellan et al., 2014].

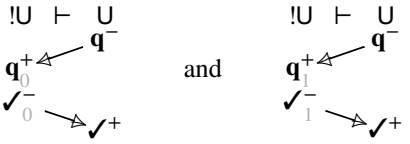
with morphisms of event structures with symmetry – we call those *strong morphisms*.

Definition 7.2.1. *Consider σ and τ two (pre)strategies on a tcg A.*
*A **strong morphism** from σ to τ, is a map of ess $f : \sigma \to \tau$ satisfying:*

$$\begin{aligned} \text{compatibility with display maps:} \quad & \partial_\tau \circ f = \partial_\sigma, \\ \text{rigidity:} \quad & \textit{for all } s \leq_\sigma t, \textit{ we have } fs \leq_\tau ft. \end{aligned}$$

The associated **strong isomorphism** between (pre)strategies, is written $f : \sigma \cong \tau$. The induced equivalence relation is a useful equivalence between strategies, understood as σ and τ behaving in exactly the same way – including the choice of copy indices – up to renaming of internal events. With this definition, one can replay Chapter 6 based on tcgs and obtain a bicategory of tcgs, strategies, and strong morphisms.

However, strong morphisms miss the point of symmetry, which is to identify strategies that only differ in their choice of copy indices. For instance, we wish to identify



which are two candidates for the *dereliction* strategy on U. The dereliction must choose a copy index, but it should not matter at all which one is selected – being able to perform such identifications is vital in order to get the expected laws for the exponential (!).

To address that, we weaken the commutation property, using Definition 7.1.10:

Definition 7.2.2. *Consider $\sigma, \tau : A$ two (pre)strategies.*
*A **weak morphism** from σ to τ is a map of ess $f : \sigma \to \tau$ such that*

$$\begin{aligned} \text{weak commutation:} \quad & \partial_\tau \circ f \sim \partial_\sigma, \\ \text{rigidity:} \quad & \textit{for all } s \leq_\sigma t, \textit{ we have } fs \leq_\tau ft. \end{aligned}$$

As announced, this authorizes reindexings, of both Player and Opponent moves. But intuitively, we wish to relate strategies by *only letting Player* change their indices. In tcgs, this might be achieved by appealing to the two *positive* and *negative* sub-symmetries. This is done by first considering a refinement of Definition 7.1.10:

Definition 7.2.3. *Consider E an ess, A a tcg, and $f, g : E \to A$ maps of ess. Then f, g are **positively symmetric**, written $f \sim^+ g$, iff $\forall x \in \mathscr{C}(E)$, $\langle f, g \rangle_x \in \mathscr{S}_+(A)$.*

Intuitively, two parallel maps that target a tcg are *positively symmetric* if they only differ by a reindexing of Player moves. This induces a final notion of morphism:

Definition 7.2.4. *Consider $\sigma, \tau : A$ two (pre)strategies.*
*A **positive morphism** from σ to τ, written $f : \sigma \Rightarrow \tau$, is a map $f : \sigma \to \tau$ s.t.:*

$$\begin{aligned} \text{positive commutation:} \quad & \partial_\tau \circ f \sim^+ \partial_\sigma, \\ \text{rigidity:} \quad & \textit{for all } s \leq_\sigma t, \textit{ we have } fs \leq_\tau ft. \end{aligned}$$

Altogether we have *three* notions of mediating maps between (pre)strategies: *strong morphisms*, *weak morphisms* and *positive morphisms*. In the end, positive morphisms will turn out to be the most important; but they will all play a role in the development.

Equivalences. This zoology gets even more abundant when looking at the induced equivalences (ignoring strong isomorphism, as it does not support reindexing). Indeed, considering $f : \sigma \to \tau$ and $g : \tau \to \sigma$, we can require them to be inverses on the nose; or *up to symmetry*, *i.e.* $g \circ f \sim \mathrm{id}_\sigma$ and $f \circ g \sim \mathrm{id}_\tau$. This means we have four options:

$$\begin{array}{rllll}
\textit{weak equivalence:} & \partial_\tau \circ f \sim \partial_\sigma & \partial_\sigma \circ g \sim \partial_\tau & g \circ f \sim \mathrm{id}_\sigma & f \circ g \sim \mathrm{id}_\tau \\
\textit{positive equivalence:} & \partial_\tau \circ f \sim^+ \partial_\sigma & \partial_\sigma \circ g \sim^+ \partial_\tau & g \circ f \sim \mathrm{id}_\sigma & f \circ g \sim \mathrm{id}_\tau \\
\textit{weak isomorphism:} & \partial_\tau \circ f \sim \partial_\sigma & \partial_\sigma \circ g \sim \partial_\tau & g \circ f = \mathrm{id}_\sigma & f \circ g = \mathrm{id}_\tau \\
\textit{positive isomorphism:} & \partial_\tau \circ f \sim^+ \partial_\sigma & \partial_\sigma \circ g \sim^+ \partial_\tau & g \circ f = \mathrm{id}_\sigma & f \circ g = \mathrm{id}_\tau .
\end{array}$$

Fortunately, we shall see that they all end up inducing the same equivalence relation on (pre)strategies, which we call **thin equivalence** and write $\sigma \approx \tau$. This is thanks to some important properties of positive morphisms, that we explore now.

7.2.2 A Study of Positive Morphisms

We aim to prove that these equivalences actually coincide. On our way there, we shall develop a few properties of thin concurrent strategies that have proved crucial.

Strictification of positive equivalence. Our first step here is an easy consequence of *thin* that illustrates very well what that condition achieves: the observation that a positive equivalence actually *already is* an isomorphism.

Lemma 7.2.5. *Consider $\sigma, \tau : A$ (pre)strategies on tcg A.*
Any positive equivalence ($f : \sigma \Rightarrow \tau, g : \tau \Rightarrow \sigma$) is a positive isomorphism.

Proof. From the hypotheses, we get $g \circ f \sim \mathrm{id}_\sigma$, meaning

$$\langle g \circ f, \mathrm{id}_\sigma \rangle_x = \{(g(fs), s) \mid s \in x\} \in \mathcal{S}(\sigma)$$

for all $x \in \mathcal{C}(\sigma)$. But we also get $\partial_\sigma \circ g \circ f \sim^+ \partial_\sigma$, meaning

$$\langle \partial_\sigma \circ g \circ f, \partial_\sigma \rangle_x = \partial_\sigma \langle g \circ f, \mathrm{id}_\sigma \rangle_x \in \mathcal{S}_+(A)$$

for all $x \in \mathcal{C}(\sigma)$ – but so by *thin*, $\theta_x = \mathrm{id}_x$ for all $x \in \mathcal{C}(\sigma)$. So for all $x \in \mathcal{C}(\sigma)$ and $s \in x$, $g(f(s)) = s$. But any event $s \in \sigma$ appears in at least one configuration $[s] \in \mathcal{C}(\sigma)$, so $g \circ f = \mathrm{id}_\sigma$. Symmetrically, $f \circ g = \mathrm{id}_\tau$. □

Going back and forth around a positive equivalence yields a map $g \circ f \sim \mathrm{id}_\sigma$. But as the symmetries witnessing this display to positive symmetries in A, they must be identities as σ is thin. This property will be very important in the forthcoming development: the horizontal composition of positive isomorphisms will only yield a positive equivalence, and it is only via this lemma that it will turn out to be an isomorphism.

This strictification only holds for *positive* equivalences. To generalize it further, we shall prove that any mediating map between (pre)strategies can be made positive.

A fibration-like property. Let us investigate this *positivisation* process, and see how its crux is a fundamental *fibration-like* property of thin strategies.

Intuitively, the idea is simple: for $\boldsymbol{\sigma}$ and $\boldsymbol{\tau}$ thin (pre)strategies, a weak morphism

$$f : \boldsymbol{\sigma} \to \boldsymbol{\tau}$$

changes copy indices for both Player and Opponent. The idea is, for each $s^- \in \boldsymbol{\sigma}$, to set $f^+(s^-)$ as the unique event symmetric to $f(s)$ with the same copy index as s. By uniformity, we expect that the assignment f^+ should be completable to Player moves. But this would be intractable to formalize directly on events, so instead we reason on configurations. That f is only a *weak* morphism means that for $x \in \mathscr{C}(\boldsymbol{\sigma})$,

$$\langle \partial_{\boldsymbol{\sigma}}, \partial_{\boldsymbol{\tau}} \circ f \rangle_x : \partial_{\boldsymbol{\sigma}} x \cong_A \partial_{\boldsymbol{\tau}} f(x)$$

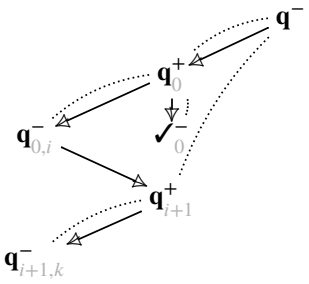
may not be in $\mathcal{S}_+(A)$. However, by Lemma 7.1.18 it factors uniquely as

$$\partial_{\boldsymbol{\sigma}} x \quad \overset{\theta^+}{\cong}{}^-_A \quad y \quad \overset{\theta^-}{\cong}{}^+_A \quad \partial_{\boldsymbol{\tau}} f(x)\,;$$

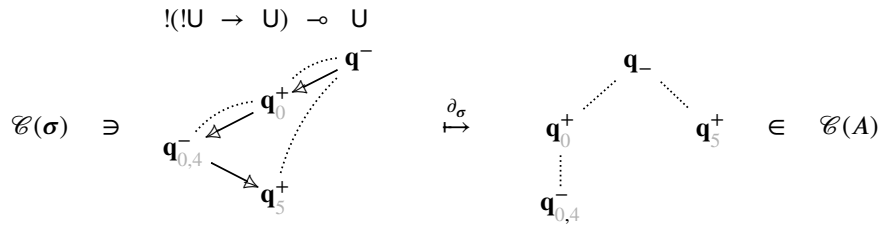
where y is understood as $\partial_{\boldsymbol{\tau}} f(x)$ with negative copy indices "set back" to those of $\partial_{\boldsymbol{\sigma}}(x)$. Now, we would like to set $f^+(x)$ by *transporting* $f(x)$ along this symmetry θ^-. In the presence of the condition *isofibration* of saturated strategies (Definition 7.1.16), this would be straightforward, but our thin strategies do not satisfy *isofibration*.

Instead, we shall prove and exploit an alternative fibration-like property for thin strategies: we shall find $z \in \mathscr{C}(\boldsymbol{\tau})$ such that $\partial_{\boldsymbol{\tau}}(z)$ is not *equal* to y, but *positively symmetric* to y. This is best first illustrated on an example. Consider

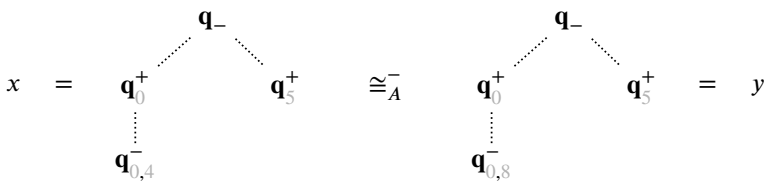
$$!(!\mathsf{U} \quad \to \quad \mathsf{U}) \quad \multimap \quad \mathsf{U}$$



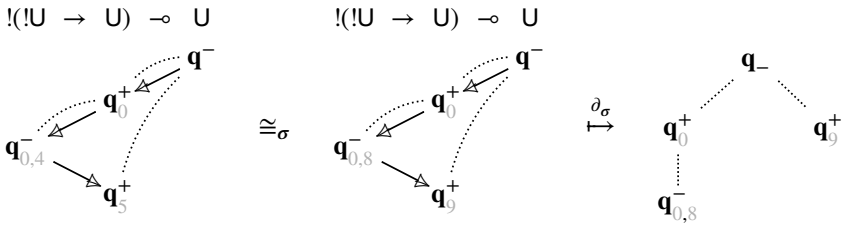
a symbolic representation of the strategy $\boldsymbol{\sigma} : A$ (with $A = !(!\mathsf{B} \multimap \mathsf{B}) \multimap \mathsf{B}$) obtained for instance by interpreting $\lambda f^{\mathbb{U}\to\mathbb{U}}.\, f\,(f\,\bot);\bot$. So $\boldsymbol{\sigma}$ admits as configuration:



Now, assume we have the negative symmetry reindexing the bottom move:



Clearly, σ cannot match y, but it can match it up to *positive symmetry*, with



positively isomorphic to y: Player can adapt to the change of Opponent copy indices by performing his own reindexing. This can be done systematically by exploiting uniformity of σ; or more precisely by a straightforward successive application of $\sim$*-receptivity* and *extension*. Finally, the process results in a unique configuration via *thin*. Altogether:

Lemma 7.2.6. *Consider A a tcg, $\sigma : A$ a (pre)strategy, $x \in \mathscr{C}(\sigma)$, $\theta^- : \partial_\sigma x \cong_A^- y$. Then, there are* unique *$\varphi : x \cong_\sigma z$ and $\theta^+ : y \cong_A^+ \partial_\sigma z$ such that*

$$\partial_\sigma \varphi = \theta^+ \circ \theta^- : \partial_\sigma x \cong_A \partial_\sigma z\,.$$

Proof. Existence. By induction on (the size of) x. If x is empty, this is clear. Otherwise, $x = x' \uplus \{s_1\}$ with s_1 maximal in x. By *restriction*, θ^- restricts to $\vartheta^- : \partial_\sigma x' \cong_A^- y'$. By induction hypothesis, there are $\varphi' : x' \cong_\sigma z'$ and $\vartheta^+ : y' \cong_A^+ \partial_\sigma z'$ such that

$$\partial_\sigma \varphi' = \vartheta^+ \circ \vartheta^- : \partial_\sigma x' \cong_A \partial_\sigma z'\,.$$

Furthermore, let us write $\theta^- = \vartheta^- \uplus \{(\partial_\sigma s_1, a_1)\} : \partial_\sigma x \cong_A^- y$ with $y = y' \uplus \{a_1\}$, by construction of ϑ^-. Now there are two cases, dependending on the polarity of s_1. If s_1, a_1 are negative, then by *extension* for $\mathcal{S}_+(A)$, there is an extension $\vartheta^+ \uplus \{(a_1^-, a_2^-)\} \in \mathcal{S}_+(A)$, set $\theta^+ = \vartheta^+ \uplus \{(a_1^-, a_2^-)\} : y \cong_A^+ \partial_\sigma z' \uplus \{a_2^-\}$. To sum up,

$$\varphi' : x' \cong_\sigma z'\,, \qquad x' \vdash_\sigma s_1^-\,, \qquad \partial_\sigma \varphi' \vdash (\partial_\sigma(s_1^-), a_2^-)\,,$$

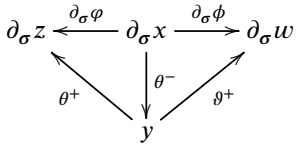
so by $\sim$-receptivity there is a unique $z' \vdash_\sigma s_2^-$ such that $\varphi' \vdash_{\mathcal{S}(\sigma)} (s_1^-, s_2^-)$ and $\partial_\sigma(s_2^-) = a_2^-$. Setting $\varphi = \varphi' \uplus \{(s_1^-, s_2^-)\}$ provides the extension of φ' as required.

Now, if s_1 is positive, then by *extension* on $\mathcal{S}(\sigma)$ there is $\varphi = \varphi' \uplus \{(s_1^+, s_2^+)\} \in \mathcal{S}(\sigma)$ with $z = z' \uplus \{s_2^+\} \in \mathscr{C}(\sigma)$. But writing $a_2^+ = \partial_\sigma s_2^+$, this means that we may set

$$\theta^+ \quad = \quad y' \uplus \{a_1^+\} \quad \overset{(\theta^-)^{-1}}{\cong_A} \quad \partial_\sigma x \quad \overset{\partial_\sigma \varphi}{\cong_A} \quad \partial_\sigma z' \uplus \{a_2^+\}$$

in $\mathcal{S}(A)$ by *groupoid* and which extends $\vartheta^+ : y' \cong^+_A \partial_\sigma z'$ with (a_1^+, a_2^+), so also $\theta^+ \in \mathcal{S}_+(A)$ by *+-receptivity*. By construction, we have $\partial_\sigma \varphi = \theta^+ \circ \theta^-$ as required.

Uniqueness. Consider alternative $\phi : x \cong_\sigma w$ and $\vartheta^+ : y \cong^+_A \partial_\sigma w$ such that $\partial_\sigma \phi = \vartheta^+ \circ \theta^- : \partial_\sigma x \cong_A \partial_\sigma w$. Then we have the following commuting diagram



in $\mathcal{S}(A)$, showing that $\partial_\sigma(\phi \circ \varphi^{-1}) \in \mathcal{S}_+(A)$, so $z = w$ and $\phi \circ \varphi^{-1} = \mathrm{id}_z$ by *thin*. It follows that $\varphi = \phi$, and then that $\theta^+ = \vartheta^+$, as required. □

This is reminiscent of Melliès' notion of *uniformity by bi-invariance* [Melliès, 2003].

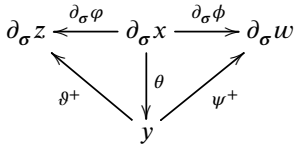
Via this fundamental lemma, we can mimic the consequences of the *isofibration* of saturated strategies, without saturation. We also mention the following generalization, which gives a similar fibration-like property without assuming that the symmetry along which the reindexing happens is negative – it is an easy consequence of the above.

Lemma 7.2.7. *Consider A a tcg, $\sigma : A$ a (pre)strategy, $x \in \mathscr{C}(\sigma)$, $\theta : \partial_\sigma x \cong_A y$.*
Then, there are unique *$\varphi : x \cong_\sigma z$ and $\vartheta^+ : y \cong^+_A \partial_\sigma z$ such that*

$$\partial_\sigma \varphi = \vartheta^+ \circ \theta : \partial_\sigma x \cong_A \partial_\sigma z\,.$$

Proof. Existence. First, we factor $\theta = \theta^+ \circ \theta^-$ for some $\theta^- : \partial_\sigma x \cong^-_A w$ and $\theta^+ : w \cong^+_A y$ as prescribed by Lemma 7.1.18. By Lemma 7.2.6, there are $\varphi : x \cong_\sigma z$ and $\psi^+ : w \cong^+_A \partial_\sigma z$ such that $\psi^+ \circ \theta^- = \partial_\sigma \varphi$. But then, setting $\vartheta^+ = \psi^+ \circ (\theta^+)^{-1} : y \cong^+_A \partial_\sigma z$, we have $\vartheta^+ \circ \theta = \psi^+ \circ \theta^- = \partial_\sigma \varphi$ as required.

Uniqueness. As in Lemma 7.2.6, for alternative $\phi : x \cong_\sigma w$ and $\psi^+ : y \cong^+_A \partial_\sigma w$ such that $\partial_\sigma \phi = \psi^+ \circ \theta : \partial_\sigma x \cong_A \partial_\sigma w$, we have the following commuting diagram



in $\mathcal{S}(A)$, showing $\partial_\sigma(\phi \circ \varphi^{-1}) \in \mathcal{S}_+(A)$, which entails uniqueness by *thin*. □

7.2.3 All Equivalences Coincide

Positivization of mediating maps. We deduce the desired positivization property:

Proposition 7.2.8. *Consider $\sigma, \tau : A$ (pre)strategies, $f : \sigma \to \tau$ a weak morphism.*
Then, there is a unique positive morphism $f^+ : \sigma \Rightarrow \tau$ such that $f \sim f^+$.

Proof. Existence. Consider $x \in \mathscr{C}(\boldsymbol{\sigma})$. By definition of weak morphisms we have

$$\langle \partial_{\boldsymbol{\sigma}}, \partial_{\boldsymbol{\tau}} \circ f \rangle_x = \{(\partial_{\boldsymbol{\sigma}}(s), \partial_{\boldsymbol{\tau}}(f(s))) \mid s \in x\} : \partial_{\boldsymbol{\sigma}} x \cong_A \partial_{\boldsymbol{\tau}}(f(x)),$$

and by Lemma 7.2.7, there are unique $\varphi_x : f(x) \cong_{\boldsymbol{\tau}} z_x$ and $\theta^+_x : \partial_{\boldsymbol{\sigma}} x \cong^+_A \partial_{\boldsymbol{\tau}} z_x$ s.t.

$$\partial_{\boldsymbol{\tau}} \varphi_x = \theta^+_x \circ \langle \partial_{\boldsymbol{\sigma}}, \partial_{\boldsymbol{\tau}} \circ f \rangle^{-1}_x,$$

let us call all that a *solution for* x. We set $f^+(x) = z_x$. This defines an action on configurations; we aim to induce a map by Lemma 7.1.9. Clearly, f^+ preserves cardinality as $x \simeq f(x)$ by local injectivity of f and $f(x) \cong_{\boldsymbol{\tau}} z$; we show that f^+ preserves unions.

Consider $x, y \in \mathscr{C}(\boldsymbol{\sigma})$ such that $x \cup y \in \mathscr{C}(\boldsymbol{\sigma})$. As above, we have

$$\varphi_{x \cup y} : f(x \cup y) \cong_{\boldsymbol{\tau}} z_{x \cup y}, \qquad \theta^+_{x \cup y} : \partial_{\boldsymbol{\sigma}}(x \cup y) \cong^+_A \partial_{\boldsymbol{\tau}} z_{x \cup y}$$

such that $\partial_{\boldsymbol{\tau}} \varphi_{x \cup y} = \theta^+_{x \cup y} \circ \langle \partial_{\boldsymbol{\sigma}}, \partial_{\boldsymbol{\tau}} \circ f \rangle^{-1}_{x \cup y}$, *i.e.* a solution for $x \cup y$. But as $f(x \cup y) = f(x) \cup f(y)$, $\partial_{\boldsymbol{\sigma}}(x \cup y) = \partial_{\boldsymbol{\sigma}}(x) \cup \partial_{\boldsymbol{\sigma}}(y)$ and $\langle \partial_{\boldsymbol{\sigma}}, \partial_{\boldsymbol{\tau}} \circ f \rangle_{x \cup y} = \langle \partial_{\boldsymbol{\sigma}}, \partial_{\boldsymbol{\tau}} \circ f \rangle_x \cup \langle \partial_{\boldsymbol{\sigma}}, \partial_{\boldsymbol{\tau}} \circ f \rangle_y$, by the *restriction* axiom it restricts to solutions for x and y, and must hence be their componentwise union. In particular, $f^+(x \cup y) = f^+(x) \cup f^+(y)$ as needed.

For the action on symmetries, for $\omega : x \cong_{\boldsymbol{\sigma}} x'$ we set as $\widetilde{f^+}(\omega)$ the composite

$$f^+(x) \overset{\varphi_x^{-1}}{\cong_{\boldsymbol{\tau}}} f(x) \overset{f(\omega)}{\cong_{\boldsymbol{\tau}}} f(x') \overset{\varphi_{x'}}{\cong_{\boldsymbol{\tau}}} f^+(x'),$$

clearly commuting with domain and codomain, so that by Lemma 7.1.9 there is a unique map of ess $f^+ : \boldsymbol{\sigma} \to \boldsymbol{\tau}$ extending the action of f^+ on configurations. For all $x \in \mathscr{C}(\boldsymbol{\sigma})$,

$$\langle \partial_{\boldsymbol{\sigma}}, \partial_{\boldsymbol{\tau}} \circ f^+ \rangle_x = \{(\partial_{\boldsymbol{\sigma}} s, \partial_{\boldsymbol{\tau}}(f^+(s))) \mid s \in x\} = \partial_{\boldsymbol{\tau}}(\varphi_x) \circ \langle \partial_{\boldsymbol{\sigma}}, \partial_{\boldsymbol{\tau}} \circ f \rangle_x = \theta^+_x \in \mathcal{S}_+(A)$$

so that $f^+ : \boldsymbol{\sigma} \Rightarrow \boldsymbol{\tau}$ is a positive morphism as required.

Uniqueness. Consider $f^+_1, f^+_2 : \boldsymbol{\sigma} \Rightarrow \boldsymbol{\tau}$ two symmetric positive morphisms. This means for all $x \in \mathscr{C}(\boldsymbol{\sigma})$ we have $\langle f^+_1, f^+_2 \rangle_x \in \mathcal{S}(\boldsymbol{\tau})$, together with

$$\begin{array}{rcl} \langle \partial_{\boldsymbol{\sigma}}, \partial_{\boldsymbol{\tau}} \circ f^+_1 \rangle_x & : & \partial_{\boldsymbol{\sigma}}(x) \cong^+_A \partial_{\boldsymbol{\tau}}(f^+_1(x)), \\ \langle \partial_{\boldsymbol{\sigma}}, \partial_{\boldsymbol{\tau}} \circ f^+_2 \rangle_x & : & \partial_{\boldsymbol{\sigma}}(x) \cong^+_A \partial_{\boldsymbol{\tau}}(f^+_2(x)), \end{array}$$

but $\langle \partial_{\boldsymbol{\sigma}}, \partial_{\boldsymbol{\tau}} \circ f^+_2 \rangle_x \circ \langle \partial_{\boldsymbol{\sigma}}, \partial_{\boldsymbol{\tau}} \circ f^+_1 \rangle^{-1}_x = \partial_{\boldsymbol{\tau}} \langle f^+_1, f^+_2 \rangle_x$, so $\partial_{\boldsymbol{\tau}} \langle f^+_1, f^+_2 \rangle_x \in \mathcal{S}_+(A)$, so that $\langle f^+_1, f^+_2 \rangle_x$ identity by *thin*. Thus $f^+_1(x) = f^+_2(x)$ and $f^+_1 = f^+_2$ by Lemma 6.1.12. □

From this, we may finally deduce the following:

Corollary 7.2.9. *For $\boldsymbol{\sigma}, \boldsymbol{\tau}$: A (pre)strategies on tcg A, the following are equivalent:*

(1) $\boldsymbol{\sigma}$ *and* $\boldsymbol{\tau}$ *are* weakly equivalent*,*
(2) $\boldsymbol{\sigma}$ *and* $\boldsymbol{\tau}$ *are* positively equivalent*,*
(3) $\boldsymbol{\sigma}$ *and* $\boldsymbol{\tau}$ *are* weakly isomorphic*,*
(4) $\boldsymbol{\sigma}$ *and* $\boldsymbol{\tau}$ *are* positively isomorphic*,*

and we write $\boldsymbol{\sigma} \approx \boldsymbol{\tau}$ if that is the case.

Proof. *(1)* $\Rightarrow$ *(2)* by Proposition 7.2.8; *(2)* $\Rightarrow$ *(4)* by Lemma 7.2.5; *(4)* $\Rightarrow$ *(3)* by definition; and *(3)* $\Rightarrow$ *(1)* by definition. □

These four variants induce the same equivalence relation on (pre)strategies, but the equivalences themselves (with the mediating maps) are not the same! In particular, going from weak equivalence to positive equivalence requires changing the maps.

From all this, it should be clear why we adopt positive morphisms and positive isomorphisms as the canonical device to compare (pre)strategies, used throughout this monograph. The developments in this section will be used often, and notably to ensure that positive morphisms are preserved by forthcoming operations on strategies.

7.2.4 Constructing Positive Morphisms

In plain concurrent games, we have argued that working with morphisms of strategies following their concrete definition – that is, as functions acting on the sets of events – is often impractical. For this purpose we introduced Lemma 6.3.4, which shows that in order to build a morphism between strategies, one can ignore trailing Opponent events: it suffices to define the action of a morphism on +-covered configurations only. In the presence of symmetry, this need for tools to help construct morphisms is strengthened, so we need a similar lemma. As with Lemma 6.3.4, positive morphisms will act on +-covered configurations; but also on +-covered *symmetries*:

Definition 7.2.10. *Consider A a tcg, and σ a (pre)strategy on A.*

*A symmetry $\theta \in \mathcal{S}(\sigma)$ is **+-covered** if its domain (equivalently, its codomain) is. We write $\mathcal{S}^+(\sigma)$ for the set of +-covered symmetries of σ.*

The reader should take particular care not to confuse the set of *+-covered symmetries of a thin prestrategy* $\sigma : A$, written $\mathcal{S}^+(\sigma)$, with the set of *positive symmetries of a tcg* A, written $\mathcal{S}_+(A)$. The former asks that maximal events are positive, relying on the fact that symmetries are order-isomorphisms and so preserve maximal events; while the latter is part of the data of a tcg, intuitively capturing those symmetries only reindexing Player moves. Note that only a tcg has positive and negative symmetries, not a (pre)strategy.

With this notion, we prove the following generalization of Lemma 6.3.4:

Lemma 7.2.11. *Consider $\sigma, \tau : A$ two strategies. Assume there is a function $f : \mathcal{C}^+(\sigma) \to \mathcal{C}^+(\tau)$ preserving unions, and monotone functions*

$$\widetilde{f} : \mathcal{S}^+(\sigma) \to \mathcal{S}^+(\tau), \qquad \check{f} : \mathcal{C}^+(\sigma) \to \mathcal{S}^+(A),$$

compatible with dom *and* cod *as in Lemma 7.1.9, such that for all $x \in \mathcal{C}^+(\sigma)$,*

$$\check{f}(x) : \partial_\sigma\, x \cong^+_A \partial_\tau\, f(x),$$

and such that for all $\theta : x \cong_\sigma y \in \mathcal{S}^+(\sigma)$, we have $\partial_\tau(\widetilde{f}(\theta)) = \check{f}(y) \circ (\partial_\sigma\, \theta) \circ \check{f}(x)^{-1}$.

Then, there is a unique positive $\widehat{f} : \sigma \Rightarrow \tau$ such that for all $x \in \mathcal{C}^+(\sigma)$, $\widehat{f}\, x = f\, x$.

Proof. Existence. We first extend f to all configurations, via the following elaboration of the reasoning in Lemma 6.3.4. Consider $x \in \mathscr{C}(\boldsymbol{\sigma})$. There is a unique $y \in \mathscr{C}^+(\boldsymbol{\sigma})$ such that $y \subseteq^- x$, obtained by removing trailing events. Now, by hypothesis we have

$$\check{f}(y) : \partial_{\boldsymbol{\sigma}}\, y \cong^+_A \partial_{\boldsymbol{\tau}}\, f(y)\,,$$

and $\partial_{\boldsymbol{\sigma}}\, y \subseteq^- \partial_{\boldsymbol{\sigma}}\, x$, thus by *extension*, there is $\check{f}(y) \subseteq^- \theta : \partial_{\boldsymbol{\sigma}}\, x \cong^+_A z$; it is actually unique because if we also have $\check{f}(y) \subseteq^- \theta' : \partial_{\boldsymbol{\sigma}}\, x \cong^+_A z'$, then $\theta' \circ \theta^{-1} : z \cong^+_A z'$ is a negative extension of $\mathrm{id}_{\partial_{\boldsymbol{\sigma}}\, x}$ hence is also a negative symmetry, and thus must be an identity by *orthogonality*. Now that $\theta : \partial_{\boldsymbol{\sigma}}\, x \cong^+_A z$ is uniquely determined, we note that $\partial_{\boldsymbol{\tau}}\, f(y) \subseteq^- z$ inducing a unique $f(y) \subseteq^- u \in \mathscr{C}(\boldsymbol{\tau})$ so that $\partial_{\boldsymbol{\tau}}\, u = z$ by *$-$-discrete opfibration*; we set $f(x) = u$ and $\check{f}(x) = \theta$. From the uniqueness properties of z, θ, u it is straightforward that f, extended in this way, preserves unions.

We must similarly extend $\widetilde{f}$ to all symmetries. If $\theta : x \cong_{\boldsymbol{\sigma}} x'$, then as above there is a unique $\varphi \subseteq^- \theta$ such that $\varphi \in \mathcal{S}^+(\boldsymbol{\sigma})$, write $\varphi : y \cong_{\boldsymbol{\sigma}} y'$, obtained by removing trailing (pairs of) negative moves. We have then $\widetilde{f}(\varphi) : f(y) \cong_{\boldsymbol{\tau}} f(y')$. Then

$$\psi \quad = \quad \partial_{\boldsymbol{\tau}}\, f(x) \overset{\check{f}(x)^{-1}}{\cong^+_A} \partial_{\boldsymbol{\sigma}}\, x \overset{\partial_{\boldsymbol{\sigma}}\, \theta}{\cong_A} \partial_{\boldsymbol{\sigma}}\, x' \overset{\check{f}(x')}{\cong^+_A} \partial_{\boldsymbol{\tau}}\, f(x')$$

is a negative extension of $\check{f}(y') \circ (\partial_{\boldsymbol{\sigma}}\, \varphi) \circ \check{f}(y)^{-1}$, which by hypothesis is $\partial_{\boldsymbol{\tau}}(\widetilde{f}(\theta))$. Thus by *$\sim$-receptivity* of $\boldsymbol{\tau}$, there is a unique $\widetilde{f}(\theta) \subseteq^- \theta'$ s.t. $\partial_{\boldsymbol{\tau}}\, \theta' = \psi$, and we set $\widetilde{f}(\theta) = \theta'$.

Clearly, this assignment is compatible with the extension of f with respect to domain and codomain, and thus by Lemma 7.1.9, there is a unique map of ess $\widehat{f} : \boldsymbol{\sigma} \to \boldsymbol{\tau}$ extending f. We must show that $\partial_{\boldsymbol{\sigma}} \sim^+ \partial_{\boldsymbol{\tau}} \circ f$, which amounts to showing

$$\check{f}(x) \quad = \quad \langle \partial_{\boldsymbol{\sigma}}, \partial_{\boldsymbol{\tau}} \circ f \rangle_x \tag{7.2}$$

for all $x \in \mathscr{C}(\boldsymbol{\sigma})$, which we prove by induction on x. If x is empty, there is nothing to prove. If x is not $+$-covered, then there is unique $y \in \mathscr{C}^+(\boldsymbol{\sigma})$ such that $y \subset^- x$. By induction hypothesis, $\check{f}(y) = \langle \partial_{\boldsymbol{\sigma}}, \partial_{\boldsymbol{\tau}} \circ f \rangle_y$, and there is a covering chain

$$y \overset{s_1^-}{\multimap\!\subset} \ \ldots \ \overset{s_n^-}{\multimap\!\subset} x$$

by induction on which it is immediate that (7.2) holds for x as well by definition of the extension of $\check{f}$. Now if x is $+$-covered, it has a maximal positive event s. Setting $y = x \setminus \{s\}$, by induction hypothesis (7.2) holds for $y \in \mathscr{C}(\boldsymbol{\sigma})$, but then it holds for x by monotonicity of $\check{f}$. Thus we have $\partial_{\boldsymbol{\sigma}} \sim^+ \partial_{\boldsymbol{\tau}} \circ \widehat{f}$ as required.

Rigidity. Similar to the reasoning in Lemma 6.3.4.

Uniqueness. Consider g another such extension. We first show by induction on x that for all $x \in \mathscr{C}(\boldsymbol{\sigma})$, we have $f(x) = g(x)$ and $\check{f}(x) = \langle \partial_{\boldsymbol{\sigma}}, \partial_{\boldsymbol{\tau}} \circ g \rangle_x$. If x is empty there is nothing to prove. If x is not $+$-covered, there is $y \in \mathscr{C}^+(\boldsymbol{\sigma})$ such that $y \subset^- x$. By induction hypothesis, we have $f(y) = g(y)$ and $\check{f}(y) = \langle \partial_{\boldsymbol{\sigma}}, \partial_{\boldsymbol{\tau}} \circ g \rangle_y$. Now we have

$$\langle \partial_{\boldsymbol{\sigma}}, \partial_{\boldsymbol{\tau}} \circ g \rangle_y \subset^- \langle \partial_{\boldsymbol{\sigma}}, \partial_{\boldsymbol{\tau}} \circ g \rangle_x \in \mathcal{S}^+(A)\,,$$

forcing $g(x)$ to coincide with the construction of $f(x)$ above; that $\check{f}(x) = \langle \partial_{\boldsymbol{\sigma}}, \partial_{\boldsymbol{\tau}} \circ g \rangle_x$ is immediate from the construction of $\check{f}(x)$ following a covering chain of $y \subset^- x$. Finally, if x is $+$-covered, the result follows by $f(x) = g(x)$ and monotonicity of $\check{f}$. □

As before, when using this lemma we will not always explicitely spell out the action on +-covered symmetries when it is transparent from that on +-covered configurations, similarly we will detail $\check{f}$ only when it is non-trivial. Though unlike for Lemma 7.1.9, it remains unclear if $\widetilde{f}$ and $\check{f}$ are necessarily unique here.

We mention in passing two direct consequences of the above. First, we have:

Corollary 7.2.12. *Consider $f, g : \boldsymbol{\sigma} \Rightarrow \boldsymbol{\tau}$ two positive morphisms between strategies. If for all $x^{\boldsymbol{\sigma}} \in \mathscr{C}^+(\boldsymbol{\sigma})$, we have $f(x^{\boldsymbol{\sigma}}) = g(x^{\boldsymbol{\sigma}})$, then $f = g$.*

Proof. By uniqueness in Lemma 7.2.11. □

And then, the following is sometimes useful to construct strong isomorphisms:

Corollary 7.2.13. *Consider $\boldsymbol{\sigma}, \boldsymbol{\tau} : A$ strategies on tcg A. Assume there are*

$$f : \mathscr{C}^+(\boldsymbol{\sigma}) \simeq \mathscr{C}^+(\boldsymbol{\tau}) \qquad \widetilde{f} : \mathscr{S}^+(\boldsymbol{\sigma}) \simeq \mathscr{S}^+(\boldsymbol{\tau})$$

order-isomorphisms compatible with dom, cod*, and display maps.*

Then, there is a strong isomorphism $\widehat{f} : \boldsymbol{\sigma} \cong \boldsymbol{\tau}$ s.t. for all $x \in \mathscr{C}^+(\boldsymbol{\sigma})$, $\widehat{f}(x) = f(x)$.

7.3 Composition and Copycat

Now that we have introduced thin strategies and the required tools to mediate between them, we are in position to compose them and introduce copycat. For both, the idea is to keep the structure developed in Chapter 6, only enriching it with symmetry.

For composition, the symmetries will exactly be pairs of causally compatible symmetries, as summarized by the following extension of Proposition 6.2.1:

Proposition 7.3.1. *Consider A, B, C tcgs, and $\boldsymbol{\sigma} : A \vdash B$ and $\boldsymbol{\tau} : B \vdash C$ strategies. Then there is a strategy $\boldsymbol{\tau} \odot \boldsymbol{\sigma} : A \vdash C$, unique up to strong iso, such that there are*

$$\begin{array}{rcl} (- \odot -) & : & \{(x^{\boldsymbol{\tau}}, x^{\boldsymbol{\sigma}}) \in \mathscr{C}^+(\boldsymbol{\tau}) \times \mathscr{C}^+(\boldsymbol{\sigma}) \mid \textit{causally compatible}\} \simeq \mathscr{C}^+(\boldsymbol{\tau} \odot \boldsymbol{\sigma}) \\ (- \odot -) & : & \{(\theta^{\boldsymbol{\tau}}, \theta^{\boldsymbol{\sigma}}) \in \mathscr{S}^+(\boldsymbol{\tau}) \times \mathscr{S}^+(\boldsymbol{\sigma}) \mid \textit{causally compatible}\} \simeq \mathscr{S}^+(\boldsymbol{\tau} \odot \boldsymbol{\sigma}) \end{array}$$

order-isomorphisms commuting with dom, cod*, and such that we have*

$$\partial_{\boldsymbol{\tau} \odot \boldsymbol{\sigma}}(\theta^{\boldsymbol{\tau}} \odot \theta^{\boldsymbol{\sigma}}) = \theta^{\boldsymbol{\sigma}}_A \parallel \theta^{\boldsymbol{\tau}}_C \,,$$

for all $\theta^{\boldsymbol{\sigma}} \in \mathscr{S}^+(\boldsymbol{\sigma})$ and $\theta^{\boldsymbol{\tau}} \in \mathscr{S}^+(\boldsymbol{\tau})$ causally compatible.

As without symmetry, this proposition packages almost all the "working concurrent game semanticist" needs to know about the composition of thin strategies. We shall gradually work our way towards its proof, and the concrete construction of composition.

In the statement, we have used for the display of symmetries the same notation as for the display of configurations in Chapter 6, that is to say, we typically write $\theta^{\boldsymbol{\sigma}} \in \mathscr{S}(\boldsymbol{\sigma})$ for symmetries in $\boldsymbol{\sigma}$ for $\boldsymbol{\sigma}$ a (pre)strategy, and we write its display as $\partial_{\boldsymbol{\sigma}}\, \theta^{\boldsymbol{\sigma}} = \theta^{\boldsymbol{\sigma}}_A \vdash \theta^{\boldsymbol{\sigma}}_B$. We have also used the notion of causal compatibility for symmetries:

Definition 7.3.2. *Take $\sigma : A \vdash C$, $\tau : B \vdash C$ prestrategies, $\theta^\sigma \in \mathcal{S}(\sigma), \theta^\tau \in \mathcal{S}(\tau)$.*

*We say that θ^σ and θ^τ are **matching** if $\theta^\sigma_B = \theta^\tau_B$; and **causally compatible** if the domains $\mathsf{dom}(\theta^\sigma), \mathsf{dom}(\theta^\tau)$ are (equivalently, if the codomains $\mathsf{cod}(\theta^\sigma), \mathsf{cod}(\theta^\tau)$ are).*

7.3.1 Interactions with Symmetry

We now set to construct concretely the composition of thin (pre)strategies. The first step is simply to enrich Definition 6.2.3 with symmetry, for A, B and C tcgs.

Definition 7.3.3. *An **interaction** on A, B, C is an ess $\mu = (|\mu|, \leq_\mu, \#_\mu, \mathcal{S}(\mu))$ with*

$$\partial : |\mu| \to |A \parallel B \parallel C|$$

a display map subject to the following conditions:

rule-abiding:	*for all $x \in \mathscr{C}(\mu)$, $\partial(x) \in \mathscr{C}(A \parallel B \parallel C)$,*
locally injective:	*for all $s_1, s_2 \in x \in \mathscr{C}(\mu)$, if $\partial(s_1) = \partial(s_2)$ then $s_1 = s_2$,*
$\sim$-preserving:	*for all $\theta \in \mathcal{S}(\mu)$, $\partial(\theta) \in \mathcal{S}(A \parallel B \parallel C)$,*
thin:	*for all $\theta \in \mathcal{S}(\mu)$, writing $\partial(\theta) \in \theta_A \parallel \theta_B \parallel \theta_C$,*
	if $\theta_A \in \mathcal{S}_-(A)$ and $\theta_C \in \mathcal{S}_+(C)$, then $\theta = \mathrm{id}_x$ for some $x \in \mathscr{C}(\mu)$.

i.e. $\partial : \mu \to A \parallel B \parallel C$ *is a* map of event structures with symmetry*; and is* thin*.*

Fix for now prestrategies $\sigma : A \vdash B$ and $\tau : B \vdash C$. Following the developments in Chapter 6, we have already constructed an interaction $\tau \circledast \sigma$ in the sense of Definition 6.2.3, that we need to enrich with symmetry. This relies on the following observation, letting us "zip" causally compatible symmetries into a bijection in $\tau \circledast \sigma$:

Lemma 7.3.4. *Consider $\theta^\sigma : x^\sigma \cong_\sigma y^\sigma$ and $\theta^\tau : x^\tau \cong_\tau y^\tau$ causally compatible.*

Then, there exists a unique bijection $\theta^\tau \circledast \theta^\sigma : x^\tau \circledast x^\sigma \simeq y^\tau \circledast y^\sigma$ such that

$$\begin{array}{ccccc} x^\sigma \parallel x^\tau_C & \xleftarrow{\Pi_\sigma} & x^\tau \circledast x^\sigma & \xrightarrow{\Pi_\tau} & x^\sigma_A \parallel x^\tau \\ {\scriptstyle \theta^\sigma \parallel \theta^\tau_C}\downarrow & & \vdots\ {\scriptstyle \theta^\tau \circledast \theta^\sigma} & & \downarrow{\scriptstyle \theta^\sigma_A \parallel \theta^\tau} \\ y^\sigma \parallel y^\tau_C & \xleftarrow[\Pi_\sigma]{} & y^\tau \circledast y^\sigma & \xrightarrow[\Pi_\tau]{} & y^\sigma_A \parallel y^\tau \end{array} \tag{7.3}$$

commutes in the category of finite sets and bijections.

Proof. Existence. As θ^σ and θ^τ are matching, the outer diagram commutes by Lemma 6.2.12. This lets us define the bijection $\theta^\tau \circledast \theta^\sigma : x^\tau \circledast x^\sigma \simeq y^\tau \circledast y^\sigma$ as either path along the diagram above. The condition holds by construction.

Uniqueness. Clearly, $\theta^\tau \circledast \theta^\sigma$ is uniquely determined by the diagram. □

This invites us to set $\mathcal{S}(\tau \circledast \sigma)$ as including all $\theta^\tau \circledast \theta^\sigma$ for $\theta^\sigma \in \mathcal{S}(\sigma)$ and $\theta^\tau \in \mathcal{S}(\tau)$ causally compatible. It remains to prove that this satisfies the required conditions.

Lemma 7.3.5. *Setting $\mathcal{S}(\boldsymbol{\tau} \circledast \boldsymbol{\sigma})$ to comprise all $\theta^{\boldsymbol{\tau}} \circledast \theta^{\boldsymbol{\sigma}}$ for $\theta^{\boldsymbol{\sigma}} \in \mathcal{S}(\boldsymbol{\sigma})$ and $\theta^{\boldsymbol{\tau}} \in \mathcal{S}(\boldsymbol{\tau})$ causally compatible defines an ess $(|\boldsymbol{\tau} \circledast \boldsymbol{\sigma}|, \leq_{\boldsymbol{\tau}\circledast\boldsymbol{\sigma}}, \#_{\boldsymbol{\tau}\circledast\boldsymbol{\sigma}}, \mathcal{S}(\boldsymbol{\tau} \circledast \boldsymbol{\sigma}))$, yielding*

$$\begin{array}{ccccc} & & \boldsymbol{\tau} \circledast \boldsymbol{\sigma} & & \\ & {\scriptstyle \Pi_{\sigma}} \swarrow & \smile & \searrow {\scriptstyle \Pi_{\tau}} & \\ \boldsymbol{\sigma} \parallel C & & & & A \parallel \boldsymbol{\tau} \\ & {\scriptstyle \partial_{\sigma}\parallel C} \searrow & & \swarrow {\scriptstyle A \parallel \partial_{\tau}} & \\ & & A \parallel B \parallel C & & \end{array} \tag{7.4}$$

a pullback in the category of ess and their maps.

Proof. We verify the axioms of isomorphism families. First, *groupoid* axioms follow from the definition. For *restriction*, by definition of $\theta^{\boldsymbol{\sigma}}$ and $\theta^{\boldsymbol{\tau}}$ matching we have

$$\begin{array}{ccccc} x^{\boldsymbol{\sigma}} \parallel x^{\boldsymbol{\tau}}_C & \xleftarrow{\mathsf{dom}} & \varphi[x^{\boldsymbol{\sigma}}, x^{\boldsymbol{\tau}}] & \xrightarrow{\mathsf{cod}} & x^{\boldsymbol{\sigma}}_A \parallel x^{\boldsymbol{\tau}} \\ \downarrow{\scriptstyle \theta^{\boldsymbol{\sigma}}\parallel\theta^{\boldsymbol{\tau}}_C} & & & & \downarrow{\scriptstyle \theta^{\boldsymbol{\sigma}}_A\parallel\theta^{\boldsymbol{\tau}}} \\ y^{\boldsymbol{\sigma}} \parallel y^{\boldsymbol{\tau}}_C & \xleftarrow[\mathsf{dom}]{} & \varphi[y^{\boldsymbol{\sigma}}, y^{\boldsymbol{\tau}}] & \xrightarrow[\mathsf{cod}]{} & y^{\boldsymbol{\sigma}}_A \parallel y^{\boldsymbol{\tau}} \end{array} \tag{7.5}$$

commuting; assume now that $u^{\boldsymbol{\sigma}} \circledast u^{\boldsymbol{\tau}} \subseteq x^{\boldsymbol{\tau}} \circledast x^{\boldsymbol{\sigma}}$; *i.e.* we have $u^{\boldsymbol{\sigma}} \subseteq x^{\boldsymbol{\sigma}}$ and $u^{\boldsymbol{\tau}} \subseteq x^{\boldsymbol{\tau}}$ causally compatible. But then by *restriction* for $\boldsymbol{\sigma}$ and $\boldsymbol{\tau}$, we have

$$\begin{array}{ccccc} u^{\boldsymbol{\sigma}} \parallel u^{\boldsymbol{\tau}}_C & \xleftarrow{\mathsf{dom}} & \varphi[u^{\boldsymbol{\sigma}}, u^{\boldsymbol{\tau}}] & \xrightarrow{\mathsf{cod}} & u^{\boldsymbol{\sigma}}_A \parallel u^{\boldsymbol{\tau}} \\ \downarrow{\scriptstyle \vartheta^{\boldsymbol{\sigma}}\parallel\vartheta^{\boldsymbol{\tau}}_C} & & & & \downarrow{\scriptstyle \vartheta^{\boldsymbol{\sigma}}_A\parallel\vartheta^{\boldsymbol{\tau}}} \\ v^{\boldsymbol{\sigma}} \parallel v^{\boldsymbol{\tau}}_C & \xleftarrow[\mathsf{dom}]{} & \varphi[v^{\boldsymbol{\sigma}}, v^{\boldsymbol{\tau}}] & \xrightarrow[\mathsf{cod}]{} & v^{\boldsymbol{\sigma}}_A \parallel v^{\boldsymbol{\tau}} \end{array}$$

with $\vartheta^{\boldsymbol{\sigma}} \subseteq \theta^{\boldsymbol{\sigma}}$ and $\vartheta^{\boldsymbol{\tau}} \subseteq \theta^{\boldsymbol{\tau}}$; in particular the diagram still commutes. *Restriction* follows by definition and Lemma 6.2.12.

For *extension*, consider again the diagram (7.5). It suffices to deal with one-step extensions, the general case follows by induction. Furthermore, we focus on an extension

$$\varphi[x^{\boldsymbol{\sigma}}, x^{\boldsymbol{\tau}}] \quad \overset{((1,s),(2,t))}{\multimap\!\subset} \quad \varphi[u^{\boldsymbol{\sigma}}, u^{\boldsymbol{\tau}}]\,,$$

with $x^{\boldsymbol{\sigma}} \overset{s}{\multimap\!\subset} u^{\boldsymbol{\sigma}}$ and $x^{\boldsymbol{\tau}} \overset{t}{\multimap\!\subset} u^{\boldsymbol{\tau}}$, *i.e.* a synchronized event between $\boldsymbol{\sigma}$ and $\boldsymbol{\tau}$ – where $\partial_{\boldsymbol{\sigma}} s$ and $\partial_{\boldsymbol{\tau}} t$ occur in B – the other cases are simpler. Moreover, we assume $\mathrm{pol}_{\boldsymbol{\sigma}}(s) = +$, the other case is symmetric. Now, by *extension* for $\boldsymbol{\sigma}$, there is $\theta^{\boldsymbol{\sigma}} \multimap\!\subset \vartheta^{\boldsymbol{\sigma}} : u^{\boldsymbol{\sigma}} \cong_{\boldsymbol{\sigma}} v^{\boldsymbol{\tau}}$. By $\sim$-*preserving* for $\boldsymbol{\sigma}$, $\partial_{\boldsymbol{\sigma}}\vartheta^{\boldsymbol{\sigma}} = \theta^{\boldsymbol{\sigma}}_A \parallel \vartheta^{\boldsymbol{\sigma}}_B$ with $\theta^{\boldsymbol{\sigma}}_B \multimap\!\subset \vartheta^{\boldsymbol{\sigma}}_B : u^{\boldsymbol{\sigma}}_B \cong_B v^{\boldsymbol{\sigma}}_B$. But then, $\sim$-*receptivity* of $\boldsymbol{\tau}$ exactly gives us a matching extension $\theta^{\boldsymbol{\tau}} \subseteq \vartheta^{\boldsymbol{\tau}}$ such that

$$\begin{array}{ccccc} u^{\boldsymbol{\sigma}} \parallel x^{\boldsymbol{\tau}}_C & \xleftarrow{\mathsf{dom}} & \varphi[u^{\boldsymbol{\sigma}}, u^{\boldsymbol{\tau}}] & \xrightarrow{\mathsf{cod}} & x^{\boldsymbol{\sigma}}_A \parallel u^{\boldsymbol{\tau}} \\ \downarrow{\scriptstyle \vartheta^{\boldsymbol{\sigma}}\parallel\theta^{\boldsymbol{\tau}}_C} & & & & \downarrow{\scriptstyle \theta^{\boldsymbol{\sigma}}_A\parallel\vartheta^{\boldsymbol{\tau}}} \\ v^{\boldsymbol{\sigma}} \parallel y^{\boldsymbol{\tau}}_C & \xleftarrow[\mathsf{dom}]{} & \varphi[v^{\boldsymbol{\sigma}}, v^{\boldsymbol{\tau}}] & \xrightarrow[\mathsf{cod}]{} & y^{\boldsymbol{\sigma}}_A \parallel v^{\boldsymbol{\tau}} \end{array}$$

commutes, which provides us with the required extension by Lemma 6.2.12.

It is clear by definition (see the diagram of Lemma 7.3.4) that Π_{σ} and Π_{τ} preserve symmetry. We skip the universal property, straightforward via Lemma 7.1.9. □

At last, this provides us with the ingredients for the interaction.

Proposition 7.3.6. *There is an interaction $\tau \circledast \sigma$, unique up to iso, s.t. there are*

$$\begin{array}{lcl} (- \circledast -) & : & \{(x^{\tau}, x^{\sigma}) \in \mathscr{C}(\tau) \times \mathscr{C}(\sigma) \mid x^{\sigma}, x^{\tau} \textit{ causally compatible}\} \simeq \mathscr{C}(\tau \circledast \sigma) \\ (- \circledast -) & : & \{(\theta^{\tau}, \theta^{\sigma}) \in \mathcal{S}(\tau) \times \mathcal{S}(\sigma) \mid \theta^{\sigma}, \theta^{\tau} \textit{ causally compatible}\} \simeq \mathcal{S}(\tau \circledast \sigma) \end{array}$$

order-isomorphisms commuting with dom *and* cod*, and satisfying*

$$\partial_{\tau \circledast \sigma}(\theta^{\tau} \circledast \theta^{\sigma}) = \theta^{\sigma}_{A} \parallel \theta_{B} \parallel \theta^{\tau}_{C}$$

for all $\theta^{\sigma} \in \mathcal{S}(\sigma)$ and $\theta^{\tau} \in \mathcal{S}(\tau)$ causally compatible.

Proof. Existence. The ess $\tau \circledast \sigma$ is defined in Lemma 7.3.5, with display map $\partial_{\tau \circledast \sigma} : \tau \circledast \sigma \to A \parallel B \parallel C$ defined as either way around (7.4). The order-iso for configurations comes from Proposition 6.2.8, while for symmetries it is follows from Lemma 7.3.4 (it is immediate from (7.3) that θ^{σ} and θ^{τ} can be recovered from $\theta^{\tau} \circledast \theta^{\sigma}$).

For *thin*, take $\theta^{\tau} \circledast \theta^{\sigma} \in \mathcal{S}(\tau \circledast \sigma)$ s.t. $\theta^{\sigma}_{A} \in \mathcal{S}_{-}(A)$ and $\theta^{\tau}_{C} \in \mathcal{S}_{+}(A)$. Consider:

$$\begin{array}{ccccc} x^{\sigma} \parallel x^{\tau}_{C} & \xleftarrow{\mathsf{dom}} & \varphi[x^{\sigma}, x^{\tau}] & \xrightarrow{\mathsf{cod}} & x^{\sigma}_{A} \parallel x^{\tau} \\ {\scriptstyle \theta^{\sigma} \parallel \theta^{\tau}_{C}} \downarrow & & & & \downarrow {\scriptstyle \theta^{\sigma}_{A} \parallel \theta^{\tau}} \\ y^{\sigma} \parallel y^{\tau}_{C} & \xleftarrow[\mathsf{dom}]{} & \varphi[y^{\sigma}, y^{\tau}] & \xrightarrow[\mathsf{cod}]{} & y^{\sigma}_{A} \parallel y^{\tau} \end{array}$$

and seeking a contradiction, assume the induced $\psi : \varphi[x^{\sigma}, x^{\tau}] \simeq \varphi[y^{\sigma}, y^{\tau}]$ is not an identity. By *restriction*, we may assume *w.l.o.g.* that there is exactly one non-identity $((m, m'), (n, n')) \in \psi$. Several cases arise. If $m = (2, c)$, $n = (2, t)$ with $m' = (2, c')$ and $n' = (2, t')$, then $\partial_{\tau} \theta^{\tau} = \theta_{B} \parallel \theta^{\tau}_{C}$ with $\theta_{B} \in \mathcal{S}_{-}(B)$ as an identity symmetry and $\theta^{\tau}_{C} \in \mathcal{S}_{+}(C)$ by hypothesis, so θ^{τ} is an identity as τ is *thin*, contradiction. The case $m = (1, s), n = (1, a), m' = (1, s')$ and $n' = (1, a')$ is symmetric. The last case has $m = (1, s), n = (2, t), m' = (1, s')$ and $n' = (2, t')$. If $\mathrm{pol}_{\sigma}(s) = \mathrm{pol}_{\sigma}(s') = +$, then

$$\partial_{\sigma} \theta^{\sigma} = \theta^{\sigma}_{A} \parallel \theta^{\sigma}_{B} = \mathrm{id}_{x^{\sigma}_{A}} \parallel (\mathrm{id}_{x^{\sigma}_{B}} \uplus \{(b, b')\})$$

with $\partial_{\sigma}(s) = (2, b)$ and $\partial_{\sigma}(s') = (2, b')$. Then, $\mathrm{pol}_{B}(b) = \mathrm{pol}_{B}(b') = +$, so $\theta^{\sigma}_{B} \in \mathcal{S}_{+}(B)$ by $+$*-receptivity*. Of course, $\theta^{\sigma}_{A} \in \mathcal{S}_{-}(A)$ by hypothesis, so θ^{σ} is an identity by *thin*, contradiction. The final case, with $\mathrm{pol}_{\tau}(t) = \mathrm{pol}_{\tau}(t') = +$, is symmetric.

Uniqueness. Immediate from the order-isos, by Lemma 7.1.9. □

Here, an isomorphism of interactions simply means an iso of ess commuting with the display maps on the nose. We may now give the concrete definition of composition.

7.3.2 Composition with Symmetry

To defined composition, we simply add symmetry to Definition 6.2.16:

Definition 7.3.7. *The **composition** of $\boldsymbol{\sigma} : A \vdash B$ and $\boldsymbol{\tau} : B \vdash C$ comprises the components of Definition 6.2.16, enriched with a set of symmetries, defined as:*

$$\theta : x \cong_{\tau\odot\sigma} y \quad \Leftrightarrow \quad \exists \theta \subseteq \theta' : x' \cong_{\tau\circledast\sigma} y' \,.$$

We have yet to show that composition yields a (pre)strategy; and first of all, while the axioms of event structures are immediate, it is not obvious at all that $\mathcal{S}(\boldsymbol{\tau} \odot \boldsymbol{\sigma})$ is an isomorphism family: we need a tighter characterization of symmetries:

Lemma 7.3.8. *Consider $x, y \in \mathscr{C}(\boldsymbol{\tau} \odot \boldsymbol{\sigma})$, and a symmetry $\theta : x \cong_{\tau\odot\sigma} y$.*
There exists a unique *$[\theta]_{\tau\circledast\sigma}$, the **interaction witness** of θ, s.t. $\theta \subseteq [\theta]_{\tau\circledast\sigma}$, and:*

$$[\theta]_{\tau\circledast\sigma} : [x]_{\tau\circledast\sigma} \cong_{\tau\circledast\sigma} [y]_{\tau\circledast\sigma} \,.$$

Proof. Existence. By definition, there is $\theta \subseteq \theta' : x' \cong_{\tau\circledast\sigma} y'$. But $[x]_{\tau\circledast\sigma} \subseteq x'$, so by *restriction* we have $\theta'' : [x]_{\tau\circledast\sigma} \cong_{\tau\circledast\sigma} y''$ – note θ'' still contains θ as $\mathsf{dom}(\theta) = x \subseteq [x]_{\tau\circledast\sigma}$. Now, by Lemma 7.1.6 θ'' is an order-isomorphism. From this, it is easy to show that maximal events of y'' are in y, *i.e.* $y'' = [y]_{\tau\circledast\sigma}$.

Uniqueness. Assume $\vartheta, \vartheta' : [x]_{\tau\circledast\sigma} \cong_{\tau\circledast\sigma} [y]_{\tau\circledast\sigma}$ such that $\theta \subseteq \vartheta, \vartheta'$. Then

$$\partial_{\tau\circledast\sigma}\vartheta = \theta_A \parallel \vartheta_B \parallel \theta_C \qquad\qquad \partial_{\tau\circledast\sigma}\vartheta' = \theta_A \parallel \vartheta'_B \parallel \theta_C$$

only differ on B as $\theta \subseteq \vartheta, \vartheta'$ only adding (pairs of) synchronized events. But then,

$$\partial_{\tau\circledast\sigma}(\vartheta' \circ \vartheta^{-1}) = \mathrm{id}_{[y]_{\tau\circledast\sigma}} \parallel (\vartheta'_B \circ \vartheta_B^{-1}) \parallel \mathrm{id}_{[x]_{\tau\circledast\sigma}}$$

so $\vartheta' \circ \vartheta^{-1}$ is an identity since $\boldsymbol{\tau} \circledast \boldsymbol{\sigma}$ is *thin*; therefore $\vartheta = \vartheta'$. □

We can now conclude that the composition yields an event structure with symmetry:

Proposition 7.3.9. *We have $\boldsymbol{\tau} \odot \boldsymbol{\sigma} = (|\boldsymbol{\tau} \odot \boldsymbol{\sigma}|, \leq_{\tau\odot\sigma}, \#_{\tau\odot\sigma}, \mathcal{S}(\boldsymbol{\tau} \odot \boldsymbol{\sigma}))$, an ess.*
Moreover, we have order-isomorphisms commuting with dom *and* cod*:*

$$\begin{array}{rcl} (- \odot -) & : & \{(x^\tau, x^\sigma) \in \mathscr{C}(\boldsymbol{\tau}) \times \mathscr{C}(\boldsymbol{\sigma}) \mid x^\sigma, x^\tau \text{ minimal } \textit{caus. comp.}\} \;\simeq\; \mathscr{C}(\boldsymbol{\tau} \odot \boldsymbol{\sigma}) \\ (- \odot -) & : & \{(\theta^\tau, \theta^\sigma) \in \mathcal{S}(\boldsymbol{\tau}) \times \mathcal{S}(\boldsymbol{\sigma}) \mid \theta^\sigma, \theta^\tau \text{ minimal } \textit{caus. comp.}\} \;\simeq\; \mathcal{S}(\boldsymbol{\tau} \odot \boldsymbol{\sigma}) \,, \end{array}$$

*where $\theta^\sigma, \theta^\tau$ **minimal** means $\mathsf{dom}(\theta^\sigma), \mathsf{dom}(\theta^\tau)$ (or $\mathsf{cod}(\theta^\sigma), \mathsf{cod}(\theta^\tau)$) minimal.*

Proof. The definition of the event structure and order-isomorphism on configurations are developed in Proposition 6.2.18, which we only need to extend with symmetries. The conditions for isomorphism families follow directly from those for $\mathcal{S}(\boldsymbol{\tau} \circledast \boldsymbol{\sigma})$ in combination with Lemma 7.3.8. For the order-isomorphism on symmetries, first we note that the isomorphism of Proposition 7.3.6 refine to an order-isomorphism:

$$(- \circledast -) \quad : \quad \{(\theta^\tau, \theta^\sigma) \in \mathcal{S}(\boldsymbol{\tau}) \times \mathcal{S}(\boldsymbol{\sigma}) \mid \theta^\sigma, \theta^\tau \text{ min. caus. comp.}\} \quad \simeq \quad \mathcal{S}^v(\boldsymbol{\tau} \circledast \boldsymbol{\sigma})$$

where $\mathcal{S}^v(\boldsymbol{\tau} \circledast \boldsymbol{\sigma})$ refers to symmetries whose maximal (pairs of) events are *visible*. This is clear by Lemma 6.2.12. Next, we have an order-isomorphism formed with:

$$\begin{array}{rcl} \mathcal{S}^v(\boldsymbol{\tau} \circledast \boldsymbol{\sigma}) & \to & \mathcal{S}(\boldsymbol{\tau} \odot \boldsymbol{\sigma}) \\ \theta & \mapsto & \theta \cap V^2 \end{array} \qquad \begin{array}{rcl} \mathcal{S}(\boldsymbol{\tau} \odot \boldsymbol{\sigma}) & \to & \mathcal{S}^v(\boldsymbol{\tau} \circledast \boldsymbol{\sigma}) \\ \theta & \mapsto & [\theta]_{\boldsymbol{\tau} \circledast \boldsymbol{\sigma}} \end{array}$$

where V is the set of visible events – it is easy that this defines an order-iso. □

This defines the composition of $\boldsymbol{\sigma}$ and $\boldsymbol{\tau}$ as an ess, to which we may adjoin a display map specified via Lemma 7.1.9 by completing $\partial_{\boldsymbol{\tau} \odot \boldsymbol{\sigma}}$ with $\partial_{\boldsymbol{\tau} \odot \boldsymbol{\sigma}}(\theta^{\boldsymbol{\tau}} \odot \theta^{\boldsymbol{\sigma}}) = \theta^{\boldsymbol{\sigma}}_A \vdash \theta^{\boldsymbol{\tau}}_C$.

Preservation of prestrategies. We still have not accounted for a few of the additional conditions on (pre)strategies. Before wrapping up the composition of strategies, it is sometimes technically convenient to be able to compose *prestrategies*. This holds only modulo an additional condition on $\boldsymbol{\sigma}$ and $\boldsymbol{\tau}$, called *span-courtesy*:

Definition 7.3.10. *Consider $\boldsymbol{\sigma} : A \vdash B$. We define the following condition:*

span-courtesy: *for all $s_1 \rightarrowtriangle_{\boldsymbol{\sigma}} s_2$, if $\mathrm{pol}_{\boldsymbol{\sigma}}(s_1) = +$ or $\mathrm{pol}_{\boldsymbol{\sigma}}(s_2) = -$, then writing $\partial_{\boldsymbol{\sigma}}(s_i) = (k_i, c_i)$, we have $k_1 = k_2$.*

This tames failures of courtesy, expressing that they cannot span accross the components A, B – without it, composition of prestrategies may fail $\sim$-receptivity.

Now, we have the ingredients to prove the following defining property:

Proposition 7.3.11. *Take $\boldsymbol{\sigma} : A \vdash B$ and $\boldsymbol{\tau} : B \vdash C$ span-courteous prestrategies. Then, there is a prestrategy $\boldsymbol{\tau} \odot \boldsymbol{\sigma}$, unique up to strong iso, s.t. there are order-isos:*

$$\begin{array}{rcl} (- \odot -) & : & \{(x^{\boldsymbol{\tau}}, x^{\boldsymbol{\sigma}}) \in \mathscr{C}(\boldsymbol{\tau}) \times \mathscr{C}(\boldsymbol{\sigma}) \mid x^{\boldsymbol{\sigma}}, x^{\boldsymbol{\tau}} \textit{ min. caus. comp.}\} \simeq \mathscr{C}(\boldsymbol{\tau} \odot \boldsymbol{\sigma}) \\ (- \odot -) & : & \{(\theta^{\boldsymbol{\tau}}, \theta^{\boldsymbol{\sigma}}) \in \mathcal{S}(\boldsymbol{\tau}) \times \mathcal{S}(\boldsymbol{\sigma}) \mid \theta^{\boldsymbol{\sigma}}, \theta^{\boldsymbol{\tau}} \textit{ min. caus. comp.}\} \simeq \mathcal{S}(\boldsymbol{\tau} \odot \boldsymbol{\sigma}) \end{array}$$

commuting with dom *and* cod*; such that for $\theta^{\boldsymbol{\sigma}} \in \mathcal{S}(\boldsymbol{\sigma}), \theta^{\boldsymbol{\tau}} \in \mathcal{S}(\boldsymbol{\tau})$ minimal causally compatible, $\partial_{\boldsymbol{\tau} \odot \boldsymbol{\sigma}}(\theta^{\boldsymbol{\tau}} \odot \theta^{\boldsymbol{\sigma}}) = \theta^{\boldsymbol{\sigma}}_A \parallel \theta^{\boldsymbol{\tau}}_C$. Moreover, $\boldsymbol{\tau} \odot \boldsymbol{\sigma}$ is span-courteous.*

Proof. Existence. The ess $\boldsymbol{\tau} \odot \boldsymbol{\sigma}$ is defined in Proposition 7.3.9, and the display map as above. *Thin* is immediate from *thin* for interactions, and Proposition 7.3.6.

For $\sim$-*receptive*, consider $\theta^{\boldsymbol{\tau}} \odot \theta^{\boldsymbol{\sigma}} : x^{\boldsymbol{\tau}} \odot x^{\boldsymbol{\sigma}} \cong_{\boldsymbol{\tau} \odot \boldsymbol{\sigma}} y^{\boldsymbol{\tau}} \odot y^{\boldsymbol{\sigma}}$ with extensions

$$x^{\boldsymbol{\tau}} \odot x^{\boldsymbol{\sigma}} \vdash_{\boldsymbol{\tau} \odot \boldsymbol{\sigma}} p_1^-, \qquad \partial_{\boldsymbol{\tau} \odot \boldsymbol{\sigma}}(\theta^{\boldsymbol{\tau}} \odot \theta^{\boldsymbol{\sigma}}) \vdash_{\boldsymbol{\tau} \odot \boldsymbol{\sigma}} (\partial_{\boldsymbol{\tau} \odot \boldsymbol{\sigma}}(p_1^-), (2, c_2^-)),$$

assuming *w.l.o.g.* that $\partial_{\boldsymbol{\tau} \odot \boldsymbol{\sigma}}(p_1) = (2, c_1)$ occurs in C, witnessed in the interaction by

$$x^{\boldsymbol{\tau}} \circledast x^{\boldsymbol{\sigma}} \subseteq y^{\boldsymbol{\tau}} \circledast y^{\boldsymbol{\sigma}} \vdash_{\boldsymbol{\tau} \circledast \boldsymbol{\sigma}} p_1^-,$$

where the inclusion only adds synchronized events, which *w.l.o.g.* we may assume to be causal dependencies of the extension p_1. In fact, then, $x^{\boldsymbol{\tau}} \circledast x^{\boldsymbol{\sigma}} = y^{\boldsymbol{\tau}} \circledast y^{\boldsymbol{\sigma}}$. Indeed, taking $p_0 \in y^{\boldsymbol{\tau}} \circledast y^{\boldsymbol{\sigma}}$ such that $p_0 \rightarrowtriangle_{\boldsymbol{\tau} \circledast \boldsymbol{\sigma}} p_1$, by Lemma 6.2.15 this entails that $(p_0)_{\boldsymbol{\tau}}$

is defined and $(p_0)_{\boldsymbol{\tau}} \rightarrowtriangle_{\boldsymbol{\tau}} (p_1)_{\boldsymbol{\tau}}$. But as $\boldsymbol{\tau}$ is *span-courteous*, $\partial_{\boldsymbol{\tau}}(p_0)_{\boldsymbol{\tau}} = (2, c_0)$, so p_0 is visible. Thus $p_0 \in x^{\boldsymbol{\tau}} \circledast x^{\boldsymbol{\sigma}}$ necessarily. So in fact, we have:

$$x^{\boldsymbol{\tau}} \circledast x^{\boldsymbol{\sigma}} \vdash_{\boldsymbol{\tau}\circledast\boldsymbol{\sigma}} p_1^- , \qquad \partial_{\boldsymbol{\tau}\circledast\boldsymbol{\sigma}}(\theta^{\boldsymbol{\tau}} \circledast \theta^{\boldsymbol{\sigma}}) \vdash_{\boldsymbol{\tau}\circledast\boldsymbol{\sigma}} ((3, c_1), (3, c_2))$$

which projects to $x^{\boldsymbol{\tau}} \vdash_{\boldsymbol{\tau}} (p_1)_{\boldsymbol{\tau}}$ and $\partial_{\boldsymbol{\tau}}(\theta_{\boldsymbol{\tau}}) \vdash_{\boldsymbol{\tau}} (\partial_{\boldsymbol{\tau}}((p_1)_{\boldsymbol{\tau}}), (2, c_2))$, and by *∼-receptive* for $\boldsymbol{\tau}$, there is a unique $\theta^{\boldsymbol{\tau}} \vdash_{\boldsymbol{\tau}} ((p_1)_{\boldsymbol{\tau}}, t_2)$ such that $\partial_{\boldsymbol{\tau}}(t_2) = c_2$. Now, by Proposition 7.3.9 we have $y^{\boldsymbol{\tau}} \odot y^{\boldsymbol{\sigma}} \multimap (y^{\boldsymbol{\tau}} \uplus \{t_2\}) \odot y^{\boldsymbol{\sigma}}$ – those being easily seen minimal causally compatible – write $p_2 \in \boldsymbol{\tau} \odot \boldsymbol{\sigma}$ the added move. Likewise, we have

$$\theta^{\boldsymbol{\tau}} \odot \theta^{\boldsymbol{\sigma}} \overset{(p_1,p_2)}{\multimap} (\theta^{\boldsymbol{\tau}} \uplus \{((p_1)_{\boldsymbol{\tau}}, t_2)\}) \odot \theta^{\boldsymbol{\sigma}}$$

which fits the requirements; uniqueness is direct from uniqueness of t_2.

For *span-courteous*, take $p_1 \rightarrowtriangle_{\boldsymbol{\tau}\odot\boldsymbol{\sigma}} p_2$ s.t. $\mathrm{pol}(p_1) = +$ or $\mathrm{pol}(p_2) = -$, say first $\mathrm{pol}(p_2) = -$, and *w.l.o.g.* p_2 occurs in C, *i.e.* $\partial_{\boldsymbol{\tau}\odot\boldsymbol{\sigma}}(p_2) = (2, c_2^-)$. By Lemma 6.2.21,

$$p_1 \rightarrowtriangle_{\boldsymbol{\tau}\circledast\boldsymbol{\sigma}} q_1 \rightarrowtriangle_{\boldsymbol{\tau}\circledast\boldsymbol{\sigma}} \cdots \rightarrowtriangle_{\boldsymbol{\tau}\circledast\boldsymbol{\sigma}} q_n \rightarrowtriangle_{\boldsymbol{\tau}\circledast\boldsymbol{\sigma}} p_2$$

with all q_i synchronized, if any. But by Lemma 6.2.15, $(q_n)_{\boldsymbol{\tau}}$ exists and $(q_n)_{\boldsymbol{\tau}} \rightarrowtriangle_{\boldsymbol{\tau}} (p_2)_{\boldsymbol{\tau}}$. But as $\boldsymbol{\tau}$ is *span-courteous*, this implies $\partial_{\boldsymbol{\tau}}((q_n)_{\boldsymbol{\tau}}) = (2, c)$, so is q_n visible – this entails $p_1 = q_n$, so $\partial_{\boldsymbol{\tau}\odot\boldsymbol{\sigma}} = (2, c)$ as required. The case where $\mathrm{pol}(p_1) = +$ is analogous.

Uniqueness up to strong iso is straightforward by Lemma 7.1.9. □

Without *span-courtesy*, *∼-receptivity* may fail to be preserved under composition – a counter-example is not terribly hard to find; we omit it for the economy of presentation.

Composition of strategies. We may finally prove our main proposition:

Proposition 7.3.1. *Consider A, B, C tcgs, and $\boldsymbol{\sigma} : A \vdash B$ and $\boldsymbol{\tau} : B \vdash C$ strategies. Then there is a strategy $\boldsymbol{\tau} \odot \boldsymbol{\sigma} : A \vdash C$, unique up to strong iso, such that there are*

$$\begin{array}{lcl} (- \odot -) & : & \{(x^{\boldsymbol{\tau}}, x^{\boldsymbol{\sigma}}) \in \mathscr{C}^+(\boldsymbol{\tau}) \times \mathscr{C}^+(\boldsymbol{\sigma}) \mid \textit{causally compatible}\} \simeq \mathscr{C}^+(\boldsymbol{\tau} \odot \boldsymbol{\sigma}) \\ (- \odot -) & : & \{(\theta^{\boldsymbol{\tau}}, \theta^{\boldsymbol{\sigma}}) \in \mathcal{S}^+(\boldsymbol{\tau}) \times \mathcal{S}^+(\boldsymbol{\sigma}) \mid \textit{causally compatible}\} \simeq \mathcal{S}^+(\boldsymbol{\tau} \odot \boldsymbol{\sigma}) \end{array}$$

order-isomorphisms commuting with dom, cod*, and such that we have*

$$\partial_{\boldsymbol{\tau}\odot\boldsymbol{\sigma}}(\theta^{\boldsymbol{\tau}} \odot \theta^{\boldsymbol{\sigma}}) = \theta^{\boldsymbol{\sigma}}_A \parallel \theta^{\boldsymbol{\tau}}_C ,$$

for all $\theta^{\boldsymbol{\sigma}} \in \mathcal{S}^+(\boldsymbol{\sigma})$ and $\theta^{\boldsymbol{\tau}} \in \mathcal{S}^+(\boldsymbol{\tau})$ causally compatible.

Proof. Existence. Being *courteous*, strategies are in particular *span-courteous*, which applies to show that $\boldsymbol{\tau} \odot \boldsymbol{\sigma}$ is a prestrategy. It is a strategy by Proposition 6.3.1, as the additional conditions do not involve symmetry. The order-isomorphisms follow from Proposition 7.3.11 specialized to +-covered configurations – Lemmas 6.3.5, 6.3.6 and 6.3.7 directly extend to symmetries as they are order-isomorphisms.

Uniqueness. By Corollary 7.2.13. □

7.3.3 Copycat

Now, we introduce the *copycat* thin strategy. As for composition, copycat will simply be the copycat strategy of Chapter 6, adequately enriched with symmetry.

Recall from Lemma 6.4.2 that for a plain game A, the configurations of $\mathbf{cc}_A$ are those $x \vdash y \in \mathscr{C}(A \vdash A)$ such that $x \supseteq^+ x \cap y \subseteq^- y$; here the intuition is that y comprises positive events imported from x, negative events already propagated to x, but also negative events that have yet to be forwarded to x – and the situation is symmetric for x.

So what should be a symmetry

$$\theta : x_1 \vdash x_2 \cong_{\mathbf{cc}_A} y_1 \vdash y_2 \,?$$

As it should display to a symmetry on $A \vdash A$, we know it should have the form $\theta_1 \vdash \theta_2$ for $\theta_1 : x_1 \cong_A y_1$ and $\theta_2 : x_2 \cong_A y_2$. But that is not all: additionally, θ_1 and θ_2 must *agree* on events that have already been propagated. We formalize this as:

Definition 7.3.12. *For any tcg A, $\mathbf{cc}_A : A \vdash A$ is defined with Definition 6.4.1, with:*

$$\begin{array}{lrcl} \text{symmetry:} & \mathscr{S}(\mathbf{cc}_A) & = & \{\theta_1 \vdash \theta_2 \in \mathscr{S}(A \vdash A) \mid \\ & & & \quad \mathsf{dom}(\theta_1 \vdash \theta_2) \in \mathscr{C}(\mathbf{cc}_A),\ \mathsf{cod}(\theta_1 \vdash \theta_2) \in \mathscr{C}(\mathbf{cc}_A), \\ & & & \quad \forall a \in \mathsf{dom}(\theta_1) \cap \mathsf{dom}(\theta_2),\ \theta_1(a) = \theta_2(a)\}\,. \end{array}$$

Note that though we have required $\theta_1(a) = \theta_2(a)$ on any element of the common domain. This looks asymmetric, but it will follow from the next lemma that we could have equivalently asked $\theta_1^{-1}(a) = \theta_2^{-1}(a)$ for any a in the common codomain.

Now we show that this gives a well-formed ess, and characterize its symmetries:

Lemma 7.3.13. *Consider A a tcg. Then, $(|\mathbf{cc}_A|, \leq_A, \#_A, \mathscr{S}(\mathbf{cc}_A))$ is an ess.*
Moreover, $\mathscr{S}(\mathbf{cc}_A) = \{\theta_1 \vdash \theta_2 \mid \theta_1, \theta_2 \in \mathscr{S}(A), \theta_1 \cap \theta_2 \subseteq^+ \theta_1 \;\&\; \theta_1 \cap \theta_2 \subseteq^- \theta_2\}$.

Proof. We first prove the characterization of symmetries. For $\subseteq$, clearly $\theta_1 \cap \theta_2 \subseteq \theta_1$ and take $(a, a') \in \theta_1$ with $\mathrm{pol}_A(a) = -$. Since $\theta_1 \vdash \theta_2 \in \mathscr{S}(\mathbf{cc}_A)$, in particular $\mathsf{dom}(\theta_1 \vdash \theta_2) \in \mathscr{C}(\mathbf{cc}_A)$. So by Lemma 6.4.2, $a \in \mathsf{dom}(\theta_2)$. Now, by definition of $\mathscr{S}(\mathbf{cc}_A)$ again, $\theta_1(a) = \theta_2(a)$, so $(a, a') \in \theta_1 \cap \theta_2$. The proof that $\theta_1 \cap \theta_2 \subseteq^- \theta_2$ is symmetric.

For $\supseteq$, consider $a \in \mathsf{dom}(\theta_1) \cap \mathsf{dom}(\theta_2)$ and *w.l.o.g.* assume $\mathrm{pol}_A(a) = +$; the other case is symmetric. Since $a \in \mathsf{dom}(\theta_2)$, there is $(a, a') \in \theta_2$. But since $\theta_1 \cap \theta_2 \subseteq^- \theta_2$, it follows that $(a, a') \in \theta_1$ as well, so $\theta_1(a) = \theta_2(a)$ as required. Next, we show $\mathsf{dom}(\theta_1 \vdash \theta_2) \in \mathscr{C}(\mathbf{cc}_A)$ – let us write $\theta_1 : x_1 \cong_A y_1$ and $\theta_2 : x_2 \cong_A y_2$ and reason via Lemma 6.4.2. Consider $a^+ \in x_2$; we must show $a^+ \in x_1$ as well. But there is some $(a^+, b^+) \in \theta_2$, so by hypothesis $(a^+, b^+) \in \theta_1$ as well, so $a^+ \in x_1$. Symmetrically, any $a^- \in x_1$ must be in x_2; so $x_1 \vdash x_2 \in \mathscr{C}(\mathbf{cc}_A)$. Symmetrically, $y_1 \vdash y_2 \in \mathscr{C}(\mathbf{cc}_A)$.

Next come the axioms for isomorphism families. *Groupoid* and *restriction* follow directly from the characterization of symmetries and Lemma 6.4.2. However, *extension* is subtle – to keep the development focused we postpone it to Lemma 7.3.17. □

Extension is really subtle. Indeed, the first significant surprise when starting the design of concurrent games with symmetry, was that copycat failed *extension* with games

plain ess with polarities (see Section A.4 in [Castellan et al., 2019]); this is a large part of what first steered us towards saturation in [Castellan et al., 2014], and of what later prompted us to look for what would eventually become the definition of tcgs. Here we opt to keep going with the development of copycat which is largely independent of the proof of *extension*, and come back to extension later on.

We mention the property on symmetries reflecting Lemma 6.4.4:

Lemma 7.3.14. *Consider A a tcg. Then, $\mathcal{S}^+(\mathbf{cc}_A) = \{\theta \vdash \theta \mid \theta \in \mathcal{S}(A)\}$.*
If $\theta \in \mathcal{S}(A)$, we write $\mathbf{cc}_\theta = \theta \vdash \theta \in \mathcal{S}^+(\mathbf{cc}_A)$ for this $+$-covered symmetry.

Proof. Immediate from Lemmas 7.3.13 and 6.4.4. □

Finally, we show that copycat indeed is a well-formed strategy.

Proposition 7.3.15. *For any tcg A, we have $\mathbf{cc}_A : A \vdash A$ a strategy.*

Proof. By Lemma 7.3.13, $\mathbf{cc}_A$ is an ess. We first check the conditions for a prestrategy (*i.e.* Definition 7.1.19). *Rule-abiding*, *locally injective*, *$\sim$-preserving*, and *$\sim$-receptive* are direct. For *thin*, consider $\theta_1 \vdash \theta_2 : x_1 \vdash x_2 \cong_{\mathbf{cc}_A} y_1 \vdash y_2$ such that $\theta_1 \in \mathcal{S}_-(A)$ and $\theta_2 \in \mathcal{S}_+(A)$. By *restriction*, this means that $\theta_1 \cap \theta_1 \in \mathcal{S}_-(A) \cap \mathcal{S}_+(A)$. By *orthogonality*, this means that $\theta_1 \cap \theta_2 = \mathrm{id}_x$ for some $x \in \mathscr{C}(A)$. But then, by Lemma 7.3.13, $\mathrm{id}_x \subseteq^- \theta_2$, so by *$-$-receptive*, $\theta_2 \in \mathcal{S}_-(A)$. So again, $\theta_2 \in \mathcal{S}_-(A) \cap \mathcal{S}_+(A)$ must be an identity by *orthogonality* – the symmetric argument shows that θ_1 is an identity.

Finally, we show the conditions of Definition 7.1.20: *courteous* follows from Lemma 6.4.3, while *receptive* is direct from Lemma 6.4.2. □

In the sequel, we shall mostly reason on copycat via the characterizations of its $+$-covered configurations and symmetries in Lemmas 6.4.2 and 7.3.13.

The extension property. Now, we set towards completing the proof of Lemma 7.3.13 and proving the *extension* property for $\mathcal{S}(\mathbf{cc}_A)$. This is a bit technical, but this development will not be referred to later on in this monograph, so skipping to Section 7.4 should not impede understanding of the rest of the text.

First, we find it interesting to start with a counter-example, to see where is the tension and how *close* extension is to fail. Consider $!(\ominus\oplus)$ obtained via the clauses of Definition 7.1.26, ignoring the negativity assumption and the positive and negative symmetries (recall that we noted below Definition 7.1.26 that those do *not* form a tcg). We may define $\mathbf{cc}_{!(\ominus\oplus)}$ exactly as in Definition 7.3.12, but it does *not* form an event structure with symmetry, as it fails *extension*! To see the problem, consider the symmetry

$$\begin{array}{ccc} \oplus_1 & \vdash & \ominus_1 \\ \theta_1 \cong & & \cong \theta_2 \\ \oplus_1 & \vdash & \ominus_2 \end{array}$$

linking configurations of $\mathbf{cc}_{!(\ominus\oplus)}$, where no event has been forwarded. The issue is that θ_1 and θ_2 have made irreconciliable choices regarding copy indices. It causes no issue

yet as they live in separated components, but the upper configuration may move to

$$\ominus_1 \oplus_1 \qquad \vdash \qquad \ominus_1$$

which, by *extension*, should trigger a corresponding extension of the bottom configuration, and of the symmetry. But the bottom configuration may only extend to $\ominus_2 \oplus_1 \vdash \ominus_2$ (by propagating to the left hand side the event $\ominus_2$ from the right hand side), and $\ominus_1 \oplus_1 \not\cong_{!(\ominus\oplus)} \ominus_1 \oplus_2$, because a symmetry in $!(\ominus\oplus)$ must follow a permutation of copy indices! Thus, extension fails. Fortunately for tcgs this cannot happen, due to:

Lemma 7.3.16. *Consider A a tcg, $\theta, \theta_1, \theta_2 \in \mathcal{S}(A)$ such that $\theta \subseteq^+ \theta_1$ and $\theta \subseteq^- \theta_2$. If $\mathsf{dom}(\theta_1)$ and $\mathsf{dom}(\theta_2)$ are compatible, then so are θ_1 and θ_2.*

Proof. First, we prove this property for the *positive* symmetries, thus assume we have $\theta, \theta_1, \theta_2 \in \mathcal{S}_+(A)$ such that $\theta \subseteq^+ \theta_1$ and $\theta \subseteq^- \theta_2$. By *extension* for $\mathcal{S}_+(A)$, we have

$$\begin{array}{ccc} \theta_1' : x_1 \cup x_2 \cong_A^+ y_1' & & \theta_2' : x_1 \cup x_2 \cong_A^+ y_2' \\ \big| \cup \big| & & \big| \cup \big| \\ \theta_1 : x_1 \cong_A^+ y_1 & & \theta_2 : x_2 \cong_A^+ y_2 \\ & \searrow \quad \swarrow & \\ & \theta : x \cong_A^+ y & \end{array}$$

so by *groupoid*, we can form $\varphi = \theta_2' \circ {\theta_1'}^{-1} : y_1' \cong_A^+ y_2'$. As both θ_1' and θ_2' contain θ, it follows that $\mathrm{id}_y \subseteq \varphi$. Now, by *restriction*, consider $\psi : x_1' \cong_A^+ y_2$ obtained by restricting φ to y_2 on the right. Since $y \subseteq y_2$, we still have $\mathrm{id}_y \subseteq \psi$. But actually that inclusion is negative, which – as $\mathrm{id}_y \in \mathcal{S}_-(A)$ – entails that $\psi \in \mathcal{S}_-(A)$ as well. Thus by *orthogonality*, $x_1' = y_2$ and $\psi = \mathrm{id}_{y_2}$. Summing up, if $(a, a') \in \theta_2$, then $(a, a') \in \theta_2'$, and there is $(a'', a') \in \varphi$ for some $(a'', a) \in {\theta_1'}^{-1}$. But then $(a'', a') \in \psi$ also, hence $a'' = a'$ and $(a, a') \in \theta_1'$; so we have proved $\theta_2 \subseteq \theta_1'$, establishing compatibility of θ_1 and θ_2. Symmetrically, the same property holds for $\mathcal{S}_-(A)$.

Now, from this we must conclude that the property holds for $\mathcal{S}(A)$, so take $\theta, \theta_1, \theta_2 \in \mathcal{S}(A)$ such that $\theta \subseteq^+ \theta_1$ and $\theta \subseteq^+ \theta_2$. By Lemma 7.1.18, we have unique factorizations

$$\theta = \theta^+ \circ \theta^-, \qquad \theta_1 = \theta_1^+ \circ \theta_1^-, \qquad \theta_2 = \theta_2^+ \circ \theta_2^-.$$

By *uniqueness* in Lemma 7.1.18, it is immediate that this factorization is monotone, *i.e.* $\theta^+ \subseteq \theta_1^+$ and $\theta^+ \subseteq \theta_2^+$, so by the reasoning above we have θ_1^+ and θ_2^+ compatible in $\mathcal{S}_+(A)$, *i.e.* $\theta_1^+ \cup \theta_2^+ \in \mathcal{S}_+(A)$. Likewise, $\theta_1^- \cup \theta_2^- \in \mathcal{S}_-(A)$. Now it is immediate that $\theta_1^- \cup \theta_2^-$ and θ_1^+ and θ_2^+ are composable, and that we have

$$(\theta_1^+ \cup \theta_2^+) \circ (\theta_1^- \cup \theta_2^-) = (\theta_1^+ \circ \theta_1^-) \cup (\theta_2^+ \circ \theta_2^-) = \theta_1 \cup \theta_2$$

which must hence be a valid symmetry, concluding that θ_1 and θ_2 are compatible. □

With the key lemma above, it is straightforward to finally deduce:

Lemma 7.3.17. *Consider A a tcg. Then, $\mathcal{S}(\mathfrak{C}_A)$ satisfies* extension.

Proof. Consider a symmetry on copycat, which may be written as

$$\begin{array}{ccc} x_1 & \vdash & x_2 \\ {\scriptstyle\theta_1}\downarrow & & \downarrow{\scriptstyle\theta_2} \\ y_1 & \vdash & y_2 \end{array}$$

and consider an extension of $x_1 \vdash x_2$. Without loss of generality, we can consider that only x_2 is extended, and with moves of the same polarity – this leaves two cases, for $x_2 \subseteq^- x_2'$ and $x_2 \subseteq^+ x_2'$. In the former case, then by *extension* we have $\theta_2 \subseteq \theta_2' : x_2' \cong_A y_2'$, and $\theta_1 \subseteq^+ \theta_1 \cap \theta_2 \subseteq^- \theta_2 \subseteq^+ \theta_2'$, so $\theta_1 \vdash \theta_2' \in \mathcal{S}(\mathbb{c}_A)$ by Lemma 7.3.13.

In the latter case, writing $x_1' = x_1 \cap x_2'$, and considering the restriction θ_1' of θ_1 to x_1',

$$\theta_1' \supseteq^+ \theta_1 \cap \theta_2 \subseteq^- \theta_2 \,,$$

so that $\theta_2' = \theta_1' \cup \theta_2 \in \mathcal{S}(A)$ by Lemma 7.3.16; and $\theta_1 \vdash \theta_2'$ is the desired extension. □

At last, this concludes the construction of the thin copycat strategy.

7.4 The Bicategory TCG

We have most of the ingredients for our bicategory **TCG**: a notion of objects (thin concurrent games), morphisms (thin strategies), 2-cells (positive morphisms). We have the data for identities (copycat strategies) and composition.

Structural 2-cells. We must still extend from **CG** the 2-cells for associativity and unity laws, which is direct from the developments above. For associators, we have:

Proposition 7.4.1. *Consider thin prestrategies $\boldsymbol{\sigma} : A \vdash B, \boldsymbol{\tau} : B \vdash C$ and $\boldsymbol{\delta} : C \vdash D$. Then, there is a strong isomorphism of strategies, the **associator***

$$a_{\sigma,\tau,\delta} : (\boldsymbol{\delta} \odot \boldsymbol{\tau}) \odot \boldsymbol{\sigma} \cong \boldsymbol{\delta} \odot (\boldsymbol{\tau} \odot \boldsymbol{\sigma}) \,,$$

such that for all $(x^{\delta} \odot x^{\tau}) \odot x^{\sigma} \in \mathcal{C}((\boldsymbol{\delta} \odot \boldsymbol{\tau}) \odot \boldsymbol{\sigma})$ and $(\theta^{\delta} \odot \theta^{\tau}) \odot \theta^{\sigma} \in \mathcal{S}((\boldsymbol{\delta} \odot \boldsymbol{\tau}) \odot \boldsymbol{\sigma})$,

$$\begin{aligned} a_{\sigma,\tau,\delta}((x^{\delta} \odot x^{\tau}) \odot x^{\sigma}) &= x^{\delta} \odot (x^{\tau} \odot x^{\sigma}) \\ a_{\sigma,\tau,\delta}((\theta^{\delta} \odot \theta^{\tau}) \odot \theta^{\sigma}) &= \theta^{\delta} \odot (\theta^{\tau} \odot \theta^{\sigma}) \,. \end{aligned}$$

Proof. To apply Lemma 7.1.9 we show the associator preserves symmetry, which is clear by the description of the symmetries in the composition in Proposition 7.3.9. □

Likewise, the developments above make it easy to define the *unitors*:

Proposition 7.4.2. *Consider $\boldsymbol{\sigma} : A \vdash B$ a thin strategy.*
*Then there are strong isomorphisms, the **unitors**:*

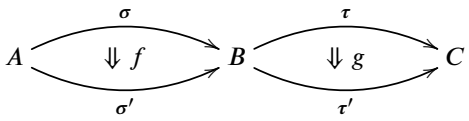
$$l_{\sigma} : \mathbb{c}_B \odot \boldsymbol{\sigma} \cong \boldsymbol{\sigma} \qquad r_{\sigma} : \boldsymbol{\sigma} \odot \mathbb{c}_A \cong \boldsymbol{\sigma}$$

such that for all $x^{\sigma} \in \mathscr{C}^{+}(\sigma)$ *and* $\theta^{\sigma} \in \mathscr{S}^{+}(\sigma)$*,*

$$\begin{array}{rclcrcl} l_{\sigma}(\mathbb{w}_{x^{\sigma}_{B}} \odot x^{\sigma}) & = & x^{\sigma} & & r_{\sigma}(x^{\sigma} \odot \mathbb{w}_{x^{\sigma}_{A}}) & = & x^{\sigma} \\ l_{\sigma}(\mathbb{w}_{\theta^{\sigma}_{B}} \odot \theta^{\sigma}) & = & \theta^{\sigma}, & & r_{\sigma}(\theta^{\sigma} \odot \mathbb{w}_{\theta^{\sigma}_{A}}) & = & \theta^{\sigma}. \end{array}$$

Proof. Obvious by Corollary 7.2.13. □

Horizontal composition. There is still one piece of data missing: the horizontal composition of positive morphisms, letting us assemble two positive morphisms as in



into $g \odot f : \tau \odot \sigma \Rightarrow \tau' \odot \sigma'$. In **CG** this was simple, as we could just set

$$(g \odot f)(x^{\tau} \odot x^{\sigma}) = g(x^{\tau}) \odot f(x^{\sigma})$$

defining a 2-cell in **CG** via Lemma 6.3.4. In **TCG** this would also work, for *strong* morphisms. But in general, for *positive* morphisms, $f(x^{\sigma}) \in \mathscr{C}(\sigma')$ and $g(x^{\tau}) \in \mathscr{C}(\tau')$ have no reason to be matching as we only have $\partial_{\sigma'} \circ f \sim^{+} \partial_{\sigma}$ and $\partial_{\tau'} \circ g \sim^{+} \partial_{\tau}$.

By the definition of positively symmetric maps, for all $x^{\tau} \odot x^{\sigma} \in \mathscr{C}^{+}(\tau \odot \sigma)$, there are (necessarily unique) $\theta^{-}_{A} \in \mathscr{S}_{-}(A), \theta^{+}_{B} \in \mathscr{S}_{+}(B), \theta^{-}_{B} \in \mathscr{S}_{-}(B)$ and $\theta^{+}_{C} \in \mathscr{S}_{+}(C)$ s.t.

$$\begin{array}{ccc} x^{\sigma} & \xrightarrow{\quad f \quad} & f(x^{\sigma}) \\ {\scriptstyle \partial_{\sigma}}\downarrow & & \downarrow{\scriptstyle \partial_{\sigma'}} \\ x^{\sigma}_{A} \vdash x^{\sigma}_{B} & \xrightarrow[\theta^{-}_{A} \vdash \theta^{+}_{B}]{} & f(x^{\sigma})_{A} \vdash f(x^{\sigma})_{B} \end{array} \qquad \begin{array}{ccc} x^{\tau} & \xrightarrow{\quad g \quad} & g(x^{\tau}) \\ {\scriptstyle \partial_{\tau}}\downarrow & & \downarrow{\scriptstyle \partial_{\tau'}} \\ x^{\tau}_{B} \vdash x^{\tau}_{C} & \xrightarrow[\theta^{-}_{B} \vdash \theta^{+}_{C}]{} & g(x^{\tau})_{B} \vdash g(x^{\tau})_{C} \end{array} \tag{7.6}$$

commute in the category of finite sets of bijections (f, g and the display maps being locally injective). Let us write $f[x^{\sigma}]_{A} = \theta^{-}_{A}$, $f[x^{\sigma}]_{B} = \theta^{+}_{B}$, $g[x^{\tau}]_{B} = \theta^{-}_{B}$ and $g[x^{\tau}]_{C} = \theta^{+}_{C}$ to emphasize that those symmetries are part of f and g, and the dependency on x^{σ} and x^{τ}. So, though we lack $f(x^{\sigma})_{B} = g(x^{\tau})_{B}$ as *matching* would require, we do have

$$\theta : f(x^{\sigma})_{B} \cong_{B} g(x^{\tau})_{B}$$

where $\theta = g[x^{\tau}]_{B} \circ f[x^{\sigma}]^{-1}_{B}$, and using that $x^{\sigma}_{B} = x^{\tau}_{B}$: $f(x^{\sigma})$ and $g(x^{\tau})$ are *matching up to symmetry*. The fundamental property we must show, is that such a situation can be massaged to extract configurations of σ' and τ' matching *on the nose*.

This is our final required conceptual tool, *synchronization up to symmetry*:

7.4.1 Synchronization up to Symmetry

Fix $\sigma : A \vdash B$ and $\tau : B \vdash C$ two (pre)strategies.

First, we must specify under which conditions we expect to be able to synchronize two configurations up to symmetry. As without symmetry, there needs to be some causal compatibility condition – accordingly, we first generalize the definition of *causally compatible pairs* (Definition 6.2.2) to account for a mediating symmetry:

Definition 7.4.3. *Consider $x^\sigma \in \mathscr{C}(\sigma), x^\tau \in \mathscr{C}(\tau)$ and $\theta : x^\sigma_B \cong_B x^\tau_B$.*
*We say that the triple $(x^\sigma, \theta, x^\tau)$ is **causally compatible** if the composite bijection*

$$\varphi[x^\sigma, \theta, x^\tau] : x^\sigma \parallel x^\tau_C \overset{\partial_\sigma \| x^\tau_C}{\simeq} x^\sigma_A \parallel x^\sigma_B \parallel x^\tau_C \overset{x^\sigma_A \| \theta \| x^\tau_C}{\simeq} x^\sigma_A \parallel x^\tau_B \parallel x^\tau_C \overset{\partial_\tau^{-1} \| x^\tau_C}{\simeq} x^\sigma_A \parallel x^\tau$$

*is **secured**, in the sense that the relation*

$$(m, n) \lhd (m', n') \qquad \Leftrightarrow \qquad m <_{\sigma \| C} m' \quad \vee \quad n <_{A \| \tau} n'$$

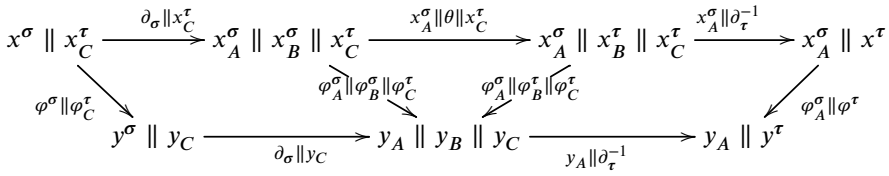
defined on (the graph of) $\varphi[x^\sigma, \theta, x^\tau]$ from the causal constraints of σ and τ, is acyclic.
*In that case, the reflexive transitive closure of $\lhd$ is a partial order $\leq_{\varphi[x^\sigma, \theta, x^\tau]}$, and $(x^\sigma, \theta, x^\tau)$ is **minimal** if the maximal elements of $\varphi[x^\sigma, \theta, x^\tau]$ occur in A or C.*

At the core of **TCG** is the following property, stating that any causally compatible triple can be massaged to produce a unique equivalent causally compatible pair:

Proposition 7.4.4. *Consider $(x^\sigma, \theta, x^\tau)$ a causally compatible triple.*
Then, there are unique $y^\sigma \in \mathscr{C}(\sigma)$ and $y^\tau \in \mathscr{C}(\tau)$ causally compatible, and

$$\varphi^\sigma : x^\sigma \cong_\sigma y^\sigma, \qquad \varphi^\tau : x^\tau \cong_\tau y^\tau,$$

such that we have $\varphi^\sigma_A \in \mathcal{S}_-(A)$, $\varphi^\tau_C \in \mathcal{S}_+(C)$, and the following diagram commutes:



(in particular, the middle triangle commutes – the squares come for free).

Proof. Securedness follows the obvious generalization of Definition 6.2.2.

Existence. For a secured bijection ϕ (top of the diagram) as above, assume adequate $\varphi^\sigma : x^\sigma \cong_\sigma y^\sigma$ and $\varphi^\tau : x^\tau \cong_\tau y^\tau$ are provided by induction hypothesis. Consider:

$$\phi \overset{(m,n)}{\rightarrowtail} \phi'$$

with ϕ' a composite bijection. We first consider the key case, with $m = (1, s)$ for $x^\sigma \vdash_\sigma s^+$, and $n = (2, t)$ for $x^\tau \vdash_\tau t^-$ synchronized. Let us write their displays as

$$\partial_\sigma(x^\sigma \uplus \{s\}) = x^\sigma_A \vdash (x^\sigma_B \uplus \{b_1\}), \qquad \partial_\tau(x^\tau \uplus \{t\}) = (x^\tau_B \uplus \{b_2\}) \vdash x^\tau_C$$

synchronizing through $\theta \uplus \{(b_1, b_2)\} \in \mathcal{S}(B)$.

By *extension* for σ, we may propagate this extension through φ^σ to obtain:

$$\psi^\sigma = \varphi^\sigma \uplus \{(s, s')\} \,:\, x^\sigma \uplus \{s\} \cong_\sigma y^\sigma \uplus \{s'\}$$

displaying to $\partial_\sigma(\psi^\sigma) = \varphi^\sigma_A \vdash (\varphi^\sigma_B \uplus \{(b_1, b')\}) \in \mathcal{S}(A \vdash B)$. But then:

$$\varphi^\tau_B \uplus \{(b_2, b')\} \in \mathcal{S}(B)$$

as it may be obtained as the composition $(\varphi^\sigma_B \uplus \{(b_1, b')\}) \circ (\theta \uplus \{(b_1, b_2)\})^{-1}$ of known symmetries in B. Then $\sim$-receptivity directly applies to provide $\psi^\tau = \varphi^\tau \uplus \{(t, t')\}$ matching this extension, concluding this case – missing verifications are immediate.

The case with s negative and t positive is symmetric. Next, consider the case $m = (2, c)$ and $n = (2, t)$ with $x^\tau \vdash_\tau t^-$, *i.e.* a visible, negative move in C. Its display is

$$\partial_\tau(x^\tau \uplus \{t\}) = x^\tau_B \vdash (x^\tau_C \uplus \{c\})\,,$$

but then *extension* of $\mathcal{S}_+(C)$ applies to provide an extension $\varphi^\tau_C \uplus \{(c, c')\} \in \mathcal{S}_+(C)$. By *$\sim$-receptivity* of τ, there is an (unique) matching extension $\psi^\tau = \varphi^\tau \uplus \{(t, t')\} \in \mathcal{S}(\tau)$, concluding this case – other components are unchanged.

Consider then the case $m = (2, c)$ and $n = (2, t)$ with $x^\tau \vdash_\tau t^+$. In that case,

$$\psi^\tau = \varphi^\tau \uplus \{(t, t')\} \in \mathcal{S}(\tau)$$

is provided by *extension* for τ. Considering its display $\partial_\tau(\psi^\tau) = \varphi^\tau_B \vdash (\varphi^\tau_C \uplus \{(c, c')\})$, we do have $\psi^\tau_C \in \mathcal{S}_+(C)$ as $\varphi^\tau_C \in \mathcal{S}_+(C)$ and by *+-receptivity* of tcgs. This concludes this case – other components are unchanged. All missing cases are symmetric. Finally, *securedness* follows from the diagram and securedness of $(x^\sigma, \theta, x^\tau)$, as all vertical morphisms in the diagram are order-isomorphisms.

Uniqueness. Consider now $\psi^\sigma \,:\, x^\sigma \cong_\sigma z^\sigma$ and $\psi^\tau \,:\, x^\tau \cong_\tau z^\tau$ also satisfying the hypotheses. But then, by composition of symmetries we also have

$$\omega^\sigma = \psi^\sigma \circ (\varphi^\sigma)^{-1} \,:\, y^\sigma \cong_\sigma z^\sigma\,, \qquad\qquad \omega^\tau = \psi^\tau \circ (\varphi^\tau)^{-1} \,:\, y^\tau \cong_\tau z^\tau\,,$$

but additionally these symmetries are causally compatible, since

$$\begin{aligned} \omega^\sigma_B &= \psi^\sigma_B \circ (\varphi^\sigma_B)^{-1} \\ &= \psi^\tau_B \circ \theta \circ (\varphi^\tau \circ \theta)^{-1} \\ &= \omega^\tau_B \end{aligned}$$

hence we have $\omega^\tau \circledast \omega^\sigma \,:\, y^\tau \circledast y^\sigma \cong_{\tau \circledast \sigma} z^\tau \circledast z^\sigma$ by Proposition 7.3.6. But additionally,

$$(\omega^\tau \circledast \omega^\sigma)_A = \psi^\sigma_A \circ (\varphi^\sigma_A)^{-1} \in \mathcal{S}_-(A)\,,$$

and likewise $(\omega^\tau \circledast \omega^\sigma)_C \in \mathcal{S}_+(C)$, so by *thin* for $\tau \circledast \sigma$, $\omega^\tau \circledast \omega^\sigma = \mathrm{id}_{y^\tau \circledast y^\sigma}$. It follows from Proposition 7.3.6 that ω^σ and ω^τ are identities, so $\varphi^\sigma = \psi^\sigma$ and $\varphi^\tau = \psi^\tau$. □

The above may be applied to compose configurations up to symmetry:

Corollary 7.4.5. *Consider* $(x^\sigma, \theta, x^\tau)$ *a* minimal *causally compatible triple.*
Then, the pair (y^σ, y^τ) *given by Proposition 7.4.4 is* minimal *causally compatible.*

Proof. By Proposition 7.4.4, we have $\varphi^\sigma : x^\sigma \cong_\sigma y^\sigma$ and $\varphi^\tau : x^\tau \cong_\tau y^\tau$ making

$$\begin{array}{ccc} x^\sigma \parallel x^\tau_C & \xrightarrow{\varphi[x^\sigma,\theta,x^\tau]} & x^\sigma_A \parallel x^\tau \\ {\scriptstyle \varphi^\sigma \parallel \varphi^\tau_C}\downarrow & & \downarrow{\scriptstyle \varphi^\sigma_A \parallel \varphi^\tau} \\ y^\sigma \parallel y^\tau_C & \xrightarrow[\varphi[y^\sigma,y^\tau]]{} & y^\sigma_A \parallel y^\tau \end{array}$$

commute. But the vertical morphisms are order-isomorphisms, inducing an order-iso between the (graphs of) $\varphi[x^\sigma, \theta, x^\tau]$ and $\varphi[y^\sigma, y^\tau]$. In particular it preserves maximal elements, so that the minimality of (y^σ, y^τ) follows from that of $(x^\sigma, \theta, x^\tau)$. □

This finally lets us define the composition of configurations up to symmetry:

Definition 7.4.6. *Consider* $(x^\sigma, \theta, x^\tau)$ *a* minimal *causally compatible triple. We set*

$$x^\tau \odot_\theta x^\sigma = y^\tau \odot y^\sigma \in \mathscr{C}(\tau \odot \sigma)$$

for y^σ, y^τ *the minimal causally compatible pair given by Corollary 7.4.5.*

It will be useful later on that this construction preserves unions:

Lemma 7.4.7. *Consider* $(x^\sigma_1, \theta_1, x^\tau_1), (x^\sigma_2, \theta_2, x^\tau_2)$ *minimal causally compatible triples, also compatible which each other in the sense that* $(x^\sigma_1 \cup x^\sigma_2, \theta_1 \cup \theta_2, x^\tau_2 \cup x^\tau_2)$ *is a minimal causally compatible triple. Then,* $x^\tau_1 \odot_{\theta_1} x^\sigma_1$ *and* $x^\tau_2 \odot_{\theta_2} x^\sigma_2$ *are compatible, and*

$$(x^\tau_1 \cup x^\tau_2) \odot_{\theta_1 \cup \theta_2} (x^\sigma_1 \cup x^\sigma_2) = (x^\tau_1 \odot_{\theta_1} x^\sigma_2) \cup (x^\tau_2 \odot_{\theta_2} x^\sigma_2)$$

Proof. Applying Proposition 7.4.4 to $(x^\sigma, \theta, x^\tau) = (x^\sigma_1 \cup x^\sigma_2, \theta_1 \cup \theta_2, x^\tau_2 \cup x^\tau_2)$, we get

$$\begin{array}{ccccccccc} x^\sigma \parallel x^\tau_C & \xrightarrow{\partial_\sigma \parallel x^\tau_C} & x^\sigma_A \parallel x^\sigma_B \parallel x^\tau_C & \xrightarrow{x^\sigma_A \parallel \theta \parallel x^\tau_C} & x^\sigma_A \parallel x^\tau_B \parallel x^\tau_C & \xrightarrow{x^\sigma_A \parallel \partial^{-1}_\tau} & x^\sigma_A \parallel x^\tau \\ & {\scriptstyle \varphi^\sigma \parallel \varphi^\tau_C} \searrow & & {\scriptstyle \varphi^\sigma_A \parallel \varphi^\sigma_B \parallel \varphi^\tau_C} \searrow \quad \swarrow {\scriptstyle \varphi^\sigma_A \parallel \varphi^\tau_B \parallel \varphi^\tau_C} & & & \swarrow {\scriptstyle \varphi^\sigma_A \parallel \varphi^\tau} \\ & y^\sigma \parallel y_C & \xrightarrow[\partial_\sigma \parallel y_C]{} & y_A \parallel y_B \parallel y_C & \xrightarrow[y_A \parallel \partial^{-1}_\tau]{} & y_A \parallel y^\tau & \end{array}$$

with similar diagrams for $(x^\sigma_1, \theta_1, x^\tau_1)$ and $(x^\sigma_2, \theta_2, x^\tau_2)$ yielding causally compatible pairs (y^σ_1, y^τ_1) and (y^σ_2, y^τ_2). But the diagram above restricts to diagrams for $(x^\sigma_1, \theta_1, x^\tau_1)$ and $(x^\sigma_2, \theta_2, x^\tau_2)$, so by the uniqueness clause of Proposition 7.4.4 it must be the componentwise union of the diagrams for $(x^\sigma_1, \theta_1, x^\tau_1)$ and $(x^\sigma_2, \theta_2, x^\tau_2)$. Thus, we have

$$\varphi[y^\sigma, y^\tau] = \varphi[y^\sigma_1, y^\tau_1] \cup \varphi[y^\sigma_2, y^\tau_2]$$

which, restricting to prime sub-secured bijections with visible top, yields

$$x^\tau \odot_\theta x^\sigma = (x^\tau_1 \odot_{\theta_1} x^\sigma_1) \cup (x^\tau_2 \odot_{\theta_2} x^\sigma_2)$$

as required, implying also their compatibility. □

7.4.2 Horizontal Composition

We can now use this to define horizontal composition of positive morphisms:

Proposition 7.4.8. *Consider $\sigma, \sigma' : A \vdash B$ and $\tau, \tau' : B \vdash C$ (pre)strategies, along with $f : \sigma \Rightarrow \sigma'$ and $g : \tau \Rightarrow \tau'$ positive morphisms.*

Then, there is a unique positive morphism $g \odot f : \tau \odot \sigma \to \tau' \odot \sigma'$ such that

$$(g \odot f)(x^\tau \odot x^\sigma) = g(x^\tau) \odot_{g[x^\tau]_B \circ f[x^\sigma]_B^{-1}} f(x^\sigma)$$

for all $x^\tau \odot x^\sigma \in \mathscr{C}(\tau \odot \sigma)$ where $f[x^\sigma]_B$ and $g[x^\tau]_B$ arise as below (7.6)*.*

Proof. Existence. Unfortunately, we cannot directly apply Lemma 7.2.11, because we want this proposition to handle positive morphisms between *prestrategies*. Thus, we construct $g \odot f$ via Lemma 7.1.9. Its action on configurations is given by definition – in particular, it follows immediately from *rigidity* that if (x^σ, x^τ) is minimal causally compatible, then $(f(x^\sigma), g[x^\tau]_B \circ f[x^\sigma]_B^{-1}, g(x^\tau))$ is minimal causally compatible. Preservation of unions follows from Lemma 7.4.7 and the fact that

$$g[x^\tau]_B \circ f[x^\sigma]_B^{-1} = (g[x_1^\tau]_B \circ f[x_1^\sigma]_B^{-1}) \cup (g[x_2^\tau]_B \circ f[x_2^\sigma]_B^{-1})$$

whenever $x^\sigma = x_1^\sigma \cup x_2^\sigma$ and $x^\tau = x_1^\tau \cup x_2^\tau$. Preservation of cardinality follows from the preservation of the display to the game (see *positivity* below).

We must show that this action extends to symmetries. Consider

$$\psi^\tau \odot \psi^\sigma : x^\tau \odot x^\sigma \cong_{\tau \odot \sigma} y^\tau \odot y^\sigma$$

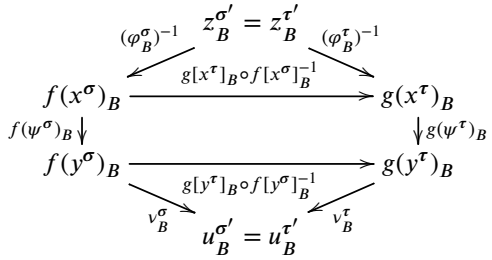
a symmetry in $\tau \odot \sigma$; so $\psi^\sigma : x^\sigma \cong_\sigma y^\sigma$, $\psi^\tau : x^\tau \cong_\tau y^\tau$ with $\psi_B^\sigma = \psi_B^\tau$. We name:

$$\begin{array}{ccccc} f(x^\sigma)_B & \xleftarrow{(\varphi_B^\sigma)^{-1}} & x_B^\sigma = x_B^\tau & \xrightarrow{g[x^\tau]_B} & g(x^\tau)_B \\ & \searrow^{(\nu_B^\tau)^{-1}} & & \swarrow_{\varphi_B^\tau} & \\ & & z_B^{\sigma'} = z_B^{\tau'} & & \end{array} \qquad \begin{array}{ccccc} f(y^\sigma)_B & \xleftarrow{f[y^\sigma]_B} & y_B^\sigma = y_B^\tau & \xrightarrow{g[y^\tau]_B} & g(y^\tau)_B \\ & \searrow^{\nu_B^\sigma} & & \swarrow_{\nu_B^\tau} & \\ & & u_B^{\sigma'} = u_B^{\tau'} & & \end{array}$$

writing $z^{\tau'} \odot z^{\sigma'} = g(x^\tau) \odot_\theta f(x^\sigma)$ for $\theta = g[x^\tau]_B \circ f[x^\sigma]_B^{-1}$, and $u^\tau \odot u^\sigma = g(y^\tau) \odot_\vartheta f(y^\sigma)$ for $\vartheta = g[y^\tau]_B \circ f[y^\sigma]_B^{-1}$. Next we note that the diagram below commutes

$$\begin{array}{ccccc} f(x^\sigma)_B & \xleftarrow{f[x^\sigma]_B} & x_B^\sigma = x_B^\tau & \xrightarrow{g[x^\tau]_B} & g(x^\tau)_B \\ \downarrow{\scriptstyle f(\psi^\sigma)_B} & & \downarrow{\scriptstyle \psi_B^\sigma = \psi_B^\tau} & & \downarrow{\scriptstyle g(\psi^\tau)_B} \\ f(y^\sigma)_B & \xleftarrow[f[y^\sigma]_B]{} & y_B^\sigma = y_B^\tau & \xrightarrow[g[y^\tau]_B]{} & g(y^\tau)_B \end{array}$$

by definition of the components involved. But this means that the following diagram

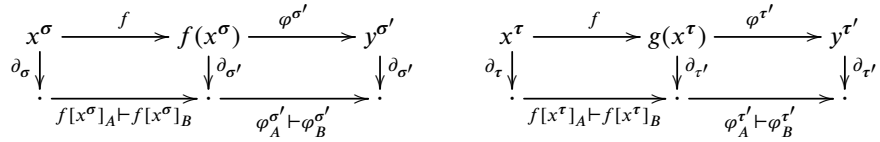


commutes as well; so we may set the image $(g \odot f)(\psi^{\tau} \odot \psi^{\sigma})$ of $\psi^{\tau} \odot \psi^{\sigma}$ as

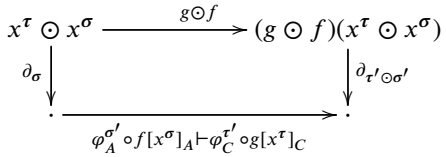
$$(\nu^{\sigma} \circ f(\psi^{\sigma}) \circ (\varphi^{\sigma})^{-1}) \odot (\nu^{\tau} \circ g(\psi^{\tau}) \circ (\varphi^{\tau})^{-1}) \,:\, (g \odot f)(x^{\tau} \odot x^{\sigma}) \cong_{\tau' \odot \sigma'} (g \odot f)(y^{\tau} \odot y^{\sigma})$$

and as Proposition 7.4.4 entails that φ^{σ} is uniquely determined by $f(\psi^{\sigma})$ and similarly for φ^{τ}, ν^{σ} and ν^{τ}, it follows that this construction is monotone.

Positivity. Write $y^{\tau'} \odot y^{\sigma'} = (g \odot f)(x^{\tau} \odot x^{\sigma})$. We have



where in each case the left hand square commutes by definition of positive morphisms, and $\varphi^{\sigma'}$, $\varphi^{\tau'}$ provided by Proposition 7.4.4 – which also ensure that $\varphi^{\sigma'}_{A} \in \mathcal{S}_{-}(A)$ and $\varphi^{\tau'}_{C} \in \mathcal{S}_{+}(C)$. Now from the above, by definition and monotonicity of the constructions,



(7.7)

commutes as well, and $\varphi^{\sigma'}_{A} \circ f[x^{\sigma}]_{A} \in \mathcal{S}_{-}(A)$ and $\varphi^{\tau'}_{C} \circ g[x^{\tau}]_{C} \in \mathcal{S}_{+}(C)$ as required.

Uniqueness. Direct consequence of Lemma 6.1.12. □

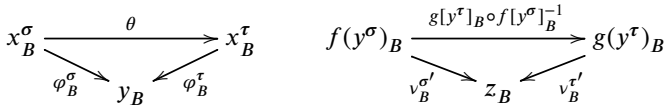
In particular, this entails that thin equivalence is a congruence, *i.e.* if $\boldsymbol{\sigma} \approx \boldsymbol{\sigma}' : A \vdash B$ and $\boldsymbol{\tau} \approx \boldsymbol{\tau}' : B \vdash C$, then $\boldsymbol{\tau} \odot \boldsymbol{\sigma} \approx \boldsymbol{\tau}' \odot \boldsymbol{\sigma}' : A \vdash C$. This will be useful when, later on, we truncate our bicategory **TCG** into a $\sim$*-category* – see Section 7.4.4.

We prove one final lemma on horizontal composition:

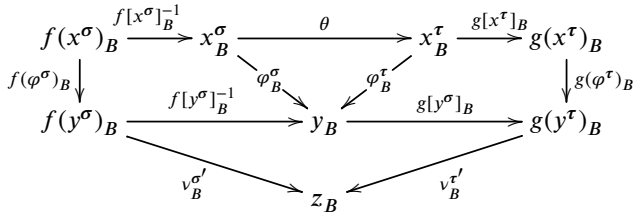
Lemma 7.4.9. *Consider $f : \boldsymbol{\sigma} \to \boldsymbol{\sigma}'$ and $g : \boldsymbol{\tau} \to \boldsymbol{\tau}'$ positive morphisms.*
Then, for all minimal causally compatible triple $(x^{\sigma}, \theta, x^{\tau})$, we have:

$$(g \odot f)(x^{\tau} \odot_{\theta} x^{\sigma}) = g(x^{\tau}) \odot_{g[x^{\tau}]_{B} \circ \theta \circ f[x^{\sigma}]_{B}^{-1}} f(x^{\sigma})\,.$$

Proof. Let us write $y^{\tau} \odot y^{\sigma} = x^{\tau} \odot_{\theta} x^{\sigma}$ and $z^{\tau} \odot z^{\sigma} = (g \odot f)(y^{\tau} \odot y^{\sigma})$; by definition of $x^{\tau} \odot_{\theta} x^{\sigma}$ and of $(g \odot f)(y^{\tau} \odot y^{\sigma})$, we have symmetries $\varphi^{\sigma} : x^{\sigma} \cong_{\sigma} y^{\sigma}$, $\varphi^{\tau} : x^{\tau} \cong_{\tau} y^{\tau}$, $\nu^{\sigma'} : f(y^{\sigma}) \cong_{\sigma'} z^{\sigma'}$ and $\nu^{\tau'} : g(y^{\tau}) \cong_{\tau'} z^{\tau'}$, satisfying the triangles:



It follows that the following diagram commutes as well:



which proves our equality by uniqueness of Proposition 7.4.4. □

7.4.3 Bicatorical Laws

Proving the bicategorical laws requires one final technical development, establishing an associativity property of synchronization up to symmetry.

Ternary synchronization up to symmetry. Fix three strategies

$$\sigma : A \vdash B, \qquad \tau : B \vdash C, \qquad \delta : C \vdash D$$

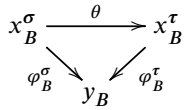
with configurations $x^{\sigma} \in \mathscr{C}(\sigma)$, $x^{\tau} \in \mathscr{C}(\tau)$, $x^{\delta} \in \mathscr{C}(\delta)$, and mediating symmetries

$$\theta : x^{\sigma}_B \cong_B x^{\tau}_B, \qquad \omega : x^{\tau}_C \cong_C x^{\delta}_C.$$

What can we say about their synchronizations up to symmetry? Assuming $(x^{\sigma}, \theta, x^{\tau})$ is minimal causally compatible, by Proposition 7.4.4 there are unique

$$\varphi^{\sigma} : x^{\sigma} \cong_{\sigma} y^{\sigma}, \qquad \varphi^{\tau} : x^{\tau} \cong_{\tau} y^{\tau}$$

such that $\varphi^{\sigma}_A \in \mathcal{S}_-(A)$, $\varphi^{\tau}_C \in \mathcal{S}_+(C)$ and the following diagram commutes



and by Corollary 7.4.5, y^{σ}, y^{τ} are minimal causally compatible. With these notations:

Lemma 7.4.10. *Assume* $(x^{\sigma}, \theta, x^{\tau})$ *and* $(x^{\tau} \odot_{\theta} x^{\sigma}, \omega \circ (\varphi^{\tau}_C)^{-1}, x^{\delta})$ *are min. caus. comp. Then, so is* $(x^{\tau}, \omega, x^{\delta})$.

Proof. In that case, then by Proposition 7.4.4 there are unique

$$\Psi^{\sigma,\tau} : y^\tau \odot y^\sigma \cong_{\tau\odot\sigma} u^\tau \odot u^\sigma , \qquad \Psi^\delta : x^\delta \cong_\delta u^\delta$$

such that $\Psi^{\sigma,\tau}_A \in \mathcal{S}_-(A)$, $\Psi^\delta_D \in \mathcal{S}_+(D)$ and the following diagram commutes:

$$\begin{array}{ccc} (y^\tau \odot y^\sigma)_C & \xrightarrow{\omega\circ(\varphi^\tau_C)^{-1}} & x^\delta_C \\ & {\scriptstyle \Psi^{\sigma,\tau}_C} \searrow \quad \swarrow {\scriptstyle \Psi^\delta_C} & \\ & u_C & \end{array}$$

recalling that $\Psi^{\sigma,\tau} : y^\tau \odot x^\sigma \cong_{\tau\odot\sigma} u^\tau \odot u^\sigma$ consists in $\Psi^\sigma : x^\sigma \cong_\sigma u^\sigma$ and $\Psi^\tau : x^\tau \cong_\tau u^\tau$ matching on B. By Corollary 7.4.5, $u^\tau \odot u^\sigma$ and u^δ are minimal causally compatible. By Lemma 6.2.24, it follows that u^τ, u^δ are minimal causally compatible. But then

$$\Psi^\tau\circ\varphi^\tau : x^\tau \cong_\tau u^\tau , \qquad \Psi^\delta : x^\delta \cong_\delta u^\delta$$

are order-isomorphisms such that the following triangle commutes:

$$\begin{array}{ccc} x^\tau_C & \xrightarrow{\omega} & x^\delta_C \\ & {\scriptstyle (\Psi^\tau\circ\varphi^\tau)_C} \searrow \quad \swarrow {\scriptstyle \Psi^\delta_C} & \\ & u_C & \end{array}$$

Following these isomorphisms, it is a direct verification to establish causal compatibility of $(x^\tau, \omega, x^\delta)$ from that of (u^τ, u^δ). □

Now, assuming that the conditions of this lemma are satisfied, we know that $(x^\tau, \omega, x^\delta)$ are causally compatible. This means, again by Proposition 7.4.4, that there are

$$\psi^\tau : x^\tau \cong_\tau z^\tau , \qquad \psi^\delta : x^\delta \cong_\delta z^\delta$$

such that the following triangle commutes:

$$\begin{array}{ccc} x^\tau_B & \xrightarrow{\omega} & x^\delta_B \\ & {\scriptstyle \psi^\tau_B} \searrow \quad \swarrow {\scriptstyle \psi^\delta_B} & \\ & z_B & \end{array}$$

Using these notations, we can then prove the following lemma:

Lemma 7.4.11. *Assume $(x^\sigma, \theta, x^\tau)$ and $(x^\tau \odot_\theta x^\sigma, \omega\circ(\varphi^\tau_C)^{-1}, x^\delta)$ are min. caus. comp. Then, so is $(x^\sigma, \psi^\tau_B\circ\theta, x^\delta \odot_\omega x^\tau)$.*

Proof. By Corollary 7.4.5, u^σ and $u^\delta \odot u^\tau$ are minimal causally compatible. As for the previous lemma, the idea is to provide symmetries making minimal causal compatibility of $(x^\sigma, \psi^\tau_B\circ\theta, x^\delta \odot_\omega x^\tau)$ boil down to that of $(u^\sigma, u^\delta \odot u^\tau)$. First there is:

$$\Psi^\sigma\circ\varphi^\sigma : x^\sigma \cong_\sigma u^\sigma ,$$

and likewise we have two symmetries

$$\Psi^{\tau}\circ\varphi^{\tau}\circ(\psi^{\tau})^{-1} : z^{\tau} \cong_{\tau} u^{\tau}\,, \qquad \Psi^{\delta}\circ(\psi^{\delta})^{-1} : z^{\delta} \cong_{\delta} u^{\delta}$$

which can be checked to have the same projection on C. So their synchronization yields

$$\Phi = (\Psi^{\delta}\circ(\psi^{\delta})^{-1}) \odot (\Psi^{\tau}\circ\varphi^{\tau}\circ(\psi^{\tau})^{-1}) : x^{\delta} \odot_{\omega} x^{\tau} \cong_{\delta\odot\tau} u^{\delta} \odot u^{\tau}\,.$$

Finally, a direct verification shows that the triangle below commutes:

$$\begin{array}{ccc} x^{\sigma}_{B} & \xrightarrow{\psi^{\tau}_{B}\circ\theta} & (x^{\delta} \odot_{\omega} x^{\tau})_{B} \\ & {\scriptstyle \Psi^{\sigma}\circ\varphi^{\sigma}}\searrow \quad \swarrow{\scriptstyle \Phi_{B}} & \\ & u_{B} & \end{array} \tag{7.8}$$

via which the announced min. caus. comp. boils down to that of $(u^{\sigma}, u^{\delta} \odot u^{\tau})$. □

At this point we are equipped to state the seeked associativity property:

Lemma 7.4.12. *If* $(x^{\sigma}, \theta, x^{\tau})$ *and* $(x^{\tau} \odot_{\theta} x^{\sigma}, \omega\circ(\varphi^{\tau}_{C})^{-1}, x^{\delta})$ *are min. caus. comp.,*

$$x^{\delta} \odot_{\omega\circ(\varphi^{\tau}_{C})^{-1}} (x^{\tau} \odot_{\theta} x^{\sigma}) = a_{\sigma,\tau,\delta}((x^{\delta} \odot_{\omega} x^{\tau}) \odot_{\psi^{\tau}_{B}\circ\theta} x^{\sigma})\,.$$

Proof. Keeping the notations above, $x^{\delta} \odot_{\omega\circ(\varphi^{\tau}_{C})^{-1}} (x^{\tau} \odot_{\theta} x^{\sigma}) = u^{\delta} \odot (u^{\tau} \odot u^{\sigma})$. Now,

$$\Psi : x^{\delta} \odot_{\omega} x^{\tau} \cong_{\delta\odot\tau} u^{\delta} \odot u^{\tau}\,, \qquad \Psi^{\sigma}\circ\varphi^{\sigma} : x^{\sigma} \cong_{\sigma} u^{\sigma}$$

are symmetries satisfying (7.8); and it is direct from the definition that $\Psi_{A} \in \mathcal{S}_{-}(A)$ and $(\Psi^{\sigma}\circ\varphi^{\sigma})_{C} \in \mathcal{S}_{+}(C)$. Consequently, by the uniqueness clause of Proposition 7.4.4,

$$(x^{\delta} \odot_{\omega} x^{\tau}) \odot_{\psi^{\tau}_{B}\circ\theta} x^{\sigma} = (u^{\delta} \odot u^{\tau}) \odot u^{\sigma}$$

and $a_{\sigma,\tau,\delta}((u^{\delta} \odot u^{\tau}) \odot u^{\sigma}) = u^{\delta} \odot (u^{\tau} \odot u^{\sigma})$ as required. □

A bicategory. At last, we have the ingredients to conclude:

Theorem 7.4.13. *There is a bicategory* **TCG** *with* games *as objects,* strategies *as morphisms, and* positive morphisms *as* 2*-cells.*

Proof. Identities are defined in Proposition 7.3.15, and composition in Proposition 7.3.1. Each strategy has an identity 2-cell (the identity positive isomorphism). Vertical composition of positive morphisms is composition in **ESS**, and horizontal composition is Proposition 7.4.8. Associators are defined in Proposition 7.4.1 and unitors in Proposition 7.4.2. It remains to prove the following laws:

Functoriality of horizontal composition. For functoriality, we compute

$$\begin{aligned}
& (g' \odot f') \circ (g \odot f)(x^{\tau} \odot x^{\sigma}) \\
= \;& (g' \odot f')(g(x^{\tau}) \odot_{g[x^{\tau}]_B \circ f[x^{\sigma}]_B^{-1}} f(x^{\sigma})) \\
= \;& g'(g(x^{\tau})) \odot_{g'[g(x^{\tau})]_B \odot g[x^{\tau}]_B \odot f[x^{\sigma}]_B^{-1} \circ f'[f(x^{\sigma})]_B^{-1}} f'(f(x^{\sigma})) \\
= \;& (g' \circ g)(x^{\tau}) \odot_{(g' \circ g)[x^{\tau}]_B \circ (f' \circ f)[x^{\sigma}]_B^{-1}} (f' \circ f)(x^{\sigma}) \\
= \;& ((g' \circ g) \odot (f' \circ f))(x^{\tau} \odot x^{\sigma})
\end{aligned}$$

using definition of $g \odot f$, Lemma 7.4.9, definition of the preservation of symmetry of $g' \circ g$ and $f' \circ f$, and definition of $(g' \circ g) \odot (f' \circ f)$. By Lemma 6.1.12, we conclude that $(g' \odot f') \circ (g \odot f) = (g' \circ g) \odot (f' \circ f)$ as required. Likewise, it is straightforward using the same lemmas that horizontal composition preserves identity positive morphisms.

Naturality of unitors. We first show the naturality of $l_{\sigma} : \mathfrak{C}_B \odot \sigma \cong \sigma$ in σ, *i.e.* that for all $\sigma, \sigma' : A \vdash B$ and $f : \sigma \Rightarrow \sigma'$ a positive morphism, we have $f \circ l_{\sigma} = l_{\sigma'} \circ (\mathfrak{C}_B \odot f)$, and by Corollary 7.2.12 it suffices to prove the equality for $+$-covered configurations. By Lemma 6.4.4, $+$-covered configurations in $\mathfrak{C}_B \odot \sigma$ have form $\mathfrak{C}_{x^{\sigma}_B} \odot x^{\sigma} \in \mathscr{C}^+(\mathfrak{C}_B \odot \sigma)$, and by definition of horizontal composition (in Proposition 7.4.8),

$$(\mathfrak{C}_B \odot f)(\mathfrak{C}_{x^{\sigma}_B} \odot x^{\sigma}) = \mathfrak{C}_{x^{\sigma}_B} \odot_{f[x^{\sigma}]_B^{-1}} f(x^{\sigma}),$$

but that must be $\mathfrak{C}_{f(x^{\sigma})_B} \odot f(x^{\sigma}) \in \mathscr{C}^+(\mathfrak{C}_B \odot \sigma')$ as it is easily proved to satisfy the properties in Proposition 7.4.4 defining it uniquely.

Now, from the characterization of the action of unitors in Proposition 7.4.2,

$$l_{\sigma'}(\mathfrak{C}_{f(x^{\sigma})_B} \odot f(x^{\sigma})) = f(x^{\sigma})$$

which is also $f(l_{\sigma}(\mathfrak{C}_{x^{\sigma}_B} \odot x^{\sigma})) = f(x^{\sigma})$. The proof for r_{σ} is symmetric.

Naturality of associators. Consider now $f : \sigma \Rightarrow \sigma' : A \vdash B$, $g : \tau \Rightarrow \tau' : B \vdash C$ and $h : \delta \Rightarrow \delta' : C \vdash D$ positive morphisms, and calculate:

$$\begin{aligned}
& (h \odot (g \odot f)) \circ a_{\sigma,\tau,\delta}((x^{\delta} \odot x^{\tau}) \odot x^{\sigma}) \\
= \;& (h \odot (g \odot f))(x^{\delta} \odot (x^{\tau} \odot x^{\sigma})) \\
= \;& h(x^{\delta}) \odot_{h[x^{\delta}]_C \circ (g \odot f)[x^{\tau} \odot x^{\sigma}]_C^{-1}} (g \odot f)(x^{\tau} \odot x^{\sigma}) \\
= \;& h(x^{\delta}) \odot_{h[x^{\delta}]_C \circ (g \odot f)[x^{\tau} \odot x^{\sigma}]_C^{-1}} (g(x^{\tau}) \odot_{g[x^{\tau}]_B \circ h[x^{\sigma}]_B^{-1}} f(x^{\sigma})) \\
= \;& h(x^{\delta}) \odot_{h[x^{\delta}]_C \circ g[x^{\tau}]_C^{-1} \circ (\varphi_C^{\tau'})^{-1}} (g(x^{\tau}) \odot_{g[x^{\tau}]_B \circ f[x^{\sigma}]_B^{-1}} f(x^{\sigma})) \\
= \;& a_{\sigma',\tau',\delta'}((h(x^{\delta}) \odot_{h[x^{\delta}]_C \circ g[x^{\tau}]_C^{-1}} g(x^{\tau})) \odot_{\psi_B^{\tau} \circ g[x^{\tau}]_B \circ f[x^{\sigma}]_B^{-1}} f(x^{\sigma})) \\
= \;& a_{\sigma',\tau',\delta'}((h(x^{\delta}) \odot_{h[x^{\delta}]_C \circ g[x^{\tau}]_C^{-1}} g(x^{\tau})) \odot_{(h \odot g)[x^{\delta} \odot x^{\tau}]_B \circ f[x^{\sigma}]_B^{-1}} f(x^{\sigma})) \\
= \;& a_{\sigma',\tau',\delta'}((h \odot g)(x^{\tau} \odot x^{\sigma}) \odot_{(h \odot g)[x^{\delta} \odot x^{\tau}]_B \circ f[x^{\sigma}]_B^{-1}} f(x^{\sigma})) \\
= \;& a_{\sigma',\tau',\delta'}(((h \odot g) \odot f)((x^{\delta} \odot x^{\tau}) \odot x^{\sigma}))
\end{aligned}$$

using Lemma 7.4.9 twice, then the witness exposed in (7.7) for the fact that $g \odot f$ is a positive morphism, then Lemma 7.4.12 (using the same notations for φ and ψ), then

again the witness in (7.7) for the fact that $h \circ g$ is a positive morphism, concluding with two applications of Lemma 7.4.9 – which finally ends the proof. □

Though we have constructed a bicategory **TCG**, most of this monograph relies on a simpler structure ignoring coherence laws, sufficient for many semantic purposes.

7.4.4 A $\sim$-category

To wrap up this chapter, we introduce a simplified higher-dimensional structure.

In order to use **TCG** for semantic purposes, we need much additional categorical structure: in particular, we must at least have a cartesian closed structure in order to interpret the simply-typed λ-calculus. Following Theorem 7.4.13, the reader may naturally expect us to aim towards constructing the *bicategorical* version of this structure. Such a construction is described by Paquet in [Paquet, 2020]. But ultimately, the explicit managing of 2-cells does not appear particularly useful for the semantic purposes of this monograph so we opted to not develop this higher-dimensional structure further.

Alternatively, we could quotient **TCG** to a category having as morphisms equivalence classes of strategies with respect to positive isomorphism (as done for instance in AJM games [Abramsky et al., 2000]). But this is actually harmful: it is methodologically better for the interpretation to yield actual concrete strategies, rather than equivalence classes (for instance, recursion is obtained via completeness properties of an ordering on strategies that does not seem to lift to equivalence classes). We also want to avoid the vagueness often encountered in switching implicitely between equivalence classes and representatives.

So we shall work neither with categories nor bicategories but with $\sim$*-categories*:

Definition 7.4.14. *A (small)* $\sim$***-category*** C *consists in a set of* objects C_0*; for each* A, B*, a set of* morphisms $C(A, B)$ *with an equivalence relation* $\sim$*; a* composition *operation*

$$(- \circ -) : C(B, C) \times C(A, B) \to C(A, C)$$

for all $A, B, C \in C_0$*; an* identity *morphism* $\mathrm{id}_A \in C(A, A)$ *for all* $A \in C_0$*, subject to:*

associativity:	*for all* $f \in C(A, B), g \in C(B, C), h \in C(C, D)$, $(h \circ g) \circ f \sim h \circ (g \circ f)$,
identity:	*for all* $f \in C(A, B)$, $\mathrm{id}_B \circ f \sim f \circ \mathrm{id}_A \sim f$,
congruence:	*for all* $f \sim f' \in C(A, B)$, $g \sim g' \in C(B, C)$, $g' \circ f' \sim g \circ f$.

In other words, a $\sim$-category is a special case of a bicategory where any two morphisms are related by at most one (invertible) 2-cell. Any bicategory yields a $\sim$-category where $f \sim g$ iff f and g are related by an isomorphism.

Using $\sim$-categories avoids quotienting morphisms, while making all coherence laws hold vacuously. All categorical notions extend transparently to $\sim$-categories, with equations holding up to $\sim$ rather than equality. For instance, functors between $\sim$-categories respect $\sim$, and preservation of identities and composition only holds up to $\sim$. In the

sequel, we shall use several standard categorical notions in the $\sim$-categorical world without explicitly spelling them out – hopefully this causes no confusion.

From now on, we regard **TCG** as a $\sim$-category:

Proposition 7.4.15. *There is* **TCG***, a $\sim$-category with tcgs as objects, strategies as morphisms, and equivalence relation $\approx$ the thin equivalence, from Corollary 7.2.9.*

7.5 History and Related Work

Genesis of TCG. I arrived as a postdoc with Glynn Winskel in Cambridge in September 2011. After participating in a first project to enrich concurrent games with winning conditions [Clairambault et al., 2012], my main task was the supposedly simple objective to extend **CG** with *symmetry* [Winskel, 2007], in particular so that concurrent games could handle replication as in AJM games [Abramsky et al., 2000]. What was initially expected to be relatively straightforward proved to be an endeavour full of surprises, for which we were soon joined by a research intern, Simon Castellan.

As discussed ealier, there are two ways to handle uniformity in (copy indices-based) game semantics. The *fat* way, and the *thin* way. The fat approach to uniformity consists in forcing strategies to be saturated: if a strategy can play one copy index, it must play non-deterministically *all* copy indices. In contrast, in the thin approach, a strategy in a given state will play *one canonically chosen* copy index. Uniformity then becomes an interactive property: any change in Opponent's copy indices causes a subsequent change in Player's copy indices. This *thin/fat* distinction is not about concurrent games at all, but present as soon as there are strategies with copy indices: the original AJM games are thin [Abramsky et al., 2000], but they can be made fat – the very first fat model is [Baillot et al., 1997a]. The fat approach is simpler; however, it conflates uniformity with non-determinism and also is not a compatible extension of the symmetry-free layer as the basic mechanisms (identity and composition) must be changed.

In developing concurrent games with symmetry, we found that we could not make the thin approach work as we lacked a good way to describe the interactive reindexing process involved in uniformity. So the first published version of concurrent games with symmetry is fat[3] [Castellan et al., 2014][4]. However, saturation made the model difficult to use, and I kept looking for a thin version, whose maturing was a long process.

Thin concurrent games first appeared in [Castellan et al., 2015] (the paper was mainly about parallel innocence – see Chapter 10. The core of this setting is the one presented in this chapter, though with many crucial changes and simplifications. Many significant improvements (including the very definition of a tcg!) are due to Simon Castellan, developed during his PhD thesis; others came while writing the journal version detailing thin concurrent games [Castellan et al., 2019], and others still appeared since.

[3] In fact, the very first version of concurrent games with symmetry is thin, and only appears in Simon Castellan's (unpublished) internship report – however, we could only make this work adopting a coarse bisimulation-like equivalence relation on strategies, whereas we wanted an isomorphism.

[4] Not very well-known is the fact that this paper also contains the very first non-deterministic generalization of Hyland-Ong innocence!

Further comments. This chapter mainly presents constructions first introduced in [Castellan et al., 2015] and improved in [Castellan et al., 2019]. However, the latter uses results from [Castellan et al., 2017a], making for a somewhat intimidating chain of dependencies. In contrast, the present chapter is entirely self-contained and constructs thin concurrent games from the ground, with full details.

Finally, this chapter builds a bicategory, whereas the papers cited above only construct a category up to positive isomorphisms. This is not a new result: in his PhD thesis, Paquet proves that thin concurrent games form a bicategory (actually a symmetric monoidal closed bicategory [Paquet, 2020]). However, Paquet does not detail the naturality of unitors and associators; so our proof complements his.

It seems fair to say that **TCG** is the most expressive and fine-grained intensional model on the market. Much of our further work builds on it, and there is a lot left to explore. I consider this to be one of the main contributions in this line of work.

Related work. The core idea behind **TCG** is splitting symmetries into *positive* and *negative* symmetries: it is thanks to this separation that we are able to factor out Opponent's reindexings when expressing uniformity and when comparing strategies. As mentioned earlier, this idea was first developed by Melliès as his approach to uniformity in asynchronous games [Melliès, 2003]. There, he considers an *asynchronous game* with *two* groups acting on plays, one for Player reindexings and one for Opponent reindexings; uniformity is there specified as a *bi-invariance* property with respect to these group actions, analogous in spirit to Lemma 7.2.6 – note that recently, Paquet showed that Melliès' mechanism could also be applied in the setting of concurrent games [Paquet, 2023], obtaining a different, more global, notion of uniformity which differs from the present one for subtle reasons.

In any case, this polarized decomposition of symmetry is clearly a fundamental structure that deserves further exploration. One such recent exploration is a joint work with Simon Forest, in which we build a bicategory of *thin spans of groupoids* sharing much of the structure of **TCG**, but completely statically [Clairambault and Forest, 2023], *i.e.* without time or causality. There, an *object* is a groupoid with two sub-groupoids, the *positive* and the *negative* ones. Morphisms are spans of groupoids selected by a biorthogonality relation ensuring that their composition satisfies a universal property analogous to Proposition 7.3.1, but without causal compatibility. Overall, we get a proof-relevant relational model sharing much of the flavour of **TCG**, but avoiding the combinatorics of concurrent strategies and their composition.

Chapter 8

Constructing Games and Strategies

The previous chapter describes the bare bones of thin concurrent games: namely, the composition mechanism and the tools to handle symmetry. In the present chapter, we describe the constructions on games and strategies that will be exploited throughout this monograph, aiming to establish categorical structures useful for semantics.

In practive, we will build from **TCG** two radically different categorical structures.

TCG *per se* is an *unpolarized* category of games, in that games have no assigned polarity: in a game A, both players may be able to start. This is in contrast with most notions of games used in game semantics, which focus on *negative* games where only Opponent starts (for call-by-name languages), or on *positive* games where only Player starts (for call-by-value languages). A widespread urban legend has it that such non-polarized game settings suffer from a lack of associativity, due to the infamous *Blass problem* [Abramsky, 2003]. In fact, this phenomenon is very specific to Blass games, and Proposition 6.2.25 ensures that composition in **TCG** is indeed associative up to iso. But **TCG** is certainly not the first unpolarized category of games, the pioneering example is Joyal's category of Conway games [Joyal, 1977]. As pioneered by Joyal, the expected categorical structure of such an unpolarized category of games is that of a *compact closed category*, *i.e.* a degenerate model of MLL where $\otimes = \parr$. Our first objective in this chapter will be to show that **TCG** is a compact closed $\sim$-category.

To prepare the ground for semantics, we must then switch to *polarized* games – *negative*, since the languages considered here are call-by-name. We shall equip the negative sub-$\sim$-category of **TCG** with (a variation of) the structure of a *Seely category* [Melliès, 2009]. Seely categories are a standard model of ILL, and thus of the simply-typed λ-calculus; and we shall use this structure as the basis for our interpretations.

8.1 A Compact closed $\sim$-Category

Compact closed categories are amongst the simplest categorical structures that exhibit higher-order behaviour. They are essentially those degenerated models of multiplicative linear logic where the $\otimes$ and its dual $\parr$ coincide – intuitively, they resemble proof nets without the correctness criterion. Like Conway games [Joyal, 1977], categories of unpolarized games are often compact closed, and this structure underlies much of the additional structure appearing in polarized categories of games, as explored by Melliès [Melliès, 2004b]. It is thus natural to start there.

Recall that a compact closed category is a symmetric monoidal category $\mathcal{C}$ where each object A has a (necessarily unique up to iso) *dual object* A^*. This means we have

$$\begin{array}{rrcccc} \text{the } \textit{unit}, & \eta_A & : & I & \to & A \otimes A^* \\ \text{the } \textit{co-unit}, & \epsilon_A & : & A^* \otimes A & \to & I \end{array}$$

two morphisms for all A, respectively akin to the *axiom* and *cut* of proof nets, satisfying two coherence axioms [Kelly and Laplaza, 1980]. A compact closed $\sim$*-category* is defined likewise, with all laws holding up to $\sim$ instead of equality.

8.1.1 A bifunctor

For thin concurrent games we start with the first component, a $\sim$-bifunctor:

$$(- \parallel -) \quad : \quad \mathbf{TCG} \times \mathbf{TCG} \to \mathbf{TCG}\,.$$

Notice that we use $\parallel$ for the tensor of the compact closed structure, as we shall reserve $\otimes$ for a later construction in the context of negative games.

Definition. On tcgs, the construction $\parallel$ is defined in Section 7.1.4. If A and B are tcgs, then $A \parallel B$ is thought of as the game formed of A and B played in parallel. Note that though the games are independent, a strategy on $A \parallel B$ may very well impose crossed immediate causal links between the two components!

On strategies, the action of $\parallel$ is given by the following:

Proposition 8.1.1. *Consider A, B, C, D tcgs, and $\sigma : A \vdash B$, $\tau : C \vdash D$ strategies. Then, there is a strategy $\sigma \parallel \tau : A \parallel C \vdash B \parallel D$, unique up to iso, s.t. there are*

$$\begin{array}{rcccc} (- \parallel -) & : & \mathscr{C}^+(\sigma) \times \mathscr{C}^+(\tau) & \simeq & \mathscr{C}^+(\sigma \parallel \tau) \\ (- \parallel -) & : & \mathcal{S}^+(\sigma) \times \mathcal{S}^+(\tau) & \simeq & \mathcal{S}^+(\sigma \parallel \tau) \end{array}$$

order-isos commuting with $\mathsf{dom}, \mathsf{cod}$*, and s.t. for all $\theta^\sigma \in \mathcal{S}^+(\sigma)$ and $\theta^\tau \in \mathcal{S}^+(\tau)$,*

$$\partial_{\sigma \parallel \tau}(\theta^\sigma \parallel \theta^\tau) = (\theta^\sigma_A \parallel \theta^\tau_C) \parallel (\theta^\sigma_B \parallel \theta^\tau_D)\,.$$

Moreover, $(- \parallel -)$ preserves $\approx$ (a property we call congruence*).*

Proof. Existence. As the notation suggests, $\boldsymbol{\sigma} \parallel \boldsymbol{\tau}$ has ess the simple parallel composition of $\boldsymbol{\sigma}$ and $\boldsymbol{\tau}$. Its display map is set as:

$$\begin{array}{rcl} \partial_{\sigma\parallel\tau}(1,s) & = & \begin{cases} (1,(1,a)) & \text{if } \partial_{\sigma}(s) = (1,a) \\ (2,(1,b)) & \text{if } \partial_{\sigma}(s) = (2,b) \end{cases} \\ \partial_{\sigma\parallel\tau}(2,t) & = & \begin{cases} (1,(2,c)) & \text{if } \partial_{\tau}(t) = (1,c) \\ (2,(2,d)) & \text{if } \partial_{\tau}(t) = (2,d), \end{cases} \end{array}$$

we omit the routine verification that this yields a strategy. All additional conditions follow as in Lemma 7.1.22 along with routine verifications.

Uniqueness. Straightforward consequence of Corollary 7.2.13.

Congruence. Immediate from the definition. □

Functoriality. Proposition 8.1.1 makes functoriality easy. First:

Lemma 8.1.2. *For any tcgs A and B, we have* $\mathfrak{C}_{A\parallel B} \approx \mathfrak{C}_A \parallel \mathfrak{C}_B$.

Proof. We calculate:

$$\begin{array}{rcl} \mathscr{C}^+(\mathfrak{C}_A \parallel \mathfrak{C}_B) & \cong & \mathscr{C}^+(\mathfrak{C}_A) \times \mathscr{C}^+(\mathfrak{C}_B) \\ & = & \{x_A \parallel x_A \mid x_A \in \mathscr{C}(A)\} \times \{x_B \parallel x_B \mid x_B \in \mathscr{C}(B)\} \\ & \cong & \{(x_A \parallel x_B) \parallel (x_A \parallel x_B) \mid x_A \parallel x_B \in \mathscr{C}(A \parallel B)\} \\ & = & \mathscr{C}^+(\mathfrak{C}_{A\parallel B}) \end{array}$$

using Proposition 8.1.1, then Lemma 6.4.4, then the obvious bijection, then Lemma 6.4.4 again – all these are order-isos, which preserve display maps in the obvious sense.

The same reasoning applies to symmetries compatibly, using Proposition 8.1.1 and Lemma 7.3.14. Finally, the seeked isomorphism follows from Corollary 7.2.13. □

For preservation of composition, consider four strategies

$$\boldsymbol{\sigma}_1 : A_1 \vdash B_1\,, \qquad \boldsymbol{\sigma}_2 : A_2 \vdash B_2\,, \qquad \boldsymbol{\tau}_1 : B_1 \vdash C_1\,, \qquad \boldsymbol{\tau}_2 : B_2 \vdash C_2\,.$$

The only additional ingredient needed is the following lemma:

Lemma 8.1.3. *Take* $x^{\sigma_1} \in \mathscr{C}^+(\boldsymbol{\sigma}_1), x^{\sigma_2} \in \mathscr{C}^+(\boldsymbol{\sigma}_2), x^{\tau_1} \in \mathscr{C}^+(\boldsymbol{\tau}_1)$ *and* $x^{\tau_2} \in \mathscr{C}^+(\boldsymbol{\tau}_2)$.

Then, $x^{\sigma_1} \parallel x^{\sigma_2} \in \mathscr{C}^+(\boldsymbol{\sigma}_1 \parallel \boldsymbol{\sigma}_2)$ *and* $x^{\tau_1} \parallel x^{\tau_2} \in \mathscr{C}^+(\boldsymbol{\tau}_1 \parallel \boldsymbol{\tau}_2)$ *are causally compatible iff* (1) x^{σ_1} *and* x^{τ_1} *are causally compatible; and* (2) x^{σ_2} *and* x^{τ_2} *likewise.*

Proof. A routine verification, left to the reader. □

Now, we can finally conclude:

Proposition 8.1.4. *Parallel composition extends to a* $\sim$*-bifunctor:*

$$(- \parallel -) \quad : \quad \mathbf{TCG} \times \mathbf{TCG} \to \mathbf{TCG}\,.$$

Proof. Preservation of copycat is by Lemma 8.1.2. For composition, we calculate:

$$\begin{aligned}
& \mathscr{C}^+((\boldsymbol{\tau}_1 \parallel \boldsymbol{\tau}_2) \odot (\boldsymbol{\sigma}_1 \parallel \boldsymbol{\sigma}_2)) \\
\cong\ & \{(x^{\tau_1} \parallel x^{\tau_2}, x^{\sigma_1} \parallel x^{\sigma_2}) \in \mathscr{C}^+(\boldsymbol{\tau}_1 \parallel \boldsymbol{\tau}_2) \times \mathscr{C}^+(\boldsymbol{\sigma}_1 \parallel \boldsymbol{\sigma}_2) \mid \text{caus. comp.}\} \\
\cong\ & \{(x^{\tau_1}, x^{\sigma_1}) \in \mathscr{C}^+(\boldsymbol{\tau}_1) \times \mathscr{C}^+(\boldsymbol{\sigma}_1) \mid \text{caus. comp.}\} \times \\
& \{(x^{\tau_2}, x^{\sigma_2}) \in \mathscr{C}^+(\boldsymbol{\tau}_2) \times \mathscr{C}^+(\boldsymbol{\sigma}_2) \mid \text{caus. comp.}\} \\
\cong\ & \mathscr{C}^+(\boldsymbol{\tau}_1 \odot \boldsymbol{\sigma}_1) \times \mathscr{C}^+(\boldsymbol{\tau}_2 \odot \boldsymbol{\sigma}_2)
\end{aligned}$$

using Proposition 7.3.1, then Proposition 8.1.1 along with Lemma 8.1.3, then Proposition 7.3.1 again. All these are order-isos, preserve display maps, and the same reasoning applies to symmetries – so the desired isomorphism follows from Corollary 7.2.13. □

8.1.2 Lifting Maps to Strategies

Besides this bifunctor, we need structural morphisms for the monoidal structure.

We introduce a method to construct them systematically. The idea is that all these morphisms are direct variations of copycat: they play copycat following a certain isomorphism of games. Accordingly, we show how to lift certain maps to strategies.

Renamings. We shall lift more than just isos: "liftable" maps are called *renamings*:

Definition 8.1.5. *A **renaming** from game A to B is a function $f : |A| \to |B|$ satisfying:*

$$\begin{array}{rl}
\text{validity:} & \forall x \in \mathscr{C}(A),\ f x \in \mathscr{C}(B) \\
\text{local injectivity:} & \forall a_1, a_2 \in x \in \mathscr{C}(A),\ f a_1 = f a_2 \ \Rightarrow\ a_1 = a_2 \\
\text{pol-preserving:} & \forall a \in A,\ \mathrm{pol}_B(f a) = \mathrm{pol}_A(a) \\
\text{sym-preserving:} & \forall \theta \in \mathscr{S}(A) \textit{(resp. } \mathscr{S}_+(A), \mathscr{S}_-(A)\textit{)}, \\
& f\,\theta \in \mathscr{S}(B), \textit{(resp. } \mathscr{S}_+(B), \mathscr{S}_-(B)\textit{)} \\
\text{strong-receptivity:} & \forall \theta \in \mathscr{S}(A), \forall f\theta \subseteq^- \varphi \in \mathscr{S}(B),\ \exists! \theta \subseteq^- \theta' \in \mathscr{S}(A)\ f\theta' = \varphi \\
\text{courtesy:} & \forall a_1 \rightarrowtriangle_A a_2,\ (\mathrm{pol}_A(a_1) = + \vee \mathrm{pol}_A(a_2) = -) \ \Rightarrow\ f a_1 \rightarrowtriangle_B f a_2\,.
\end{array}$$

We write **Ren** *for the category of renamings, and $f : A \to B$ for a renaming.*

Any isomorphism of games (*i.e.* an isomorphism preserving all structure) is a renaming. Renamings let us change the display of events of strategies, via the operation:

Definition 8.1.6. *Consider $\boldsymbol{\sigma} : A \vdash B$, with renamings $f : A^\perp \to A'^\perp$, $g : B \to B'$. Then, the **redisplayed** strategy $g \cdot \boldsymbol{\sigma} \cdot f$ has components the same as $\boldsymbol{\sigma}$, except:*

$$\partial_{g \cdot \boldsymbol{\sigma} \cdot f} = (f \parallel g) \circ \partial_{\boldsymbol{\sigma}}\,.$$

This is still a strategy – note the strategy itself is not really changed, only its display. Here, $f \parallel g$ refers to the monoidal structure of **ESS**, defined with $(f \parallel g)(1, a) = (1, f(a))$ and $(f \parallel g)(2, b) = (2, g(b))$. Note also that we can redisplay on one side of a strategy only, as in $g \cdot \boldsymbol{\sigma}$: this means $g \cdot \boldsymbol{\sigma} \cdot \mathrm{id}_A$ – likewise, $\boldsymbol{\sigma} \cdot f$ means $\mathrm{id}_B \cdot \boldsymbol{\sigma} \cdot f$.

Lifting. We motivated renamings by the *lifting* operation, which is:

Definition 8.1.7. *Consider $f : A \to B$ a renaming.*
*Its **forward lifting** $\mathfrak{w}_f : A \vdash B$ is defined as $\mathfrak{w}_f = f \cdot \mathfrak{w}_A$.*
*Its **backward lifting** $\mathfrak{m}_f : B^\perp \vdash A^\perp$ is defined as $\mathfrak{m}_f = \mathfrak{w}_{A^\perp} \cdot f$.*

This gives a simple definition of a strategy playing copycat alongside a certain map. This is a nice feature of concurrent games: such a construction is ubiquitous also in more traditional game semantics settings, but it is very rarely made explicit.

The fundamental property enjoyed by this construction is the **lifting lemma**:

Lemma 8.1.8. *Consider $\sigma : A \vdash B$, and renamings $f : A^\perp \to A'^\perp$, $g : B \to B'$.*
Then, $g \cdot \sigma \cdot f \approx \mathfrak{w}_g \odot \sigma \odot \mathfrak{m}_f$.

Proof. We show $g \cdot \sigma \approx \mathfrak{w}_g \odot \sigma$ – note $\sigma \cdot f \approx \sigma \odot \mathfrak{m}_f$ is symmetric, and the general case follows since $g \cdot \sigma \cdot f = g \cdot (\sigma \cdot f) = (g \cdot \sigma) \cdot f$ and $\odot$ is associative up to $\approx$.
By definition, $\mathscr{C}^+(\mathfrak{w}_f) = \mathscr{C}^+(\mathfrak{w}_B) = \{x_B \parallel x_B \mid x_B \in \mathscr{C}(B)\}$ (Lemma 6.4.4), with

$$\partial_{\mathfrak{w}_g}(x_B \parallel x_B) = x_B \parallel g(x_B)\,.$$

Thus a pair $(x^\sigma, x_B \parallel x_B)$ is causally compatible for σ and $\mathfrak{w}_f$ iff it is causally compatible for σ and $\mathfrak{w}_B$. We may then reuse the unitor of Proposition 6.4.9 which yields $r_\sigma : \mathscr{C}^+(\sigma \odot \mathfrak{w}_f) \cong \mathscr{C}^+(\sigma)$. Computing the display maps, we have

$$\partial_{g\cdot\sigma}(x^\sigma) = g(x^\sigma_B) = g(x_B)\,, \qquad \partial_{\mathfrak{w}_g \odot \sigma}((x_B \parallel x_B) \odot x^\sigma) = g(x_B)$$

where $x^\sigma_B = x_B$ as the pair $(x^\sigma, x_B \parallel x_B)$ is matching. The same reasoning applies to symmetries using Lemma 7.3.14. By Corollary 7.2.13, we deduce the desired iso. □

We deduce one last convenient property of lifting:

Proposition 8.1.9. *There are two $\sim$-functors:*

$$\mathfrak{w}_- : \mathbf{Ren} \to \mathbf{TCG}\,, \qquad \mathfrak{m}_- : \mathbf{Ren}^{\mathsf{op}} \to \mathbf{TCG}\,,$$

such that if $f : A \to B$ is an iso, then $\mathfrak{w}_f$ and $\mathfrak{m}_{f^\perp}$ are inverses, and $\mathfrak{w}_{f^{-1}} \approx \mathfrak{m}_{f^\perp}$.

Proof. By Lemma 8.1.8 and a direct verification via Corollary 7.2.13. □

Above, we write $f^\perp : A^\perp \to B^\perp$ for $f : A \to B$ defined as the same map as f. Note that if $f : A \to B$ is an iso, then $f^\perp : A^\perp \to B^\perp$ is an iso hence a renaming.

8.1.3 Compact Closed Structure

We may now directly apply this for the symmetric monoidal structure.

Symmetric monoidal structure. We need structural isomorphisms for **TCG**.

First, notice that **Ren** is a symmetric monoidal category. Indeed, for $f : A \to B$ and $f' : A' \to B'$ renamings, then $f \parallel f' : A \parallel A' \to B \parallel B'$ is a renaming. Furthermore, there are obvious isomorphisms of tcgs, so invertible renamings

$$\begin{array}{rcrcl} \rho_A & : & A \parallel \mathbf{1} & \to & A \\ \lambda_A & : & \mathbf{1} \parallel A & \to & A \\ \alpha_{A,B,C} & : & (A \parallel B) \parallel C & \to & A \parallel (B \parallel C) \\ s_{A,B} & : & A \parallel B & \to & B \parallel A \end{array}$$

natural in A, B, C and satisfying the coherence laws of a symmetric monoidal category.

From this and the lifting operation introduced above, we prove:

Proposition 8.1.10. *Consider the $\sim$-bifunctor $(- \parallel -)$ along with the empty tcg $\mathbf{1}$, and*

$$\begin{array}{rcrcl} \boldsymbol{\rho}_A & : & A \parallel \mathbf{1} & \to & A \\ \boldsymbol{\lambda}_A & : & \mathbf{1} \parallel A & \to & A \\ \boldsymbol{\alpha}_{A,B,C} & : & (A \parallel B) \parallel C & \to & A \parallel (B \parallel C) \\ \boldsymbol{s}_{A,B} & : & A \parallel B & \to & B \parallel A \end{array}$$

defined with $\boldsymbol{\rho}_A = \mathfrak{C}_{\rho_A}, \boldsymbol{\lambda}_A = \mathfrak{C}_{\lambda_A}, \boldsymbol{\alpha}_{A,B,C} = \mathfrak{C}_{\alpha_{A,B,C}}$ and $\boldsymbol{s}_{A,B} = \mathfrak{C}_{s_{A,B}}$.
This data makes **TCG** *a symmetric monoidal $\sim$-category.*

Proof. The bifunctor is constructed in Proposition 8.1.4. The coherence laws follow from the symmetric monoidal structure of **Ren** along with Proposition 8.1.9.

It remains to establish naturality, which is now relatively direct. We detail naturality of $\boldsymbol{s}_{A,B}$ for illustration purposes: take $\boldsymbol{\sigma} : A \vdash A'$ and $\boldsymbol{\tau} : B \vdash B'$. Observe

$$s_{A',B'} \cdot (\boldsymbol{\sigma} \parallel \boldsymbol{\tau}) \quad \approx \quad (\boldsymbol{\tau} \parallel \boldsymbol{\sigma}) \cdot s^{\perp}_{A,B} \tag{8.1}$$

as follows directly by Corollary 7.2.13 applied to the obvious isomorphisms $\mathscr{C}^+(\boldsymbol{\sigma} \parallel \boldsymbol{\tau}) \cong \mathscr{C}^+(\boldsymbol{\tau} \parallel \boldsymbol{\sigma})$ and $\mathscr{S}^+(\boldsymbol{\sigma} \parallel \boldsymbol{\tau}) \cong \mathscr{S}^+(\boldsymbol{\tau} \parallel \boldsymbol{\sigma})$. From that we calculate:

$$\begin{array}{rcl} \boldsymbol{s}_{A',B'} \odot (\boldsymbol{\sigma} \parallel \boldsymbol{\tau}) & \approx & s_{A',B'} \cdot (\boldsymbol{\sigma} \parallel \boldsymbol{\tau}) \\ & \approx & (\boldsymbol{\tau} \parallel \boldsymbol{\sigma}) \cdot s^{\perp}_{A,B} \\ & \approx & (\boldsymbol{\tau} \parallel \boldsymbol{\sigma}) \odot \mathfrak{B}_{s^{\perp}_{A,B}} \\ & \approx & (\boldsymbol{\tau} \parallel \boldsymbol{\sigma}) \odot \mathfrak{C}_{s_{B,A}} \\ & \approx & (\boldsymbol{\tau} \parallel \boldsymbol{\sigma}) \odot \boldsymbol{s}_{B,A} \end{array}$$

using Lemma 8.1.8, equation (8.1), Lemma 8.1.8 again, Proposition 8.1.9, and definition of $\boldsymbol{s}_{A,B}$. All other naturality squares follow using the same route. □

Compact closed structure. Briefly recall that a compact closed category $\mathcal{C}$ is a symmetric monoidal category where each object comes with a *dual* A^* and two morphisms

$$\epsilon_A : A^* \otimes A \to 1 \qquad \eta_A : 1 \to A \otimes A^*,$$

the *unit* and the *co-unit*, satisfying two equations (see Theorem 8.1.12). A compact closed category is automatically symmetric monoidal closed: the monoidal closure can be defined as $A \multimap B = A^* \otimes B$. While this structure is not adequate to directly interpret a programming language, it is the fundamental structure expected of an unpolarized category of games, and it will be helpful for further constructions.

In **TCG**, the dual of a game A is as expected $A^\perp$. The unit and co-unit are:

Definition 8.1.11. *Consider A a tcg. The **co-unit** $\epsilon_A : A^\perp \parallel A \vdash \mathbf{1}$ and **unit** $\eta_A : \mathbf{1} \vdash A \parallel A^\perp$, formed with ess respectively $\mathfrak{C}_{A^\perp}$ and $\mathfrak{C}_A$ with display maps respectively*

$$\begin{array}{rcccl} \partial_{\epsilon_A} & : & |\mathfrak{C}_{A^\perp}| & \to & |A^\perp \parallel A \vdash \mathbf{1}| \\ & & (1,a) & \mapsto & (1,(1,a)) \\ & & (2,a) & \mapsto & (1,(2,a)), \end{array} \qquad \begin{array}{rcccl} \partial_{\eta_A} & : & |\mathfrak{C}_A| & \to & |\mathbf{1} \vdash A \parallel A^\perp| \\ & & (1,a) & \mapsto & (2,(1,a)) \\ & & (2,a) & \mapsto & (2,(2,a)) \end{array}$$

are valid strategies.

Altogether, this provides the data for the following theorem:

Theorem 8.1.12. **TCG** *is a compact closed $\sim$-category.*

Proof. It remains to prove the two duality conditions:

$$\begin{array}{rcl} \mathfrak{C}_A & \approx & \rho_A \odot (\mathfrak{C}_A \parallel \epsilon_A) \odot \alpha_{A,A^\perp,A} \odot (\eta_A \parallel \mathfrak{C}_A) \odot \lambda_A^{-1} \\ \mathfrak{C}_{A^\perp} & \approx & \lambda_{A^\perp} \odot (\epsilon_A \parallel \mathfrak{C}_{A^\perp}) \odot \alpha^{-1}_{A^\perp,A,A^\perp} \odot (\mathfrak{C}_{A^\perp} \parallel \eta_{A^\perp}) \odot \rho^{-1}_{A^\perp} \end{array}$$

which follow from Corollary 7.2.13 using Proposition 7.3.1, Lemmas 6.4.4 and 7.3.14 and the direct verification that all matching pairs involved are causally compatible. □

As a compact closed $\sim$-category, **TCG** is a model of MLL and hence of the linear λ-calculus. But not more: like Conway games, **TCG** has neither products nor coproducts (as analysed by Melliès [Melliès, 2004b]). It also does not support the resource modality ! of Definition 7.1.26, which was defined on *negative* tcgs.

This missing structure may be recovered by switching to a *polarized* category of games: here *negative* games, where Opponent always starts (see Definition 7.1.24).

8.2 Negative Winning Games

As mentioned earlier, in order to obtain the categorical structure required to interpret programming language, one must commit to either *negative* games (which have products but no coproducts); or *positive* games (which have coproducts but no products). A pleasant feature of game semantics is that this choice reflects the evaluation order that one wishes to represent: call-by-name languages are naturally interpreted in categories of negative games, while call-by-value languages are interpreted in categories of positive games [Honda and Yoshida, 1999]. Here, we focus on negative games.

Winning games. We also introduce another condition: *winning* games and strategies. This mechanism, adapted from [Melliès, 2005, Melliès and Tabareau, 2007], has two purposes: *(1)* it compensates the inherently affine nature of game semantics, tightening the connections with relational-like semantics (see Section 10.4); and *(2)* it replaces the traditional but often bulky notion of *well-bracketing* in banning control operators like call/cc. We wire this into our ambient game semantics. We call this *winning* as it sets as common objective for both players to reach a winning position with as many Player as Opponent moves – and where intuitively speaking, all questions have an answer.

Relative Seely categories. Here, a reader familiar with categorical models of linear logic might expect us to construct a *Seely category*, a categorical model of intuitionistic linear logic [Melliès, 2009]. One can indeed construct a Seely category of thin concurrent games [Castellan and Clairambault, 2021]. But that construction is slightly unpleasant due to types such as $o \multimap (o \otimes o)$, *i.e.* tensors appearing on the right hand side of an arrow, which do not appear in our languages[1]. Worse, this construction breaks the clean connection with the relational model. So we shall aim for a slightly weaker categorical structure called a *relative Seely category*, which slightly restricts the monoidal closure, banning in particular tensors on the right of an arrow. Like Seely categories, relative Seely categories still enjoy that the Kleisli category for ! is cartesian closed.

8.2.1 Relative Seely categories

Though that is a detour in our game-theoretic narrative, we must first define what we mean by relative Seely categories. Relative Seely categories model the fragment of intuitionistic linear logic with formulas those generated by the grammar

$$\begin{array}{rcl} S, T & ::= & \top \mid S \& T \mid A \multimap S \\ A, B & ::= & 1 \mid A \otimes B \mid S \mid !S \end{array}$$

splitting formulas into *strict* S, T and *general* A, B formulas. Note that this covers all formulas involved in Girard's call-by-name translation for types.

In the following definition, we make use of the standard notions of relative adjunctions and relative comonads. For the reader unfamiliar with these, the definition also contains explicit data. For full details, see Appendix A.

Definition 8.2.1. *A **relative Seely category** is a symmetric monoidal category $(\mathcal{C}, \otimes, 1)$ equipped with a full subcategory $\mathcal{C}_s$ together with the following data and axioms:*

- *$\mathcal{C}_s$ has finite products $(\&, \top)$ preserved by the inclusion functor $J : \mathcal{C}_s \hookrightarrow \mathcal{C}$.*
- *For every $B \in \mathcal{C}$ there is a functor $B \multimap - : \mathcal{C}_s \to \mathcal{C}_s$, such that there is*

$$\Lambda(-) : \mathcal{C}(A \otimes B, S) \simeq \mathcal{C}(A, B \multimap S).$$

a bijection natural in $A \in \mathcal{C}$ and $S \in \mathcal{C}_s$.

[1] This triggers a duplication of the o on the left with conflict between the two copies. Monoidal closure works, but with a cumbersome technical overhead that we prefer to avoid here. Note that *products* on the right hand side of an arrow are fine; indeed we shall eventually form a cartesian closed Kleisli category.

- *There is a $\mathbf{J}$-relative comonad $! : \mathcal{C}_s \to \mathcal{C}$: we have, for every $S \in \mathcal{C}_s$, an object $!S \in \mathcal{C}$ and a **dereliction** morphism $\mathsf{der}_S : !S \to S$, and for every $\sigma : !S \to T$, a **promotion** $\sigma^! : !S \to !T$, subject to three axioms [Altenkirch et al., 2010]:*

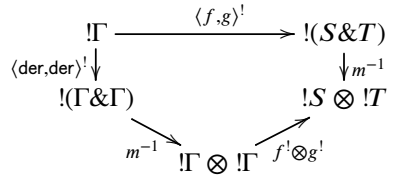
$$\begin{array}{rcll} \mathsf{der}_T \circ \sigma^! & = & \sigma & (\sigma : !S \to T) \\ \mathsf{der}^!_S & = & \mathrm{id}_{!S} & (S \in \mathcal{C}_s) \\ (\tau \circ \sigma^!)^! & = & \tau^! \circ \sigma^! & (\sigma : !S \to T, \tau : T \to U)\,, \end{array}$$

  which make $! : \mathcal{C}_s \to \mathcal{C}$ a functor, via $!\sigma = (\sigma \circ \mathsf{der}_S)^!$ for $\sigma : S \to T$.

- *The functor $! : \mathcal{C}_s \to \mathcal{C}$ is symmetric strong monoidal $(\mathcal{C}_s, \&, \top) \to (\mathcal{C}, \otimes, 1)$, so*

$$m_0 : 1 \to !\top \qquad\qquad m_{S,T} : !S \otimes !T \to !(S\&T)$$

  are natural isos, additional compatible with promotion: the diagram



  commutes for all $\Gamma, S, T \in \mathcal{C}_s$, $f \in \mathcal{C}(!\Gamma, S), g \in \mathcal{C}(!\Gamma, T)$.

Note any Seely category is canonically a relative Seely category with $\mathcal{C} = \mathcal{C}_s$. For any relative Seely category, the Kleisli category associated with !, denoted $\mathcal{C}_!$, is cartesian closed: it has objects those of $\mathcal{C}_s$, and $\mathcal{C}_!(S,T) = \mathcal{C}(!S, T)$.

Lemma 8.2.2. *For a relative Seely category $\mathcal{C}$, the Kleisli category $\mathcal{C}_!$ is cartesian closed with finite products given as in $\mathcal{C}_s$, and function space $S \Rightarrow T = !S \multimap T$.*

Proof. The proof is essentially as for Seely categories; see Appendix A.1. □

For **TCG** these definitions must of course be taken in $\sim$-categorical form: as before this means that all operations preserve $\sim$, and all conditions hold up to $\sim$; yielding a notion of **relative Seely $\sim$-category**. We omit the straightforward adaptation.

We now start building the concrete relative Seely $\sim$-category **NTCG**.

8.2.2 Playing Board Games

Objects of **NTCG** are certain negative tcgs, equipped with an additional component κ, a so-called *payoff function* expressing the *winning conditions*. More precisely, κ records which configurations are complete and assigns responsibility of non-completeness to one of the players otherwise. The resulting object will be the main notion of game used in the rest of this monograph, so we need a distinctive name that remains "gamey":

Boards. We now give the full definition of the objects of **NTCG**:

Definition 8.2.3. *A **board** is a tcg A along with $\kappa_A : \mathscr{C}(A) \to \{-1, 0, +1\}$ a **payoff function**, such that this data satisfies the following conditions:*

invariant:	*for all $\theta : x \cong_A y$, we have $\kappa_A(x) = \kappa_A(y)$,*
race-free:	*for all $a \frown_A a'$, we have $\mathrm{pol}_A(a) = \mathrm{pol}_A(a')$.*
forestial:	*for all $a_1, a_2, a \in A$, if $a_1, a_2 \leq_A a$, then $a_1 \leq_A a_2$ or $a_2 \leq_A a_1$,*
alternating:	*for all $a_1, a_2 \in A$, if $a_1 \rightarrowtriangle_A a_2$, then $\mathrm{pol}_A(a_1) \neq \mathrm{pol}_A(a_2)$,*

*A **$-$-board** is additionally negative, and must also satisfy:*

$$\text{initialized:} \quad \kappa_A(\emptyset) \geq 0\,.$$

*Finally, a $-$-board A is **strict** if $\kappa_A(\emptyset) = 1$ and all its initial moves are in pairwise conflict. It is **well-opened** if it is strict with exactly one initial move.*

The payoff function κ_A assigns a value to each configuration. Configurations with payoff 0 are called complete: they correspond to *terminated* executions, which have reached an adequate stopping point where in particular all calls have adequately returned. Otherwise, κ_A assigns a responsibility for why a configuration is non-complete. If $\kappa_A(x) = -1$ then Player is responsible, otherwise it is Opponent.

Basic $-$-boards. It seems natural to introduce first the boards for PCF base types.

Recall that in Figures 7.7, 7.8 and 7.9 we introduced tcgs U, B and N respectively for the unit type, booleans and natural numbers. These three tcgs are turned into boards by adjoining them a payoff function which, for all three, is defined similarly by

$$\begin{aligned} \kappa(\emptyset) &= 1 \\ \kappa(\{\mathbf{q}^-\}) &= -1 \\ \kappa(\{\mathbf{q}^-, a^+\}) &= 0 \end{aligned}$$

which is exhaustive as maximal configurations have only two elements.

We keep the notations U, B and N for these boards. They are *strict*: the payoff function essentially *forces* Opponent to ask the initial question. In these three basic boards, maximal configurations have null payoff – as mentioned earlier, we call those *complete*:

Definition 8.2.4. *Consider A a board, and $x \in \mathscr{C}(A)$.*
*We say that x is **complete** iff $\kappa_A(x) = 0$, and write $x \in \mathscr{C}^0(A)$.*

Complete configurations are intuitively those where every call is answered, in that they reflect the complete plays of traditional game semantics (see Definition 3.3.6). They shall also inform the link with relational semantics: this will be detailed in Section 10.4, but note already that $\mathscr{C}^0(\mathsf{U})$, $\mathscr{C}^0(\mathsf{B})$ and $\mathscr{C}^0(\mathsf{N})$ are in bijection with the usual relational model interpretation of the corresponding types.

The other basic $-$-boards are the units. In the presence of the payoff function the empty tcg $\mathbf{1}$ splits into two units, reflecting the units of multiplicative and additive conjunctions in linear logic: the **top** $\top$ has $\kappa_\top(\emptyset) = 1$, while the **one** $\mathbf{1}$ has $\kappa_{\mathbf{1}}(\emptyset) = 0$ – we overload the notation for the empty tcg $\mathbf{1}$, which should not cause any confusion.

$\otimes$	-1	0	$+1$
-1	-1	-1	-1
0	-1	0	$+1$
$+1$	-1	$+1$	$+1$

$\parr$	-1	0	$+1$
-1	-1	-1	$+1$
0	-1	0	$+1$
$+1$	$+1$	$+1$	$+1$

Figure 8.1: Payoff for $\otimes$ and $\parr$

Dual, tensor and par. First, the **dual** (Definition 7.1.23) extends with payoff via $\kappa_{A^\perp}(x) = -\kappa_A(x)$ as expected. Of course, the dual does not preserve $-$-boards.

Like the units, *parallel composition* splits into two:

Definition 8.2.5. *Consider A and B two boards.*
*Their **tensor** $A \otimes B$ and their **par** $A \parr B$ are $A \parallel B$ enriched with:*

$$\kappa_{A\otimes B}(x_A \parallel x_B) = \kappa_A(x_A) \otimes \kappa_B(x_B)\,, \qquad \kappa_{A\parr B}(x_A \parallel x_B) = \kappa_A(x_A) \parr \kappa_B(x_B)$$

with the operations $\otimes$ and $\parr$ defined on $\{-1, 0, +1\}$ in Figure 8.1.

The tensor of two $-$-boards is still a $-$-board, though tensor does not preserve *strict* $-$-boards. The par also preserves $-$-boards, but we shall not use it on $-$-boards: if A and B are $-$-boards, let us use $A \vdash B$ to denote the board $A^\perp \parr B$ used to define the strategies *from A to B*. Observe that even if A and B are $-$-boards, $A \vdash B$ is not.

The with. We only consider the additive conjunction of linear logic: the with. But in order to define it, we must first define a new operation of ess and tcgs.

Definition 8.2.6. *Let A_1 and A_2 be two tcgs.*
*Then, we define their **sum** $A_1 + A_2$ as comprising the components:*

$$\begin{array}{rrcl}
\text{events:} & |A_1 \parallel A_2| & = & |A_1| + |A_2| \\
\text{causality:} & (i,a) \leq_{A_1\parallel A_2} (j,a') & \Leftrightarrow & i = j \;\&\; a \leq_{A_i} a' \\
\text{conflict:} & (i,a) \#_{A_1\parallel A_2} (j,a') & \Leftrightarrow & i \neq j \;\vee\; a \#_{E_i} a'\,, \\
\text{symmetry:} & \theta \in \mathcal{S}(A_1 \parallel A_2) & \Leftrightarrow & \exists \theta_i \in \mathcal{S}(A_i), \theta = \theta_1 \parallel \theta_2\,, \\
\text{positive symmetries:} & \theta_1 \parallel \theta_2 \in \mathcal{S}_+(A_1 + A_2) & \Leftrightarrow & \theta_1 \in \mathcal{S}_+(A_1) \,\&\, \theta_2 \in \mathcal{S}_+(A_2) \\
\text{negative symmetries:} & \theta_1 \parallel \theta_2 \in \mathcal{S}_-(A_1 + A_2) & \Leftrightarrow & \theta_1 \in \mathcal{S}_-(A_1) \,\&\, \theta_2 \in \mathcal{S}_-(A_2)\,.
\end{array}$$

where, necessarily, one of θ_1 or θ_2 must be empty.

Ignoring the positive and negative symmetries, this also yields an operation $+$ on plain event structures with symmetry that we shall use later on.

If A, B are tcgs and $x_A \in \mathscr{C}(A)$, we write $(1, x_A) \in \mathscr{C}(A + B)$ as a shorthand for $\{1\} \times x_A$ and likewise for $(2, x_B) = \{2\} \times x_B \in \mathscr{C}(A+B)$ for $x_B \in \mathscr{C}(B)$. Note that all configurations of $A + B$ have the form $(1, x_A)$ for $x_A \in \mathscr{C}(A)$ or $(2, x_B)$ for $x_B \in \mathscr{C}(B)$. For non-empty configurations, this decomposition is *unique*. We shall also use the corresponding notations for symmetries, with *e.g.* $(1, \theta_A) : (1, x_A) \cong_{A+B} (1, y_A)$ for $\theta_A : x_A \cong_A y_A$ comprising all $((1, a), (1, a'))$ for $(a, a') \in \theta_A$.

This sum operation yields the *with* operation on strict boards:

Definition 8.2.7. *Consider A and B two strict $-$-boards.*
*Then, their **with** $A\&B$ is the strict $-$-board with tcg the sum $A + B$ and*

$$\kappa_{A\&B}(1, x_A) = \kappa_A(x_A), \qquad \kappa_{A\&B}(2, x_B) = \kappa_B(x_B),$$

for non-empty configurations and $\kappa_{A\&B}(\emptyset) = 1$.

As we will see, this construction will give a cartesian product in the subcategory of strict $-$-boards. It can also be applied to non-strict $-$-boards, but then it is not a product: if one of the A_i is not strict then the corresponding projection does not respect payoff (in the sense of Definition 8.2.14), because we have set $\kappa_{A_1\&A_2}(\emptyset) = 1$. On the other hand having $\kappa_{A_1\&A_2}(\emptyset) = 0$ breaks the correspondence with the relational model, since the empty configuration does not correspond in a canonical way to one of the components.

In the sequel, we should use the obvious n-ary generalization of the product, with respect to which any strict $-$-board decomposes into a with of well-opened $-$-boards:

Lemma 8.2.8. *Consider A a strict $-$-board. Then there is a family $(A_i)_{i\in I}$ of well-opened $-$-boards, unique up to isomorphism, such that $A \cong \&_{i\in I} A_i$.*

Proof. Straightforward. □

We need notations for configurations of this board. Writing $\mathscr{C}^{\neq\emptyset}(A)$ (resp. $\mathcal{S}^{\neq\emptyset}(A)$) for the set of non-empty configurations (resp. symmetries) of A, we observe:

Lemma 8.2.9. *Consider $(A_i)_{i\in I}$ a family of well-opened $-$-boards. Then there are*

$$\begin{array}{rcl}
\mathscr{C}^{\neq\emptyset}(\&_{i\in I} A_i) & \cong & \sum_{i\in I} \mathscr{C}^{\neq\emptyset}(A_i) \\
\mathcal{S}^{\neq\emptyset}(\&_{i\in I} A_i) & \cong & \sum_{i\in I} \mathcal{S}^{\neq\emptyset}(A_i) \\
\mathcal{S}^{\neq\emptyset}_{-}(\&_{i\in I} A_i) & \cong & \sum_{i\in I} \mathcal{S}^{\neq\emptyset}_{-}(A_i) \\
\mathcal{S}^{\neq\emptyset}_{+}(\&_{i\in I} A_i) & \cong & \sum_{i\in I} \mathcal{S}^{\neq\emptyset}_{+}(A_i)
\end{array}$$

order-isos commuting with dom *and* cod.

Proof. Straightforward. □

For $x \in \mathscr{C}^{\neq\emptyset}(A_i)$ (resp. $\theta \in \mathcal{S}^{\neq\emptyset}(A_i)$) we write $(i, x) \in \mathscr{C}^{\neq\emptyset}(\&_{i\in I} A_i)$ (resp. $(i, \theta) \in \mathcal{S}^{\neq\emptyset}(\&_{i\in I} A_i)$. The final linear connective we shall consider is the linear arrow:

Linear arrow. First, we define it in the case the rhs board is well-opened:

Definition 8.2.10. *Consider A a $-$-board and B a well-opened $-$-board.*
Then, $A \multimap B$ has tcg as defined in Definition 7.1.25, and payoff function:

$$\kappa_{A\multimap B}(x_A \parallel x_B) = \kappa_{A\vdash B}(x_A \parallel x_B) = \kappa_{A^\perp}(x_A) \parr \kappa_B(x_B).$$

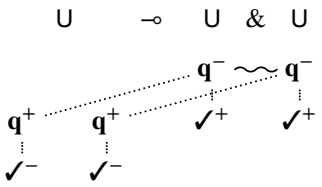


Figure 8.2: The board U ⊸ (U&U)

The definition of relative Seely categories requires us to define $A \multimap B$ not only when B is well-opened (which has no particular status in the definition of relative Seely categories), but when it is strict. In that case, $A \multimap B$ may be defined directly via the decomposition of Lemma 8.2.8, as done in the following definition:

Definition 8.2.11. *Consider A a $-$-board, and B a strict $-$-board, with $B \cong \&_{i\in I} B_i$. Then, we define $A \multimap B = \&_{i\in I} A \multimap B_i$.*

The construction is illustrated in Figure 8.2.2, in which we omit payoff. The construction duplicates the argument arena, with one copy for each initial move of B. In that, the reader familiar with Hyland-Ong games – or the attentive reader of Part I) will recognize its arrow arena construction (modulo the conflict, see *e.g.* Figure 4.1).

It will be useful to have the analogue of Lemma 8.2.9 for the linear arrow:

Lemma 8.2.12. *Consider A, B $-$-boards with B strict. Then, there are:*

$$\begin{array}{rcl}
\mathscr{C}^{\neq\emptyset}(A \multimap B) & \cong & \mathscr{C}(A) \times \mathscr{C}^{\neq\emptyset}(B) \\
\mathcal{S}^{\neq\emptyset}(A \multimap B) & \cong & \mathcal{S}(A) \times \mathcal{S}^{\neq\emptyset}(B) \\
\mathcal{S}^{\neq\emptyset}_{-}(A \multimap B) & \cong & \mathcal{S}_{+}(A) \times \mathcal{S}^{\neq\emptyset}_{-}(B) \\
\mathcal{S}^{\neq\emptyset}_{+}(A \multimap B) & \cong & \mathcal{S}_{-}(A) \times \mathcal{S}^{\neq\emptyset}_{+}(B)
\end{array}$$

order-isos commuting with dom *and* cod.

Proof. If B is well-opened, then $\mathscr{C}^{\neq\emptyset}(A \multimap B) \cong \mathscr{C}(A) \times \mathscr{C}^{\neq\emptyset}(B)$ is obvious. Now if $B \cong \sum_{i\in I} B_i$ via Lemma 8.2.8, we compose bijections:

$$\begin{array}{rcl}
\mathscr{C}^{\neq\emptyset}(A \multimap B) & \cong & \displaystyle\sum_{i\in I} \mathscr{C}^{\neq\emptyset}(A \multimap B_i) \\
& \cong & \displaystyle\sum_{i\in I} \mathscr{C}(A) \times \mathscr{C}^{\neq\emptyset}(B_i) \\
& \cong & \displaystyle \mathscr{C}(A) \times (\sum_{i\in I} \mathscr{C}^{\neq\emptyset}(B_i)) \\
& \cong & \mathscr{C}(A) \times \mathscr{C}^{\neq\emptyset}(B)
\end{array}$$

using Lemma 8.2.9, the well-opened case, distributivity, followed by Lemma 8.2.9 again. The same construction applies to symmetries in a compatible way. □

Following this lemma, we adopt the convention that for each $x_A \in \mathscr{C}(A)$ and $x_B \in \mathscr{C}^{\neq\emptyset}(B)$, $x_A \multimap x_B \in \mathscr{C}^{\neq\emptyset}(A \multimap B)$ denotes the corresponding configuration.

Bang. The next definition enriches Definition 7.1.26 with payoff:

Definition 8.2.13. *Consider A a $-$-board.*
Then, $!A$ has tcg as in Definition 7.1.26 enriched with:

$$\begin{array}{rcll} \kappa_{!A}(\|_{i\in I}\ x_i) & = & \bigotimes_{i\in I} \kappa_A(x_i) & (I \subseteq \mathbb{N},\ \forall i \in I,\ x_i \neq \emptyset) \\ \kappa_{!A}(\emptyset) & = & 0 & \end{array}$$

where $\|_{i\in I}\ x_i = \biguplus_{i\in I}(\{i\} \times x_i)$.

So the payoff for the bang acts as a n-ary tensor for the non-empty components, while the empty configuration gets assigned payoff zero. This means that it is considered *complete*: a complete execution might not initiate any of the copies; but a copy initiated must be brought to a complete state if the whole computation is to be complete.

A pleasing consequence of this definition is that if A is strict, then the symmetry classes of complete configurations of $!A$ are in one-to-one correspondence with finite multisets of symmetry classes of complete configurations of A, informing the link with the relational model – again, see Section 10.4 for more details.

8.2.3 Negative Winning Strategies

Winning strategies. First, we define what it means for a strategy to be well-behaved with respect to payoff. Recall that this has two effects: it compensates for the inherently affine nature of strategies, and it also replaces the traditional well-bracketing condition.

Intuitively, a strategy is *winning* when its stopping points are complete, *or* the incompleteness can be attributed – via the payoff function – to Opponent:

Definition 8.2.14. *Consider A a board, and $\sigma : A$ a strategy. We define conditions:*

$$\begin{array}{rl} \text{negative:} & \textit{for all } s \in \sigma \textit{ minimal for } \leq_\sigma\textit{, we have } \mathrm{pol}_A(\partial_\sigma\, s) = -, \\ \text{winning:} & \textit{for all } x^\sigma \in \mathscr{C}^+(\sigma)\textit{, } \kappa_A(\partial_\sigma\, x^\sigma) \geq 0. \end{array}$$

Negative means that Opponent always starts, also in strategies. *Winning* forces σ to be strictly linear rather than affine, *i.e.* it *must* investigate arguments not marked with a !. For instance, the strategy displayed in Figure 8.3 is non-winning: as U is strict, the fact that the strategy does not investigate its argument is punished by a payoff of -1 on the left, yielding a global payoff of $-1 \parr 0 = -1$ for the maximal configuration.

This definition also ensures a measure of well-bracketing, though it may not be obvious to the reader. Conceptually, Melliès and Tabareau's insight is that well-bracketing is a linearity constraint [Melliès and Tabareau, 2007]: in a well-bracketed completed (in the sense that the initial question receives an answer) execution, each question receives exactly one answer. Still, it might be helpful to examine a concrete example of how typically non-well-bracketed behaviour in the traditional sense is banned. Such an

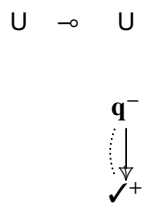


Figure 8.3: An affine non-winning strategy

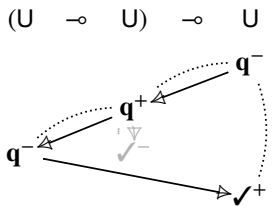


Figure 8.4: A non-winning strategy

example appears in Figure 8.4: the configuration without the gray event is +-covered. Its payoff is $(-1 \otimes +1) \parr 0 = -1$, where the -1 is the price of Player omitting to answer the "pending" left-most question. But this is actually quite subtle and one should not attempt to match this notion of well-bracketing with the traditional ones. In particular, the same causal pattern may very well appear in a winning strategy: for instance, we invite the reader to verify that the strategy in Figure 7.6 (on the left) is winning.

We obtain a $\sim$-category of $-$-boards and negative winning strategies:

Proposition 8.2.15. *There is* **NTCG***, a $\sim$-category with $-$-boards as objects, negative winning strategies on $A \vdash B$ as morphisms from A to B, and $\approx$ as equivalence.*

Proof. Copycat. Negative is immediate by inspection of Definition 6.4.1. For *winning*, consider $x_A \parallel x_A \in \mathscr{C}^+(\mathbb{Q}_A)$ exploiting Lemma 6.4.4. Then, $\kappa_{A \vdash A}(x_A \parallel x_A) = -p \parr p$ for $p = \kappa_A(x_A)$, and for all $p \in \{-1, 0, +1\}$, $-p \parr p \geq 0$ as required.

Composition. For *negative*, assume *w.l.o.g.* that there is $p \in \tau \odot \sigma$ minimal positive. In particular, $x^\tau \odot x^\sigma = \{p\} \in \mathscr{C}^+(\tau \odot \sigma)$. Now p cannot occur in C, as minimal events in C are negative, so it must occur in A. That means $x^\sigma \in \mathscr{C}^+(\sigma)$ is non-empty. As σ is negative, there is a minimal (necessarily) negative event in x^σ, which must occur in B. As x^σ and x^τ are matching, x^τ is non-empty as well. By *negative*, there is a minimal (necessarily) negative event in x^τ, which must occur in C. But as $\partial_{\tau \odot \sigma}(x^\tau \odot x^\sigma) = x^\sigma_A \parallel x^\tau_C$, $x^\tau \odot x^\sigma$ must have an event occurring in C, absurd.

For *winning*, consider $x^\tau \odot x^\sigma \in \mathscr{C}^+(\tau \odot \sigma)$, with $\partial_{\tau \odot \sigma}(x^\tau \odot x^\sigma) = x^\sigma_A \parallel x^\tau_C$. If $\kappa_A(x^\sigma_A) = -1$, then $\kappa_{A \vdash C}(x^\sigma_A \parallel x^\tau_C) = 1$ and we are done. Likewise if $\kappa_C(x^\tau_C) = 1$, we are done, so assume neither. Now if $\kappa_A(x^\sigma_A) = 1$, as σ is winning we must have $\kappa_B(x^\sigma_B) = \kappa_B(x^\tau_B) = 1$ as well; and as τ is winning this entails $\kappa_C(x^\tau_C) = 1$, absurd. Symmetrically, $\kappa_C(x^\tau_C) = -1$ entails $\kappa_A(x^\sigma_A) = -1$, absurd. The only case left has $\kappa_A(x^\sigma_A) = \kappa_C(x^\tau_C) = 0$, so that $\kappa_{A \vdash C}(x^\sigma_A \parallel x^\tau_C) = 0 \geq 0$ as required. □

The properties of the equivalence relation $\approx$ packaged in Proposition 7.4.15 are undisturbed by these now conditions and apply transparently.

8.2.4 Symmetric Monoidal Structure

We now extend the symmetric monoidal structure.

Tensor. First, we extend the tensor of strategies:

Proposition 8.2.16. *The tensor operation of Definition 8.2.5 yields a $\sim$-bifunctor:*

$$(- \otimes -) : \mathbf{NTCG} \times \mathbf{NTCG} \to \mathbf{NTCG}\,.$$

Proof. Consider $\boldsymbol{\sigma} : A \vdash B, \boldsymbol{\tau} : C \vdash D$, their tensor $\boldsymbol{\sigma} \otimes \boldsymbol{\tau}$ is defined on strategies as in Proposition 8.1.1. We must check that this preserves *negative* and *winning*.

Negative. Consider $m \in \boldsymbol{\sigma} \parallel \boldsymbol{\tau}$ minimal. Seeking a contradiction, assume m is positive, so $\{m\}$ is $+$-covered. By Proposition 8.1.1, $\{m\}$ may be written as $x^{\sigma} \parallel x^{\tau} \in \mathscr{C}^{+}(\boldsymbol{\sigma} \otimes \boldsymbol{\tau})$. As $\partial_{\sigma\otimes\tau}(m) \in A \otimes C \vdash B \otimes D$, m occurs in one of the four components. Say *w.l.o.g.* that it is in A or B, then x^{σ} must be a singleton. But as $\boldsymbol{\sigma}$ is negative, $\partial_{\boldsymbol{\sigma}}\, x^{\sigma}$ is a singleton whose only event is negative in $A \vdash B$. By the constraints on display maps given by Proposition 8.1.1, it follows that m is negative, contradiction.

Winning. Consider $x^{\sigma} \parallel x^{\tau} \in \mathscr{C}^{+}(\boldsymbol{\sigma} \otimes \boldsymbol{\tau})$. If $\kappa_A(x^{\sigma}_A) = -1$ or $\kappa_C(x^{\tau}_C) = -1$, then we are done. As $\boldsymbol{\sigma}$ and $\boldsymbol{\tau}$ are winning, this directly entails that we cannot have $\kappa_B(x^{\sigma}_B) = -1$ or $\kappa_D(x^{\tau}_D) = -1$ either, so $\kappa_B(x^{\sigma}_B), \kappa_D(x^{\tau}_D) \geq 0$. Now, assume $\kappa_{A\otimes C}(x^{\sigma}_A \parallel x^{\tau}_C) = 1$. If $\kappa_A(x^{\sigma}_A) = 1$, then as $\boldsymbol{\sigma}$ is winning we have $\kappa_B(x^{\sigma}_B) = 1$. But as $\kappa_D(x^{\tau}_D) \geq 0$, $\kappa_{B\otimes D}(x^{\sigma}_B \parallel x^{\tau}_D) = 1$ and we are done. Likewise if $\kappa_C(x^{\tau}_C) = 1$, we may conclude. □

Lifting. The monoidal structure shall be obtained following Section 8.1.2, by lifting maps. However, we must first define those renamings that yield winning strategies.

Definition 8.2.17. *Consider A, B two boards, and $f : A \to B$ a renaming.*
*We say that f is **winning** if for all $x_A \in \mathscr{C}(A)$, $\kappa_B(f(x_A)) \geq \kappa_A(x_A)$.*

As expected, winning renamings preserve winning strategies:

Lemma 8.2.18. *Consider winning renamings $f : A^{\perp} \to A'^{\perp}$ and $g : B \to B'$.*
Then, $g \cdot \boldsymbol{\sigma} \cdot f : A' \vdash B'$ is winning.

Proof. Consider $x^{\sigma} \in \mathscr{C}^{+}(\boldsymbol{\sigma})$. If $\kappa_{A'}(f(x^{\sigma}_A)) = -1$ or $\kappa_{B'}(g(x^{\sigma}_B)) = 1$ then we are done, so assume otherwise. If $\kappa_{A'}(f(x^{\sigma}_A)) = 1$, then $\kappa_{A'^{\perp}}(f(x^{\sigma}_A)) = -1$, but since f is winning, we have $\kappa_{A'^{\perp}}(f(x^{\sigma}_A)) \geq \kappa_{A^{\perp}}(x^{\sigma}_A)$ so $\kappa_{A^{\perp}}(x^{\sigma}_A) = -1$ as well. Now since $\boldsymbol{\sigma}$ is winning, this entails $\kappa_B(x^{\sigma}_B) = 1$. But then $\kappa_{B'}(g(x^{\sigma}_B)) \geq \kappa_B(x^{\sigma}_B)$ as g is winning, so $\kappa_{B'}(g(x^{\sigma}_B)) = 1$, contradiction. Symmetrically, if $\kappa_{B'}(g(x^{\sigma}_B)) = -1$, this entails $\kappa_{A'}(f(x^{\sigma}_A)) = -1$ as $f, \boldsymbol{\sigma}$ and g are winning, contradiction. □

It follows directly that we may lift winning renamings to strategies:

Corollary 8.2.19. *Consider $f : A \to B$ a winning renaming. Then,*

(1) $\mathfrak{w}_f : A \vdash B$ *is a negative, winning strategy,*
(2) $\mathfrak{w}_f : B^{\perp} \vdash A^{\perp}$ *is a negative, winning strategy.*

Proof. Immediate by Lemma 8.2.18. □

And finally, we deduce the symmetric monoidal structure:

Proposition 8.2.20. *Proposition 8.2.16 along with the data of Proposition 8.1.10 makes* **NTCG** *a symmetric monoidal $\sim$-category.*

Proof. Direct from Propositions 8.2.16, 8.1.10 and Corollary 8.2.19. □

8.2.5 Cartesian Product of Strict $-$-Boards

Next, we must define the cartesian structure. Recall from the definition of relative Seely categories (Definition 8.2.1) that the cartesian product is not required to be defined on the whole category, but only on a full subcategory $\mathcal{C}_s$. Here, we will consider the full $\sim$-subcategory $\mathbf{NTCG}_s$ with as objects the *strict* $-$-boards.

Projections. First, we define the projections. First we need renamings:

Lemma 8.2.21. *Consider A, B strict $-$-boards.*
The injections $\iota_A : A^\perp \to (A\&B)^\perp$, $\iota_B : B^\perp \to (A\&B)^\perp$ are winning renamings.

Proof. Immediate verification. □

The above lets us define the projections via lifting:

$$\pi_A = \mathfrak{B}_{\iota_A} : A\&B \vdash A\,, \qquad \pi_B = \mathfrak{B}_{\iota_B} : A\&B \vdash B$$

winning negative strategies by Corollary 8.2.19.

Pairing. As for earlier constructions, we build pairing via a universal construction:

Proposition 8.2.22. *For $-$-boards Γ, A, B, with A, B strict, and negative winning strategies $\sigma : \Gamma \vdash A, \tau : \Gamma \vdash B$, there is a negative winning strategy $\langle \sigma, \tau \rangle : \Gamma \vdash A\&B$, unique up to iso, s.t. there are order-isos:*

$$\begin{array}{rcl} \mathscr{C}^+(\sigma) + \mathscr{C}^+(\tau) & \simeq & \mathscr{C}^+(\langle \sigma, \tau \rangle) \\ \mathscr{S}^+(\sigma) + \mathscr{S}^+(\tau) & \simeq & \mathscr{S}^+(\langle \sigma, \tau \rangle) \end{array}$$

commuting with $\mathsf{dom}, \mathsf{cod}$*, and such that for all $\theta^\sigma \in \mathscr{S}^+(\sigma)$ and $\theta^\tau \in \mathscr{S}^+(\tau)$, we have*

$$\partial_{\langle \sigma,\tau \rangle}(\iota_\sigma(\theta^\sigma)) = \theta^\sigma_\Gamma \parallel (\theta^\sigma_A \parallel \emptyset)\,, \qquad \partial_{\langle \sigma,\tau \rangle}(\iota_\tau(\theta^\tau)) = \theta^\tau_\Gamma \parallel (\emptyset \parallel \theta^\tau_B)$$

with $\iota_\sigma : \mathscr{C}^+(\sigma) \to \mathscr{C}^+(\langle \sigma, \tau \rangle)$ and $\iota_\tau : \mathscr{C}^+(\tau) \to \mathscr{C}^+(\langle \sigma, \tau \rangle)$ the induced injections.
Moreover, $\langle -, - \rangle$ preserves $\approx$.

Proof. Existence. We set $\langle \boldsymbol{\sigma}, \boldsymbol{\tau} \rangle$ as having as ess $\boldsymbol{\sigma} + \boldsymbol{\tau}$, with display maps given by

$$\begin{array}{rcl} \partial_{\langle \boldsymbol{\sigma},\boldsymbol{\tau} \rangle}(\iota_{\boldsymbol{\sigma}}(x^{\boldsymbol{\sigma}})) & = & x^{\boldsymbol{\sigma}}_\Gamma \vdash (1, x^{\boldsymbol{\sigma}}_A) \\ \partial_{\langle \boldsymbol{\sigma},\boldsymbol{\tau} \rangle}(\iota_{\boldsymbol{\tau}}(x^{\boldsymbol{\tau}})) & = & x^{\boldsymbol{\tau}}_\Gamma \vdash (2, x^{\boldsymbol{\tau}}_B) \end{array}$$

where $\partial_{\boldsymbol{\sigma}}(x^{\boldsymbol{\sigma}}) = x^{\boldsymbol{\sigma}}_\Gamma \vdash x^{\boldsymbol{\sigma}}_A$ and $\partial_{\boldsymbol{\tau}}(x^{\boldsymbol{\tau}}) = x^{\boldsymbol{\tau}}_\Gamma \vdash x^{\boldsymbol{\tau}}_B$; and likewise for symmetries. All required verifications are immediate, and this yields a negative winning strategy.

Uniqueness. Direct consequence of Corollary 7.2.13.

Congruence. Direct from the definition. □

We need one last lemma to conclude with the cartesian product:

Lemma 8.2.23. *Consider Γ, A, B $-$-boards with A, B strict, and $\boldsymbol{\alpha} : \Gamma \vdash A\&B$. Then,*

$$\begin{array}{rcl} \mathscr{C}^+(\pi_A \odot \boldsymbol{\alpha}) & \cong & \{x^{\boldsymbol{\alpha}} \in \mathscr{C}^+(\boldsymbol{\alpha}) \mid \partial_{\boldsymbol{\alpha}}\, x^{\boldsymbol{\alpha}} = x^{\boldsymbol{\alpha}}_{\Gamma} \vdash (1, x^{\boldsymbol{\alpha}}_{A})\} \\ \mathscr{S}^+(\pi_A \odot \boldsymbol{\alpha}) & \cong & \{\theta^{\boldsymbol{\alpha}} \in \mathscr{S}^+(\boldsymbol{\alpha}) \mid \partial_{\boldsymbol{\alpha}}\, \theta^{\boldsymbol{\alpha}} = \theta^{\boldsymbol{\alpha}}_{\Gamma} \vdash (1, \theta^{\boldsymbol{\alpha}}_{A})\} \end{array}$$

are order-isomorphisms compatible with display maps – and likewise for π_B.

Proof. A direct adaptation of the proof of Proposition 6.4.9. □

Proposition 8.2.24. *For any two strict $-$-boards A and B, $A\&B$ is a cartesian product of A and B in* **NTCG**. *Moreover, $\top$ is terminal.*

Proof. We prove the universal property of cartesian products.

Existence. For $\boldsymbol{\sigma} : \Gamma \vdash A$, $\boldsymbol{\tau} : \Gamma \vdash B$ we form $\langle \boldsymbol{\sigma}, \boldsymbol{\tau} \rangle$ via Proposition 8.2.22. Then:

$$\begin{array}{rcl} \mathscr{C}^+(\pi_A \odot \langle \boldsymbol{\sigma}, \boldsymbol{\tau} \rangle) & \cong & \{x \in \mathscr{C}^+(\langle \boldsymbol{\sigma}, \boldsymbol{\tau} \rangle) \mid \partial\, x = x_{\Gamma} \vdash (1, x_A)\} \\ & \cong & \mathscr{C}^+(\boldsymbol{\sigma}) \end{array}$$

using Lemma 8.2.23 followed by Proposition 8.2.22, and those order-isos preserve display maps. The same reasoning applies to symmetries, so that $\pi_A \odot \langle \boldsymbol{\sigma}, \boldsymbol{\tau} \rangle \approx \boldsymbol{\sigma}$ by Corollary 7.2.13. Symmetrically, we have $\pi_B \odot \langle \boldsymbol{\sigma}, \boldsymbol{\tau} \rangle \approx \boldsymbol{\tau}$ as well.

Uniqueness. Consider $\boldsymbol{\alpha} : \Gamma \vdash A\&B$ such that $\pi_A \odot \boldsymbol{\alpha} \approx \boldsymbol{\sigma}$ and $\pi_B \odot \boldsymbol{\alpha} \approx \boldsymbol{\tau}$. First,

$$\boldsymbol{\alpha} \quad \cong \quad \langle \pi_A \odot \boldsymbol{\alpha}, \pi_B \odot \boldsymbol{\alpha} \rangle$$

follows from Lemma 8.2.23, uniqueness of Proposition 8.2.22 and immediate verifications. Finally, we have $\langle \pi_A \odot \boldsymbol{\alpha}, \pi_B \odot \boldsymbol{\alpha} \rangle \approx \langle \boldsymbol{\sigma}, \boldsymbol{\tau} \rangle$ as $\langle -, - \rangle$ preserves $\approx$.

Terminal. For any $-$-board Γ, it is immediate that the empty strategy $\boldsymbol{\epsilon} : \Gamma \vdash \top$ is a winning negative strategy. Moreover, given any $\boldsymbol{\alpha} : \Gamma \vdash \top$ negative, any minimal event in $\boldsymbol{\alpha}$ must be negative and hence map to an event in $\top$ – but there are none. □

This concludes the investigation of cartesian products in **NTCG**.

8.2.6 Relative Closure

The final piece of the linear structure is the relative closure (see Definition 8.2.1). The main operation we must define is **currying** – as for the other operations, we characterize it uniquely up to isomorphism via its configurations.

Proposition 8.2.25. *Consider Γ, A, B $-$-boards with B strict.*

For $\boldsymbol{\sigma} : \Gamma \otimes A \vdash B$, there is $\Lambda(\boldsymbol{\sigma}) : \Gamma \vdash A \multimap B$, unique up to iso, s.t. there are

$$\begin{array}{rcccl} \Lambda(-) & : & \mathscr{C}^+(\boldsymbol{\sigma}) & \cong & \mathscr{C}^+(\Lambda(\boldsymbol{\sigma})) \\ \Lambda(-) & : & \mathscr{S}^+(\boldsymbol{\sigma}) & \cong & \mathscr{S}^+(\Lambda(\boldsymbol{\sigma})) \end{array}$$

order-isos commuting with $\mathsf{dom}, \mathsf{cod}$*, and such that for all θ^{σ} non-empty,*

$$\partial_{\Lambda(\boldsymbol{\sigma})}(\Lambda(\theta^{\sigma})) \quad = \quad \theta^{\sigma}_{\Gamma} \vdash \theta^{\sigma}_{A} \multimap \theta^{\sigma}_{B} \tag{8.2}$$

whenever $\partial_{\boldsymbol{\sigma}}(\theta^{\sigma}) = \theta^{\sigma}_{\Gamma} \parallel \theta^{\sigma}_{A} \vdash \theta^{\sigma}_{B}$.

Moreover, $\Lambda(-)$ preserves $\approx$.

Proof. Note that (8.2) relies on Lemmas 7.1.22, 8.2.9 and 8.2.12, via which there are

$$\mathscr{C}(\Gamma \otimes A) \times \mathscr{C}^{\neq\emptyset}(B) \quad \cong \quad \mathscr{C}(\Gamma) \times \mathscr{C}^{\neq\emptyset}(A \multimap B) \tag{8.3}$$

$$\mathscr{S}(\Gamma \otimes A) \times \mathscr{S}^{\neq\emptyset}(B) \quad \cong \quad \mathscr{S}(\Gamma) \times \mathscr{S}^{\neq\emptyset}(A \multimap B) \tag{8.4}$$

order-isos preserving dom and cod – by negativity of $\boldsymbol{\sigma}$, for $\theta^{\boldsymbol{\sigma}} \in \mathscr{S}^{\neq\emptyset}(\boldsymbol{\sigma})$, $\theta^{\boldsymbol{\sigma}}_B \neq \emptyset$.

Existence. We set $\Lambda(\boldsymbol{\sigma})$ with the same ess as $\boldsymbol{\sigma}$, so in particular $\mathscr{C}^+(\boldsymbol{\sigma}) = \mathscr{C}^+(\Lambda(\boldsymbol{\sigma}))$ and $\mathscr{S}^+(\boldsymbol{\sigma}) = \mathscr{S}^+(\Lambda(\boldsymbol{\sigma}))$. It remains to redefine the display map of $\Lambda(\boldsymbol{\sigma})$; but given the equalities above and Lemma 7.1.9, (8.2) may be read as a definition.

Uniqueness. Follows from Corollary 7.2.13.

Congruence. Consider $f : \boldsymbol{\sigma} \approx \boldsymbol{\tau}$ a positive isomorphism. As $\Lambda(-)$ leaves the ess unchanged, it suffices to show that $f : \Lambda(\boldsymbol{\sigma}) \cong \Lambda(\tau)$ is still a positive isomorphism. Now, by hypothesis, we know that for all $x^{\boldsymbol{\sigma}} \in \mathscr{C}(\boldsymbol{\sigma})$, the following square commutes:

$$\begin{array}{ccc} x^{\boldsymbol{\sigma}} & \xrightarrow{\quad f \quad} & f(x^{\boldsymbol{\sigma}}) \\ {\scriptstyle \partial_{\boldsymbol{\sigma}}}\downarrow & & \downarrow{\scriptstyle \partial_{\boldsymbol{\tau}}} \\ x^{\boldsymbol{\sigma}}_\Gamma \parallel x^{\boldsymbol{\sigma}}_A \vdash x^{\boldsymbol{\sigma}}_B & \xrightarrow[\theta_\Gamma \parallel \theta_A \vdash \theta_B]{} & y^{\boldsymbol{\tau}}_\Gamma \parallel y^{\boldsymbol{\tau}}_A \vdash y^{\boldsymbol{\tau}}_B \end{array}$$

where $\theta_\Gamma \parallel \theta_A \vdash \theta_B : x^{\boldsymbol{\sigma}}_\Gamma \parallel x^{\boldsymbol{\sigma}}_A \vdash x^{\boldsymbol{\sigma}}_B \cong^+_{\Gamma\otimes A \vdash B} y^{\boldsymbol{\tau}}_\Gamma \parallel y^{\boldsymbol{\tau}}_A \vdash y^{\boldsymbol{\tau}}_B$ is a positive symmetry.

Now, from Lemma 8.2.12 and definition of $\partial_{\Lambda(\boldsymbol{\sigma})}, \partial_{\Lambda(\boldsymbol{\tau})}$, the following typechecks:

$$\begin{array}{ccc} x^{\boldsymbol{\sigma}} & \xrightarrow{\quad f \quad} & f(x^{\boldsymbol{\sigma}}) \\ {\scriptstyle \partial_{\boldsymbol{\sigma}}}\downarrow & & \downarrow{\scriptstyle \partial_{\boldsymbol{\tau}}} \\ x^{\boldsymbol{\sigma}}_\Gamma \vdash x^{\boldsymbol{\sigma}}_A \multimap x^{\boldsymbol{\sigma}}_B & \xrightarrow[\theta_\Gamma \vdash \theta_A \multimap \theta_B]{} & y^{\boldsymbol{\tau}}_\Gamma \vdash y^{\boldsymbol{\tau}}_A \multimap y^{\boldsymbol{\tau}}_B \end{array}$$

and it commutes by monotonicity of all constructions involved, since maps in this square are in particular maps of event structures (regarding configurations as conflict-free event structures) and maps coinciding on configurations must be equal (Lemma 6.1.12).

Finally, from Lemmas 7.1.22, 8.2.9 and 8.2.12, $\theta_\Gamma \vdash \theta_A \multimap \theta_B$ is positive. □

Next in the construction of the closure comes that $\Lambda(-)$ has an inverse:

Lemma 8.2.26. *For any Γ, A, B $-$-boards with B strict, we have a bijection:*

$$\Lambda(-) : \mathbf{NTCG}(\Gamma \otimes A, B) \simeq \mathbf{NTCG}(\Gamma, A \multimap B)$$

preserving and reflecting $\approx$.

Proof. Given $\boldsymbol{\sigma} : \Gamma \vdash A \multimap B$, $\Lambda^{-1}(\boldsymbol{\sigma}) : \Gamma \otimes A \vdash B$ has same ess as $\boldsymbol{\sigma}$, with

$$\partial_{\Lambda^{-1}(\boldsymbol{\sigma})}(x^{\boldsymbol{\sigma}}) = x^{\boldsymbol{\sigma}}_\Gamma \parallel x^{\boldsymbol{\sigma}}_A \vdash x^{\boldsymbol{\sigma}}_B$$

for $x^{\boldsymbol{\sigma}} \in \mathscr{C}^{\neq\emptyset}(\boldsymbol{\sigma})$, using (8.3) and likewise for symmetries via (8.4). As in the proof of Proposition 8.2.25, via Lemma 7.1.9 this may be indeed read as a definition.

That $\Lambda^{-1}(-)$ preserves $\approx$ is as in the proof of Proposition 8.2.25. Finally, from the definition we have $\Lambda(\Lambda^{-1}(\boldsymbol{\sigma})) = \boldsymbol{\sigma}$ and $\Lambda^{-1}(\Lambda(\boldsymbol{\tau})) = \boldsymbol{\tau}$, *actual equalities*, as definitions of $\Lambda(-)$ and Λ^{-1} follow opposite directions of the isomorphism (8.3). □

As usual, **evaluation** $\mathbf{ev}_{A,B} : (A \multimap B) \otimes A \vdash B$ may be defined as

$$\mathbf{ev}_{A,B} = \Lambda^{-1}(\mathbb{cc}_{A \multimap B})$$

the uncurrying of the identity, for A, B $-$-boards with B strict. We have:

Lemma 8.2.27. *Consider A, B $-$-boards with B strict. Then,*

$$\begin{array}{rcl} \mathscr{C}^{\neq\emptyset,+}(\mathbf{ev}_{A,B}) & = & \{\mathbb{cc}_{x_A \multimap x_B} \mid x_A \in \mathscr{C}(A),\ x_B \in \mathscr{C}^{\neq\emptyset}(A)\}\,, \\ \mathcal{S}^{\neq\emptyset,+}(\mathbf{ev}_{A,B}) & = & \{\mathbb{cc}_{\theta_A \multimap \theta_B} \mid \theta_A \in \mathcal{S}(A),\ \theta_B \in \mathcal{S}^{\neq\emptyset}(B)\} \end{array}$$

with $\partial_{\mathbf{ev}_{A,B}}(\mathbb{cc}_{x_A \multimap x_B}) = (x_A \multimap x_B) \parallel x_A \vdash x_B$ and idem for symmetries.

Proof. Direct from Lemmas 6.4.4, 7.3.14 and the definition of $\Lambda^{-1}(-)$. □

Finally, uncurrying may be equivalently computed by composing with evaluation:

Lemma 8.2.28. *Consider $\sigma : \Gamma \vdash A \multimap B$ a negative winning strategy.*
Then, $\Lambda^{-1}(\sigma) \approx \mathbf{ev}_{A,B} \odot (\sigma \otimes \mathbb{cc}_A)$.

Proof. We calculate:

$$\begin{array}{rcl} & & \mathscr{C}^{+}(\mathbf{ev}_{A,B} \odot (\sigma \otimes \mathbb{cc}_A)) \\ & \cong & \{(\mathbb{cc}_{x_A \multimap x_B}, (x^\sigma \parallel \mathbb{cc}_{y_A})) \mid \text{matching, c.c.}\} \\ & = & \{(\mathbb{cc}_{x_A \multimap x_B}, (x^\sigma \parallel \mathbb{cc}_{y_A})) \mid \text{matching}\} \\ & = & \{(\mathbb{cc}_{x^\sigma_A \multimap x^\sigma_B}, (x^\sigma \parallel \mathbb{cc}_{x^\sigma_A})) \mid x^\sigma \in \mathscr{C}^{+}(\sigma)\} \\ & \cong & \mathscr{C}^{+}(\sigma) \end{array}$$

using Propositions 7.3.1 and 8.1.1, Lemmas 6.4.4 and 8.2.27; a direct verification to ensure that no causal loop arises; and resolving the constraints due to *matching* by definition of the display maps. All order-isomorphisms involved are compatible with display maps, and the same reasoning compatibly applies to symmetries, so altogether we get a strong isomorphism by Corollary 7.2.13. □

Finally, we may conclude the universal property of relative closure for **NTCG**:

Proposition 8.2.29. *Consider Γ, A, B $-$-arenas with B strict.*
For any $\sigma : \Gamma \otimes A \vdash B$ negative and winning, there is a unique (up to $\approx$) $\Lambda(\sigma) : \Gamma \vdash A \multimap B$ such that $\mathbf{ev}_{A,B} \odot (\Lambda(\sigma) \otimes \mathbb{cc}_A) \approx \sigma$.

Proof. Existence. First, $\Lambda(\sigma)$ is obtained by Proposition 8.2.25. Now we have

$$\mathbf{ev}_{A,B} \odot (\Lambda(\sigma) \otimes \mathbb{cc}_A) \approx \Lambda^{-1}(\Lambda(\sigma)) \approx \sigma$$

by Lemma 8.2.28 followed by Lemma 8.2.26.
Uniqueness. If $\tau : \Gamma \vdash A \multimap B$ satisfies the desired property, then $\Lambda^{-1}(\tau) \approx \sigma$ by Lemma 8.2.28, so $\tau \approx \Lambda(\sigma)$ since $\Lambda(-)$ preserves $\approx$ and by Lemma 8.2.26. □

8.3 Non-Linear Structure

Next, we describe the non-linear structure in boards.

8.3.1 The Resource Modality

Last but not least, we must define !, the resource modality. This is where all the somewhat heavy handling of symmetries up to now finally strikes in.

Functorial action. On a strict $-$-board A, the (non-strict) $-$-board $!A$ was defined in Definition 8.2.13. We must extend it to a functor, thus first define the functorial action.

As observed below Definition 7.1.26, the construction $!(-)$ applies to plain ess, so we may form the ess $!\sigma$ which we must equip with a display map. For that, we use:

Lemma 8.3.1. *For E an ess, there is an order-isomorphism*

$$[-] : \mathsf{Fam}\left(\mathscr{C}^{\neq\emptyset}(E)\right) \simeq \mathscr{C}(!E)$$

with $\mathsf{Fam}(X)$ the set of families of elements of X indexed by finite subsets of $\mathbb{N}$.

Proof. If $(x_i)_{i\in I}$ is a family of non-empty configurations of E, we write

$$[(x_i)_{i\in I}] = \|_{i\in I} x_i = \biguplus_{i\in I} \{i\} \times x_i \in \mathscr{C}(!E)\,,$$

and this decomposition is clearly unique if we insist that all x_is are non-empty. □

This order-iso for configurations of $!E$ is very much like similar isomorphisms we encountered for other constructions, see *e.g.* Lemma 7.1.22 for parallel composition / tensor, Lemma 8.2.9 with &, and Lemma 8.2.12 for $\multimap$. However, unlike those, it does not extend to symmetries. This is because symmetries of $!E$ do not restrict componentwise, on the contrary, their role is precisely to span accross components. However:

Lemma 8.3.2. *Consider A a strict $-$-board. Then, there is an order-isomorphism*

$$[-] : \mathsf{Fam}(\mathcal{S}_+^{\neq\emptyset}(A)) \cong \mathcal{S}_+(!A)$$

compatible with dom *and* cod.

Proof. Direct from the definition. □

In order to define the strategy $!\sigma$, it suffices to assign a display map:

Proposition 8.3.3. *Consider A, B strict $-$-boards, and $\sigma : A \vdash B$ negative winning. Then, the ess $!\sigma$ may be equipped with a display map $\partial_{!\sigma}$ such that*

$$\partial_{!\sigma}([x^i \mid i \in I]) = [x^i_A \mid i \in I] \vdash [x^i_B \mid i \in I]$$

where $\partial_\sigma(x^i) = x^i_A \vdash x^i_B$ for $i \in I$; making $!\sigma : !A \vdash !B$ negative winning.
Moreover, $!(-)$ preserves $\approx$.

Proof. Strategy. The display map defined above on configurations preserves unions and cardinal; moreover it is straightforward to extend it to symmetries. This gives a map of ess by Lemma 7.1.9, we omit the verifications that this yields a strategy.

Negative, winning. A n-ary analogue of the tensor (see Proposition 8.2.16).

Congruence. Consider $f : \boldsymbol{\sigma} \approx \boldsymbol{\tau}$. We construct $!f : !\boldsymbol{\sigma} \approx !\boldsymbol{\tau}$ simply componentwise. Clearly, it is an isomorphism, we must show that it commutes with the display to the board up to a positive symmetry. For that, consider $[x^i \mid i \in I] \in \mathscr{C}(!\boldsymbol{\sigma})$. For $i \in I$

$$\begin{array}{ccc} x^i & \xrightarrow{\quad f \quad} & f(x^i) \\ {\scriptstyle \partial_{\boldsymbol{\sigma}}}\downarrow & & \downarrow{\scriptstyle \partial_{\boldsymbol{\tau}}} \\ x^i_A \vdash x^i_B & \xrightarrow[\theta^i_A \vdash \theta^i_B]{} & y^i_A \vdash y^i_B \end{array}$$

where $\theta^i_A \in \mathcal{S}_-(A)$ and $\theta^i_B \in \mathcal{S}_+(B)$. It follows by Lemma 8.3.2 that the diagram

$$\begin{array}{ccc} [x^i \mid i \in I] & \xrightarrow{\quad !f \quad} & [f(x^i) \mid i \in I] \\ {\scriptstyle \partial_{!\boldsymbol{\sigma}}}\downarrow & & \downarrow{\scriptstyle \partial_{!\boldsymbol{\tau}}} \\ [x^i_A \mid i \in I] \vdash [x^i_B \mid i \in I] & \xrightarrow[[\theta^i_A \mid i \in I] \vdash [\theta^i_B \mid i \in I]]{} & [y^i_A \mid i \in I] \vdash [y^i_B \mid i \in I] \end{array}$$

typechecks, where $[\theta^i_A \mid i \in I] \in \mathcal{S}_-(!A)$ and $[\theta^i_B \mid i \in I] \in \mathcal{S}_+(!A)$. It also commutes as it typechecks for all sub-configurations of $[x^i \mid i \in I]$, all as constructions involved are monotone and preserve cardinal, by Lemma 6.1.12 this implies commutation. □

It remains to prove that $!(-)$ is a $\sim$-functor:

Proposition 8.3.4. *We have a $\sim$-functor* $!(-) : \mathbf{NTCG} \to \mathbf{NTCG}$.

Proof. Preservation of copycat. We build an isomorphism $!(\mathbf{cc}_A) \approx \mathbf{cc}_{!A}$. This is done via Corollary 7.2.13, by constructing an order-isomorphism between $\mathcal{S}^+(!(\mathbf{cc}_A))$ and $\mathcal{S}^+(\mathbf{cc}_{!A})$ compatible with dom and cod and with display maps. First, notice that

$$\mathcal{S}^+(\mathbf{cc}_{!A}) \cong \mathcal{S}(!A) \tag{8.5}$$

by Lemma 7.3.14. From the definition and Lemma 7.3.14, $\mathcal{S}(!\mathbf{cc}_A)$ comprises

$$\theta : [x^i_A \parallel x^i_A \mid i \in I] \cong [y^j_A \parallel y^j_A \mid j \in J]$$

given by some bijection $\pi : I \simeq J$ and $\theta^i \parallel \theta^i : x^i_A \parallel x^i_A \cong_{\mathbf{cc}_A} y^j_A \parallel j^j_A$ – but this is clearly isomorphic to the data of $\pi : I \simeq J$ and a family of symmetries $\theta^i : x^i_A \cong_A y^i_A$, *i.e.* to $\mathcal{S}(!A)$. Together with (8.5), this yields a bijection $\mathcal{S}^+(\mathbf{cc}_{!A}) \simeq \mathcal{S}^+(!\mathbf{cc}_A)$ which is easily checked to be an order-isomorphism preserving display maps as required.

Preservation of composition. We build an iso $\mathcal{S}^+(!(\boldsymbol{\tau} \odot \boldsymbol{\sigma})) \cong \mathcal{S}^+(!(\boldsymbol{\tau}) \odot !(\boldsymbol{\sigma}))$.

By definition and Proposition 7.3.1, symmetries in $\mathcal{S}^+(!(\boldsymbol{\tau} \odot \boldsymbol{\sigma}))$ are given by a bijection $\pi : I \simeq J$, and for each $i \in I$, a symmetry

$$\theta^{\boldsymbol{\tau}}_i \odot \theta^{\boldsymbol{\sigma}}_i : x^{\boldsymbol{\tau}}_i \odot x^{\boldsymbol{\sigma}}_i \cong_{\boldsymbol{\tau} \odot \boldsymbol{\sigma}} y^{\boldsymbol{\tau}}_{\pi(i)} \odot y^{\boldsymbol{\sigma}}_{\pi(i)}$$

in $\mathcal{S}^+(\boldsymbol{\tau} \odot \boldsymbol{\sigma})$. Then $\pi : I \simeq J$ together with the family of $\theta_i^{\boldsymbol{\tau}} : x_i^{\boldsymbol{\tau}} \cong_{\boldsymbol{\tau}} y^{\boldsymbol{\tau}}_{\pi(i)}$ yields

$$\theta^{!\boldsymbol{\tau}} : [x_i^{\boldsymbol{\tau}} \mid i \in I] \cong_{!\boldsymbol{\tau}} [y_j^{\boldsymbol{\tau}} \mid j \in J] .$$

Likewise, set $I' = \{i \in I \mid \theta_i^{\boldsymbol{\sigma}} \neq \emptyset\}$, and $\pi' : I' \simeq J'$ as the restriction of π. Then, $\pi' : I' \simeq J'$ together with the family of $\theta_i^{\boldsymbol{\sigma}} : x_i^{\boldsymbol{\sigma}} \cong_{\boldsymbol{\sigma}} y^{\boldsymbol{\sigma}}_{\pi'(i)}$ for $i \in I'$ yields

$$\theta^{!\boldsymbol{\sigma}} : [x_i^{\boldsymbol{\sigma}} \mid i \in I'] \cong_{!\boldsymbol{\sigma}} [y_j^{\boldsymbol{\sigma}} \mid j \in J'] .$$

It is direct that $\theta^{!\boldsymbol{\sigma}}$ and $\theta^{!\boldsymbol{\tau}}$ are causally compatible, forming $\theta^{!\boldsymbol{\tau}} \odot \theta^{!\boldsymbol{\sigma}} \in \mathcal{S}^+(!\boldsymbol{\tau} \odot !\boldsymbol{\sigma})$, and that the construction preserves unions and display maps.

Reciprocally, symmetries in $\mathcal{S}^+(!\boldsymbol{\tau} \odot !\boldsymbol{\sigma})$ are given by causally compatible

$$\theta^{!\boldsymbol{\sigma}} : x^{!\boldsymbol{\sigma}} \cong_{!\boldsymbol{\sigma}} y^{!\boldsymbol{\sigma}} , \qquad \theta^{!\boldsymbol{\tau}} : x^{!\boldsymbol{\tau}} \cong_{!\boldsymbol{\tau}} y^{!\boldsymbol{\tau}}$$

respectively given by $\pi^{\boldsymbol{\sigma}} : I \simeq J$ and $\theta_i^{\boldsymbol{\sigma}} : x_i^{\boldsymbol{\sigma}} \cong_{\boldsymbol{\sigma}} y^{\boldsymbol{\sigma}}_{\pi(i)}$ for $i \in I$; and by $\pi^{\boldsymbol{\tau}} : I' \simeq J'$ and $\theta_i^{\boldsymbol{\tau}} : x_i^{\boldsymbol{\tau}} \cong_{\boldsymbol{\tau}} y^{\boldsymbol{\tau}}_{\pi'(i)}$ for $i \in I'$. Now, for $i \in I$, $\theta_i^{\boldsymbol{\sigma}}$ is non-empty. Since $\boldsymbol{\sigma}$ is negative, $(\theta_i^{\boldsymbol{\sigma}})_B$ is non-empty. So as $\theta^{!\boldsymbol{\sigma}}$ and $\theta^{!\boldsymbol{\tau}}$ are matching, we must have $i \in I'$ as well with $\theta_i^{\boldsymbol{\tau}}$ non-empty. This also entails that $\pi(i) = \pi'(i)$, so that $I \subseteq I'$, $J \subseteq J'$ and $\pi \subseteq \pi'$. For $i \in I' \setminus I$, set $x_i^{\boldsymbol{\sigma}} = \emptyset$, $y_i^{\boldsymbol{\sigma}} = \emptyset$, and $\theta_i^{\boldsymbol{\sigma}} = \emptyset$. Now it is direct that for any $i \in I$, the symmetries $\theta_i^{\boldsymbol{\sigma}}$ and $\theta_i^{\boldsymbol{\tau}}$ are matching causally compatible, so that $\theta_i^{\boldsymbol{\tau}} \odot \theta_i^{\boldsymbol{\sigma}} \in \mathcal{S}^+(\boldsymbol{\tau} \odot \boldsymbol{\sigma})$. So altogether we have a bijection $\pi : I \simeq J$, and for all $i \in I$ a symmetry

$$\theta_i^{\boldsymbol{\tau}} \odot \theta_i^{\boldsymbol{\sigma}} : x_i^{\boldsymbol{\tau}} \odot x_i^{\boldsymbol{\sigma}} \cong_{\boldsymbol{\tau} \odot \boldsymbol{\sigma}} y_i^{\boldsymbol{\tau}} \odot y_i^{\boldsymbol{\sigma}} ,$$

that is, the data for a symmetry in $!(\boldsymbol{\tau} \odot \boldsymbol{\sigma})$. It is a direct verification that this transformation preserves unions and display maps.

These operations are defined on symmetries but the same applies to configurations compatibly, informing an isomorphism $!(\boldsymbol{\tau} \odot \boldsymbol{\sigma}) \approx !\boldsymbol{\tau} \odot !\boldsymbol{\sigma}$ by Corollary 7.2.13. □

Notice that we formulated the latter isomorphism directly on symmetries, whereas in many earlier proofs, we worked on configurations and only remarked that the same applies to symmetries. In most cases (*e.g.* tensor, with, composition), the construction applies to configurations and symmetries in exactly the same way. But the situation is really different for $!(-)$: this is the only construction that introduces new, non-trivial symmetries, and so treats the two levels genuinely differently.

A $\sim$-comonad. It remains to equip $!(-)$ with adequate structural morphisms, which as before will be obtained by lifting appropriate renamings:

Lemma 8.3.5. *Consider A a $-$-board. Then, we have winning renamings:*

$$\begin{array}{rcccl} \mathsf{der}_A & : & A^\perp & \to & (!A)^\perp \\ & & a & \mapsto & (0, a) \end{array} \qquad\qquad \begin{array}{rcccl} \mathsf{dig}_A & : & (!!A)^\perp & \to & (!A)^\perp \\ & & (i, (j, a)) & \mapsto & (\langle i, j \rangle, a) \end{array}$$

with $\langle -, - \rangle : \mathbb{N} \times \mathbb{N} \simeq \mathbb{N}$ a fixed bijection.

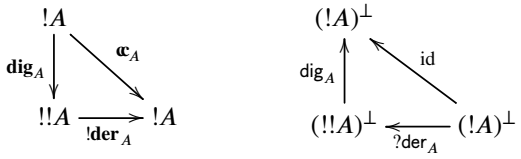
Proof. Direct verification. □

Using these, for any $-$-board A we may obtain strategies for the comonad structure

$$\mathbf{der}_A : !A \vdash A\,, \qquad \mathbf{dig}_A : !A \vdash !!A$$

respectively called **dereliction** and **digging** after the following components in linear logic, simply by backward lifting: $\mathbf{der}_A = \mathfrak{w}_{\mathsf{der}_A}$ and $\mathbf{dig}_A = \mathfrak{w}_{\mathsf{dig}_A}$.

We shall prove that this makes $(!, \mathbf{der}, \mathbf{dig})$ a $\sim$-comonad on **NTCG**, but this requires us to prove a number of coherence laws. For the symmetric monoidal structure, recall that such coherence laws simply followed by equations on renamings, and concluding from functoriality of lifting (*i.e.* Proposition 8.1.9). However here, the coherence laws for renamings do not hold on the nose. Consider for example one of the unit laws:



with the corresponding diagram on renamings on the right hand side, where $?\mathsf{der}_A(i,a) = (i, \mathsf{der}_A(a)) = (i,(0,a))$. This diagram for renamings does not commute: the bottom-left path takes (i,a) to $(\langle i,0\rangle, a)$ instead of (i,a) as it should. But of course, this is no surprise: it is well-known that laws for exponentials require reindexing, and it is exactly the reason why we constructed an equivalence $\approx$ on strategies authorizing reindexings.

Note then that the diagram above does commute up to *positive symmetry*, *i.e.*

$$\mathsf{dig}_A \circ ?\mathsf{der}_A \sim^+ \mathrm{id}_{(!A)^\perp}$$

where recall that $\sim^+$, defined in Definition 7.2.3, formalizes that the two maps are equal up to a *positive symmetry*, *i.e.* reindexing of Player moves.

This property will be sufficient to establish positive isomorphisms, exploiting:

Lemma 8.3.6. *Consider $f, f' : A \to B$ winning renamings such that $f \sim^+ f'$ (respectively, $g, g' : B^\perp \to A^\perp$ such that $g \sim^+ g'$). Then, $\mathfrak{c}_f \approx \mathfrak{c}_{f'}$ (resp. $\mathfrak{w}_g \approx \mathfrak{w}_{g'}$).*

Proof. The positive isos are simply identity maps between the underlying ess, and commutation with the display maps up to positive symmetry is obvious. □

Now, we can prove the expected comonad laws.

Proposition 8.3.7. *We have $(!, \mathbf{der}, \mathbf{dig})$ a $\sim$-comonad on* **NTCG**.

Proof. Naturality. We show naturality of $\mathbf{der}_A$. Consider $\sigma : A \vdash B$. By Lemma 8.1.8, $\mathcal{S}^+(\sigma \odot \mathbf{der}_A) \cong \mathcal{S}^+(\sigma)$ with, for $\theta \in \mathcal{S}^+(\sigma)$, $\partial(\theta) = (0,\theta_A) \vdash \theta_B$ – where $(0,\theta_A)$ stands for $\{((0,a),(0,a')) \mid (a,a') \in \theta_A\}$. Likewise by Proposition 7.3.1 and Lemma 7.3.14, $\mathcal{S}^+(\mathbf{der}_B \odot !\sigma)$ comprises matching pairs of $\theta^{!\sigma} \in \mathcal{S}^+(!\sigma)$ and $\theta_B \parallel \theta_B \in \mathcal{S}^+(\mathfrak{c}_B)$, where the latter displays to $(0,\theta_B) \vdash \theta_B$. So by *matching*, $\theta^{!\sigma}$ spans only one copy of index 0, and thus must display to some $(0,\theta_A) \vdash (0,\theta_B)$. From this,

the required order-isomorphism follows easily.

Next, we show naturality of $\mathbf{dig}_A$. First, symmetries in $\mathcal{S}^+(!!\boldsymbol{\sigma} \odot \mathbf{dig}_A)$ are somewhat hard to describe: by Lemma 8.1.8, they correspond to those

$$\theta : [[x^{i,j} \mid j \in J_i] \mid i \in I] \quad \cong \quad [[y^{k,l} \mid l \in L_k] \mid k \in K] \tag{8.6}$$

given by *(1)* a bijection $\pi : I \simeq K$; *(2)* for each $i \in I$, a bijection $\pi_i : J_i \simeq L_{\pi(i)}$; and *(3)* for each $i \in I, j \in J_i$, a symmetry $\theta^{i,j} : x^{i,j} \cong_{\boldsymbol{\sigma}} y^{\pi(i),\pi_i(j)}$. This is displayed to

$$\begin{array}{rcccc} \theta_{!A} & : & [x_A^{i,j} \mid \langle i,j\rangle \in X] & \cong & [y_A^{k,l} \mid \langle k,j\rangle \in Y] \\ \theta_B & : & [[x_B^{i,j} \mid j \in J_i] \mid i \in I] & \cong & [[y_B^{k,l} \mid l \in L_k] \mid k \in K] \end{array}$$

where X comprises those $\langle i,j\rangle$ such that $i \in I$ and $j \in J_i$ and $x_A^{i,j}$ is non-empty, and likewise for Y. For the other side of the naturality diagram, by Proposition 7.3.1 and Lemma 7.3.14, symmetries in $\mathcal{S}^+(\mathbf{dig}_B \odot !\boldsymbol{\sigma})$ correspond to pairs

$$\begin{array}{rcccc} \theta^{!\sigma} & : & [x^m \mid m \in X] & \cong & [y^n \mid n \in Y] \\ \theta_{!!B} & : & [[x^{i,j} \mid i \in I] \mid j \in J_i] & \cong & [[y^{k,l} \mid k \in K] \mid l \in L_k] \end{array}$$

which are matching, implying that X comprises those $\langle i,j\rangle$ such that $i \in I$ and $j \in J_i$; Y comprises those $\langle k,l\rangle$ such that $k \in K$ and $l \in L_k$, and $\theta^{!\sigma}$ first follows a bijection $\pi : I \simeq K$ and for each $i \in I$ a bijection $J_i \simeq K_{\pi(i)}$. This means that equivalently,

$$\theta^{!\sigma} : [[x^{\langle i,j\rangle} \mid j \in J_i] \mid i \in I] \cong [[y^{\langle k,l\rangle} \mid l \in L_k] \mid k \in K]$$

a symmetry in $!!\boldsymbol{\sigma}$. This matches (8.6); this correspondence is an order-iso compatible with display maps so that $!!\boldsymbol{\sigma} \odot \mathbf{dig}_A \approx \mathbf{dig}_B \odot !\boldsymbol{\sigma}$ as required by Corollary 7.2.13.

Coherence. First, we define renamings:

$$?\mathsf{der}_A : (!A)^\perp \to (!!A)^\perp, \qquad ?\mathsf{dig}_A : (!!A)^\perp \to (!!!A)^\perp$$

simply copying the index for the outer !. Now, Proposition 8.3.4 (in particular, preservation of copycat) provides us with isomorphisms of ess

$$!\mathfrak{C}_A \cong \mathfrak{C}_{!A}, \qquad !(\mathfrak{C}_{!A}) \cong \mathfrak{C}_{!!A}$$

which simply by verifying compatibility with display maps, yield positive isomorphisms

$$!(\mathfrak{D}_{\mathsf{der}_A}) \approx \mathfrak{D}_{?\mathsf{der}_A}, \qquad !(\mathfrak{D}_{\mathsf{dig}_A}) \approx \mathfrak{D}_{?\mathsf{dig}_A}, \tag{8.7}$$

so that finally, noting the positive symmetries between renamings

$$\begin{array}{rcl} \mathsf{dig}_A \circ ?\mathsf{der}_A & \sim^+ & \mathrm{id} \\ \mathsf{dig}_A \circ \mathsf{der}_{!A} & \sim^+ & \mathrm{id} \\ \mathsf{dig}_A \circ \mathsf{dig}_{!A} & \sim^+ & \mathsf{dig}_A \circ ?\mathsf{dig}_{!A} \end{array}$$

we get the coherence laws up to $\approx$ by (8.7), Lemma 8.3.6 and Proposition 8.1.9. □

This structure lets us define the particularly important following operation. If A and B are strict $-$-boards and $\sigma : !A \vdash B$, then its **promotion** is

$$\sigma^{!} \quad : \quad !A \vdash !B \tag{8.8}$$

defined as $\sigma^{!} = !(\sigma) \odot \mathbf{dig}_A$, involving both digging and the functorial action of !. Promotion may be taken as a primitive, letting us state:

Corollary 8.3.8. *Consider the $\sim$-functor $! : \mathbf{NTCG}_s \to \mathbf{NTCG}$, together with:* (1) *the* dereliction *strategy $\mathsf{der}_A : !A \vdash A$ for strict A; and* (2) *the promotion operation $\sigma^{!} : !A \vdash !B$ for all $\sigma : !A \vdash B$ with A and B strict.*

This yields a relative $\sim$-comonad*, with respect to the inclusion $\mathbf{NTCG}_s \subseteq \mathbf{NTCG}$.*

Proof. Straightforward. □

But since we have a regular comonad, why emphasize this less standard relative comonad structure? Well, because as we shall see in Section 10.4, the comonad structure is not preserved by the collapse to the relational model: we shall see that the interplay between the collapse and $!A$ only works well when A is strict.

Seely isomorphisms. First, we have an equality $!\top = \mathbf{1}$, so the corresponding isomorphism is realized by the identity. We need one final isomorphism

$$\mathbf{mon}_{A,B} : !A \otimes !B \cong !(A \& B)\,,$$

which we shall obtain via the following renamings:

Lemma 8.3.9. *Consider A, B two strict $-$-boards. Then, we have winning renamings:*

$$\begin{array}{rcccc} \mathsf{mon}_{A,B} & : & (!(A\&B))^{\perp} & \to & (!A \otimes !B)^{\perp} \\ & & (i,(j,m)) & \mapsto & (j,(i,m)) \end{array}$$

$$\begin{array}{rcccc} \mathsf{mon}^{*}_{A,B} & : & (!A \otimes !B)^{\perp} & \to & (!(A\&B))^{\perp} \\ & & (i,(j,m)) & \mapsto & (\langle i,j\rangle,(i,m)) \end{array}$$

satisfying $\mathsf{mon}_{A,B} \circ \mathsf{mon}^{*}_{A,B} \sim^{+} \mathrm{id}$ *and* $\mathsf{mon}^{*}_{A,B} \circ \mathsf{mon}_{A,B} \sim^{+} \mathrm{id}$.

Proof. Straightforward. □

As expected, the required isomorphism $\mathbf{mon}_{A,B}$ is obtained by backward lifting, *i.e.*

$$\mathbf{mon}_{A,B} = \mathfrak{B}_{\mathsf{mon}_{A,B}}\,, \qquad \mathbf{mon}^{-1}_{A,B} = \mathfrak{B}_{\mathsf{mon}^{*}_{A,B}}\,,$$

inverses up to positive iso as follows from Lemmas 8.3.9, 8.3.6 and Proposition 8.1.9.

Note that these strategies cannot be defined via forward lifting, as neither $\mathsf{mon}_{A,B}$ nor $\mathsf{mon}^{*}_{A,B}$ would be valid renamings if typed without the $(-)^{\perp}$, both failing *receptive*.

It remains to show that the Seely isomorphism is compatible with promotion. For that, it is convenient to introduce one last map. Consider the winning renaming

$$\begin{array}{rcccc} \mathsf{con}_A & : & ((!A) \otimes (!A))^{\perp} & \to & (!A)^{\perp} \\ & & (i,(j,a)) & \mapsto & (\langle i,j\rangle, a)\,, \end{array}$$

called *contraction* (for A a $-$-board), which induces $\mathbf{con}_A = \mathfrak{B}_{\mathsf{con}_A} : !A \vdash !A \otimes !A$. Using this additional strategy, we first prove the following lemma:

Lemma 8.3.10. *Consider Γ, A, B strict $-$-boards, and $\boldsymbol{\sigma} : !\Gamma \vdash A, \boldsymbol{\tau} : !\Gamma \vdash B$. Then, the following diagram commutes up to positive isomorphism:*

$$\begin{array}{ccc} !!\Gamma & \xrightarrow{!\langle \boldsymbol{\sigma},\boldsymbol{\tau}\rangle} & !(A\&B) \\ {\scriptstyle \mathbf{con}_{!\Gamma}}\downarrow & & \downarrow{\scriptstyle \mathbf{mon}^{-1}_{A,B}} \\ !!\Gamma \otimes !!\Gamma & \xrightarrow[!\boldsymbol{\sigma}\otimes !\boldsymbol{\tau}]{} & !A \otimes !B \end{array}$$

Proof. By Lemma 8.1.8, the bottom-left path is $(!\boldsymbol{\sigma} \otimes !\boldsymbol{\tau}) \cdot \mathsf{con}_{!\Gamma}$, with ess $!\boldsymbol{\sigma} \parallel !\boldsymbol{\tau}$.

We now examine the $+$-covered symmetries in the upper-right strategy. Via Proposition 7.3.1 and Lemma 7.3.14, those correspond to symmetries $\theta \in \mathcal{S}^{+}(!(\boldsymbol{\sigma} + \boldsymbol{\tau}))$ whose display on $!(A\&B)$ is in the image of $\mathsf{mon}^{*}_{A,B}$, *i.e.* of the form

$$\begin{array}{cl} & \{(\langle 1,i\rangle,(1,a)),(\langle 1,j\rangle,(1,a'))) \mid ((i,a),(j,a')) \in \theta_A\} \\ \uplus & \{(\langle 2,i\rangle,(2,b)),(\langle 2,j\rangle,(2,b'))) \mid ((i,b),(j,b')) \in \theta_B\} \end{array}$$

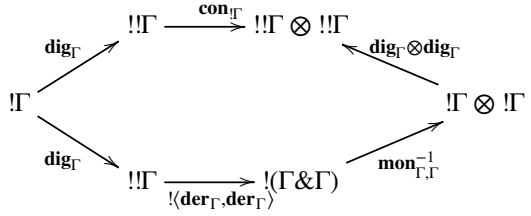
for $\theta_A \in \mathcal{S}(!A)$ and $\theta_B \in \mathcal{S}(!B)$; *i.e.* symmetries using copy indices of the form $\langle 1,-\rangle$ for A and $\langle 2,-\rangle$ for B. This means that θ also decomposes uniquely in this way, relating events of $\boldsymbol{\sigma}$ on indices $\langle 1,-\rangle$, and events of $\boldsymbol{\tau}$ on indices $\langle 2,-\rangle$. This directly informs the required order-isomorphism with $!\boldsymbol{\sigma} \parallel !\boldsymbol{\tau}$; we omit further details. □

With this lemma, we may finally prove compatibility of **mon** with promotion:

Lemma 8.3.11. *Consider Γ, A, B strict $-$-boards, and $\boldsymbol{\sigma} : !\Gamma \vdash A, \boldsymbol{\tau} : !\Gamma \vdash B$. Then, the following diagram commutes up to positive symmetry:*

$$\begin{array}{ccc} !\Gamma & \xrightarrow{\langle \boldsymbol{\sigma},\boldsymbol{\tau}\rangle^{!}} & !(S\&T) \\ {\scriptstyle \langle \mathsf{der},\mathsf{der}\rangle^{!}}\downarrow & & \downarrow{\scriptstyle m^{-1}} \\ !(\Gamma\&\Gamma) & & !S \otimes !T \\ & {\scriptstyle m^{-1}}\searrow \quad \nearrow{\scriptstyle \boldsymbol{\sigma}^{!}\otimes\boldsymbol{\tau}^{!}} & \\ & !\Gamma \otimes !\Gamma & \end{array}$$

Proof. Via Lemma 8.3.10, it remains to show that the following diagram commutes



up to positive symmetry. We first observe that considering the winning renaming

$$\begin{array}{rcccc} \mu_\Gamma & : & (!(\Gamma\&\Gamma))^\perp & \to & (!!\Gamma)^\perp \\ & & (i,(j,m)) & \mapsto & (i,(0,m))\,, \end{array}$$

we have $!\langle \mathbf{der}_\Gamma, \mathbf{der}_\Gamma\rangle \approx \mathfrak{w}_{\mu_\Gamma}$, as is establish by a – by now routine – analysis of symmetries of $!\langle \mathbf{der}_\Gamma, \mathbf{der}_\Gamma\rangle$. The diagram then follows from Proposition 8.1.9, Lemma 8.3.6 and a direct computation of the composition of the renamings involved. □

We may finally conclude the main result of this chapter:

Theorem 8.3.12. *There is* **NTCG***, a relative Seely ∼-category.*

Proof. The only components which were not detailed are naturality of $\mathbf{mon}_{A,B}$, and the coherence diagrams making $!(-)$ a strong symmetric monoidal ∼-functor. All these follow from Proposition 8.1.9 via a now routine methodology; we omit details. □

And of course (see Appendix A.1 for the construction of the Kleisli category):

Corollary 8.3.13. *The Kleisli ∼-category* $\mathbf{NTCG}_!$ *is cartesian closed.*

In the sequel, we write $\odot_!$ for Kleisli composition, *i.e.* composition in $\mathbf{NTCG}_!$.

8.3.2 Recursion

The last piece of structure that we shall construct in **NTCG** is the interpretation of the *fixpoint combinator*, used to interpret recursion. Following the standard denotational semantics practice, this infinite strategy will be obtained as a least fixed point with respect to a partial order on concrete strategies.

First, we introduce this partial order:

Definition 8.3.14. *Consider A a board, and* $\sigma, \tau : A$ *winning strategies.*
We write $\sigma \lhd \tau$ *if* $\mathscr{C}(\sigma) \subseteq \mathscr{C}(\tau)$ *– so in particular* $|\sigma| \subseteq |\tau|$*, and additionally:*

(1) *for all* $s_1, s_2 \in \sigma$, $s_1 \leq_\sigma s_2$ *iff* $s_1 \leq_\tau s_2$,
(2) *for all* $s_1, s_2 \in \sigma$, $s_1 \#_\sigma s_2$ *iff* $s_1 \#_\tau s_2$,
(3) *for all* $x, y \in \mathscr{C}(\sigma)$ *and bijection* $\theta : x \simeq y$, *we have* $\theta \in \mathscr{S}(\sigma)$ *iff* $\theta \in \mathscr{S}(\tau)$,
(4) *for all* $s \in \sigma$, $\partial_\sigma(s) = \partial_\tau(s)$,

i.e. *all components compatible with the inclusion.*

Strategies on A, ordered by $\lhd$, form a *directed complete partial order*, with respect to which all operations on strategies are easily shown to be continuous. We have:

Proposition 8.3.15. *Consider A a board, and D a directed set of strategies on A. Then D has a least fixed point for $\lhd$, written $\vee D$, and satisfying:*

$$\mathscr{C}^+(\vee D) = \bigcup_{\sigma \in D} \mathscr{C}^+(\sigma)\,, \qquad \mathscr{S}^+(\vee D) = \bigcup_{\sigma \in D} \mathscr{S}^+(\sigma)\,.$$

Moreover, if every $\sigma \in D$ is winning, then so is $\vee D : A$.

Proof. We set $|\vee D| = \bigcup_{\sigma \in D} |\sigma|$, and take the union as well for all other components. All verifications are straightforward, and omitted. □

For any $-$-boards A and B this turns $\mathbf{NTCG}(A \vdash B)$ into a dcpo. We have:

Lemma 8.3.16. *The following operations on strategies are continuous:*

$$\begin{array}{rcrcl} - \odot - & : & \mathbf{NTCG}(B \vdash C) \times \mathbf{NTCG}(A \vdash B) & \to & \mathbf{NTCG}(A \vdash C) \\ - \otimes - & : & \mathbf{NTCG}(A \vdash B) \times \mathbf{NTCG}(C \vdash D) & \to & \mathbf{NTCG}(A \otimes C \vdash B \otimes D) \\ \langle -, - \rangle & : & \mathbf{NTCG}(\Gamma \vdash A) \times \mathbf{NTCG}(\Gamma \vdash B) & \to & \mathbf{NTCG}(\Gamma \vdash A \& B) \\ (-)^! & : & \mathbf{NTCG}(!\Gamma \vdash A) & \to & \mathbf{NTCG}(!\Gamma \vdash !A) \end{array}$$

Proof. Straightforward. □

The plan is then to follow the standard denotational semantics route, and define the recursion combinator via Kleene's fixed point theorem, as the least fixed point of a well-chosen functional continous via the proposition above.

For that, we must deal with a minor inconvenience: Kleene's fixed point theorem applies to continuous functionals over a dcpo *with a least element*, but the dcpo $\mathbf{NTCG}(A)$ of winning strategies over a board A *does not* in general have a least element. Indeed, strategies on A cannot in general be empty, as they must have – by receptivity – events matching the negative minimal events of A. But "matching" is loose: they are free to *name* those events arbitrarily. We solve this as in [Castellan et al., 2019]: we choose one minimal $\bot_A : A$, serving as the canonical least element. Then for any $\sigma : A$, we pick an isomorphic $\sigma \approx \sigma^\flat : A$ s.t. $\bot_A \lhd \sigma^\flat$, obtained by renaming minimal events. We write $\mathcal{D}_A$ for the pointed dcpo of strategies above $\bot_A$. For this new operation:

Lemma 8.3.17. *For any board A, the following operation is continuous:*

$$(-)^\flat : \mathbf{NTCG}(A) \to \mathcal{D}_A\,.$$

Proof. Straightforward. □

As all operations involved are continuous, for any strict $-$-board A we have

$$\begin{array}{rcrcl} F & : & \mathcal{D}_{!\mathsf{T} \vdash (A \to A) \to A} & \to & \mathcal{D}_{!\mathsf{T} \vdash (A \to A) \to A} \\ & & \sigma & \mapsto & (\lambda f^{A \to A}.\, f\,(\sigma\, f))^\flat\,, \end{array}$$

continuous, written in λ-calculus syntax relying on the constructions on strategies corresponding to the cartesian closed structure of $\mathbf{NTCG}_!$ [Lambek and Scott, 1988]. Here, we use $A \to B$ as a notation for $!A \multimap B$. Hence by Kleene's fixpoint theorem,

$$\mathcal{Y}_A = \bigvee_{n \in \mathbb{N}} F^n(\bot) \in \mathbf{NTCG}_!(\top, (A \to A) \to A)$$

is a least fixpoint of F. Finally, in the presence of a strict $-$-board Γ serving as context, we set $\mathcal{Y}_{\Gamma,A} = \mathcal{Y}_A \odot_! e_\Gamma$ where $e_\Gamma \in \mathbf{NTCG}_!(\Gamma, \top)$ is the terminal morphism, yielding

$$\mathcal{Y}_{\Gamma,A} \in \mathbf{NTCG}_!(\Gamma, !(!A \multimap A) \multimap A) \tag{8.9}$$

which will provide the interpretation of recursion throughout this monograph.

8.4 History and Related Work

Compact Closed Category. Following Joyal's compact closed category of Conway games [Joyal, 1977], it is certainly expected that a category of unpolarized games should be compact closed – a notable exception to that is of course Blass games [Blass, 1992] which famously do not form a category. A detailed analysis of this so-called "Blass problem" in the context of Conway games may be found in [Melliès, 2004b].

For our particular technical setting of concurrent games on event structures, the compact closed structure was first described in [Castellan et al., 2017a] without symmetry, and extended to thin concurrent games in [Castellan et al., 2019].

Seely Category. Seely categories is a convenient categorical axiomatization of models of intuitionistic linear logic, in the presence of a product. Many categories of negative games form Seely categories, starting with *simple games* [Hyland, 1997] with respect to several choices of exponential modalities.

In the original presentation of thin concurrent games (first in conference version in [Castellan et al., 2015] then detailed in [Castellan et al., 2019]), we did not construct a model of intuitionistic linear logic. Indeed, we opted to construct a category called *Concurrent Hyland-Ong games*, with objects exactly the same *arenas* as in traditional Hyland-Ong games. We wanted a conservative extension of HO games: the appeal was to make it easier to adapt and extend the developments that led to their versatility and impact – and in particular, concepts such as *justifiers* and conditions on strategies such as *innocence*. Technically, this was done by building a tcg from any Hyland-Ong arena via a variant of the resource modality (inspired by [Harmer et al., 2007]) performing duplications not just at the root, but deep within the arena.

In developing probabilistic concurrent games in [Castellan et al., 2018b], we reverted to an AJM-style !, which made it easier to link with the weighted relational model. But then we (re-)discovered (Melliès, at least, was certainly aware of this from his work in asynchronous games) that justifiers, and in general the necessary ingredients to the Hyland-Ong machinery, were already available in concurrent games even with an AJM-style exponential! This is one of the advantages of the causal setting, as opposed to

traditional AJM games. In retrospect, this makes *Concurrent Hyland-Ong Games* misguided: they suffer from the same defect as traditional Hyland-Ong games (lacking a linear decomposition), while working in the Kleisli category of a Seely category of thin concurrent games gives you the best of both AJM and HO worlds.

Concerning *relative* Seely category: from **TCG** one can get an actual Seely category [Castellan and Clairambault, 2021]. But the general linear arrow construction is awkward, and the interest of the added expressiveness is unclear for semantic purposes. Finally, the full Seely category structure lacks a clean connection with relational semantics. *Relative* Seely categories give a good compromise for semantics of call-by-name.

Winning Strategies. Finally, the *payoff* mechanism and winning strategies we use here are an adaptation of similar mechanisms in [Melliès, 2005] (with a later refinement in [Melliès and Tabareau, 2007]) where they serve essentially the same purpose as here: they cope for the otherwise inherently *affine* nature of game semantics, making strategies more linear and identifying the *stopping* configurations of games, those corresponding points of the web in the relational model. Note that we first adapted this mechanism to concurrent games on event structures in [Clairambault and de Visme, 2020].

It might be unclear to the reader why a mechanism making strategies *linear* is appropriate for a model intended for the languages considered in this monograph, which are non-linear and include full recursion and divergence. Firstly, the term *linear* is not really adequate; we should rather say "following the linearity constraints", which are made more liberal by using the exponential. Secondly, our notion of winningness does not force termination, it only constrains resource usage of successful terminating executions, if they exist. More precisely, our (new) restriction of winningness to $+$*-covered* configurations has the subtle effect to prevent from forcing Player to respond in diverging branches, say *i.e.* a diverging boolean.

Part III

Disentangling Parallelism and State

Introduction to Part III

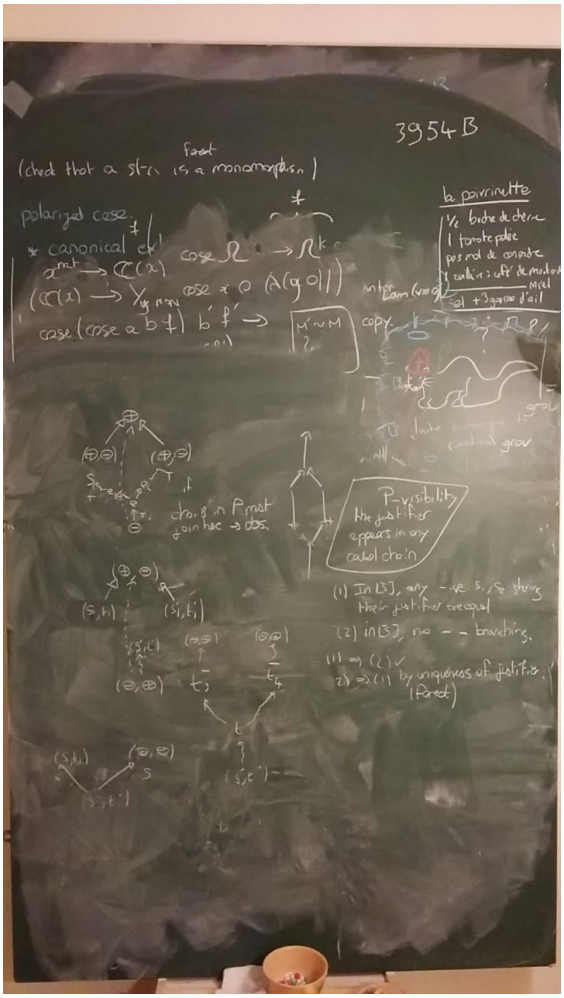


Figure 8.5: The Inception of Parallel Innocence

Now comes the core of this monograph: from **NTCG** we build a cartesian closed category $\multimap\!\triangleright$-**Strat** in which we *disentangle* parallelism and state, as in Section 1.7.

More precisely, we first show that $\multimap\!\triangleright$-**Strat** is an intensionally fully abstract denotational model for $\mathsf{IA}_{/\!/}$, by providing an interpretation-preserving functor

$$\circlearrowleft\text{-}\mathsf{Unf} \ : \ \multimap\!\triangleright\text{-}\mathbf{Strat} \to \circlearrowleft\text{-}\mathbf{Strat}$$

to non-alternating play-based strategies as described in Chapter 5. The category $\multimap\!\triangleright$-**Strat** is close to $\mathbf{NTCG}_!$ but is not exactly that, for two reasons: firstly, because **NTCG** and $\circlearrowleft$-**Strat** do not have the same objects, and there is no easy way to reconstruct an arena in the sense of Hyland-Ong game from a board – so the objects of $\multimap\!\triangleright$-**Strat** will remember both. Secondly, nothing in **NTCG** guarantees the logical well-bracketing used in $\circlearrowleft$-**Strat**, so that must be reimposed as well. This is detailed in Chapter 9.

Once $\multimap\!\triangleright$-**Strat** is constructed along with its interpretation of $\mathsf{IA}_{/\!/}$ and its unfolding to $\circlearrowleft$-**Strat**, we set to build our two conditions on strategies, *sequentiality* and *globularity*, respectively banning *parallelism* and *state*. We first focus on *globularity*. The heart of globularity is *parallel innocence*, a condition that removes state by banning certain causal patterns that are deemed inherently stateful. Parallel innocence can be regarded as a conservative extension of traditional Hyland-Ong innocence, in the sense that paired with sequentiality, we obtain a category (equivalent to) $\downarrow\!\uparrow$-**Inn**. As a conceptually important aside, we show that parallel innocence allows a simple functor to the relational model, compatibly with the structure of a model of intuitionistic linear logic. Finally we introduce *globularity*, which refines parallel innocence by adding a constraint with respect to Questions and Answers. This is detailed in Chapter 10.

Next, we focus on *sequentiality* – we show that in the presence of sequentiality, $\circlearrowleft\text{-}\mathsf{Unf}$ can be refined to an *alternating* unfolding, an interpretation-preserving

$$\downarrow\!\uparrow\text{-}\mathsf{Unf} \ : \ \multimap\!\triangleright\text{-}\mathbf{Seq} \to \downarrow\!\uparrow\text{-}\mathbf{Strat}$$

which entails that $\multimap\!\triangleright$-**Seq** is intensionally fully abstract for IA. We also show that in combination with globularity, $\downarrow\!\uparrow\text{-}\mathsf{Unf}$ yields an interpretation-preserving functor

$$\downarrow\!\uparrow\text{-}\mathsf{Unf} \ : \ \multimap\!\triangleright\text{-}\mathbf{SeqGlob} \to \downarrow\!\uparrow\text{-}\mathbf{InnWB}$$

proving $\multimap\!\triangleright$-**SeqGlob** intensionally fully abstract for PCF. This appears in Chapter 11.

Finally, it remains to establish that $\multimap\!\triangleright$-**Glob** is fully abstract for $\mathsf{PCF}_{/\!/}$. This is proved by an elaborate finite definability argument. To bring concurrent strategies closer to syntax, we must first show that globular strategies enjoy a so-called *meager form*, akin to the forest of *P-views* for traditional sequential innocence. This meager form provides the appropriate notion of *finiteness* or *size* of a globular strategy, on which the definability argument operates. We then show a *factorization* result, to the effect that any globular strategy factors into what is essentially a pure λ-term, and a globular strategy on a *first-order* arena. It remains to define those first-order strategies, which we do – up to the equivalence generated by the relational collapse. This is detailed in Chapter 12.

Altogether, this disentangles parallelism and state.

Chapter 9

The Causal Semantics of $\mathtt{IA}_{/\!/}$ and its Unfolding

The purpose of this chapter is to construct $\multimap\!\!\triangleright$-**Strat** – the causal analogue of $\circlearrowleft$-**Strat**, the fully abstract model of $\mathtt{IA}_{/\!/}$ – and describe the interpretation of $\mathtt{IA}_{/\!/}$.

The outline is as follows. In Section 9.1, we will first describe the interpretation of $\mathtt{IA}_{/\!/}$ in $\mathbf{NTCG}_!$ as constructed in Corollary 8.3.13. In Section 9.2, we shall refine $\mathbf{NTCG}_!$ into $\multimap\!\!\triangleright$-**Strat**, adding a missing condition so that it unfolds to $\circlearrowleft$-**Strat**. In Section 9.3, we detail the unfolding to $\circlearrowleft$-**Strat** and prove full abstraction for $\mathtt{IA}_{/\!/}$.

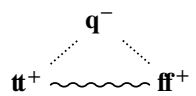
9.1 Interpretation of $\mathtt{IA}_{/\!/}$

We rely on the cartesian closed $\sim$-category $\mathbf{NTCG}_!$ from Part II (see Corollary 8.3.13). Thus, building on the standard interpretation of the simply-typed λ-calculus, it remains to give an interpretation to all basic datatypes and primitives of $\mathtt{IA}_{/\!/}$.

9.1.1 Interpretation of $\mathsf{PCF}_{/\!/}$

Interpretation of types. The cartesian closed structure of $\mathbf{NTCG}_!$ fixes the interpretation of the arrow constructor, which we must complete with an interpretation for the ground types $\mathbb{B}, \mathbb{N}$ and $\mathbb{U}$. Earlier we have already provided $-$-boards B, N and U for those ground types, with for instance, for the booleans, the $-$-board: with for instance

B =



In the sequel, we shall refer to these as the **small** boards for ground types. The small ground boards are very natural, but in tension with the fact that in the play-based games

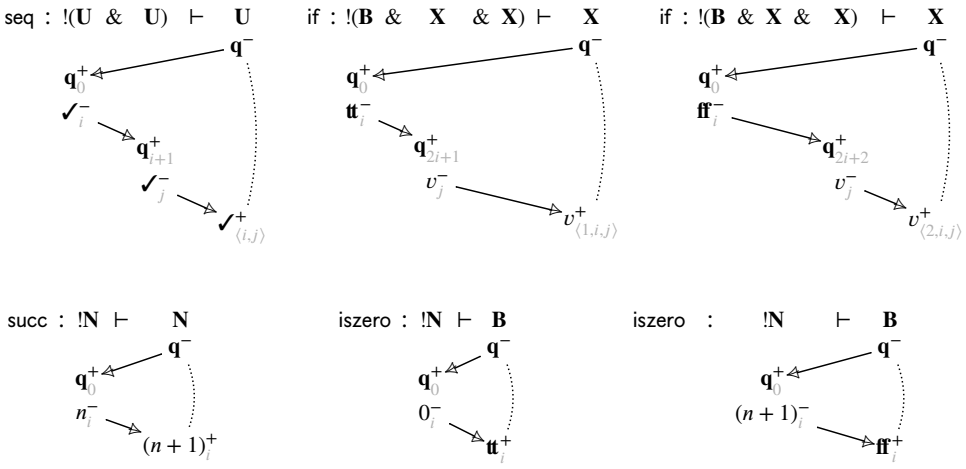
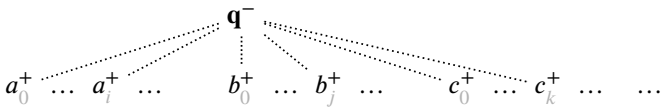


Figure 9.1: Configurations of basic strategies for PCF

of Part I, a question may be answered multiple times (possibly with the same answer). So as to avoid a cumbersome mismatch[1], we introduce an alternative interpretation of base types, where all possible answers are replicated. This is handled by:

Definition 9.1.1. *Consider $V = \{a, b, c, \dots\}$ an at most countable set of* values*.*
We define the strict $-$-board ground(V) *as having essp:*



with symmetries all bijections only changing copy indices. All symmetries are positive; negative symmetries are restricted to identities. Finally, the payoff function is:

$$\begin{array}{rcccl} \kappa & : & \mathscr{C}(\mathsf{ground}(V)) & \to & \{-1,0,+1\} & \\ & & \emptyset & \mapsto & +1 & \\ & & \{\mathbf{q}^-, v_i^+\} & \mapsto & 0 & \\ & & x & \mapsto & -1 & \textit{otherwise} \end{array}$$

Each value has countably many copies, with no conflict. Thus, when playing a value, a strategy must provide an accompanying copy index. Player may answer *several times*, either with the same or different values, but this is banned from strategies by the *winning* mechanism. However, Opponent may still answer several times (if the board appears

[1] In non-alternating games, (logical) well-bracketing corrects this. The mismatch occurs only in the upcoming alternating unfolding, as in the alternating case well-bracketing is on strategies rather than on plays.

$$
\begin{array}{rcl}
[\![M;\ N]\!] & = & \mathsf{seq} \odot_! \langle [\![M]\!], [\![N]\!] \rangle \\
[\![\mathbf{if}\ M\ N_1\ N_2]\!] & = & \mathsf{if} \odot_! \langle [\![M]\!], [\![N_1]\!], [\![N_2]\!] \rangle \\
[\![\mathbf{succ}\ M]\!] & = & \mathsf{succ} \odot_! [\![M]\!] \\
[\![\mathbf{pred}\ M]\!] & = & \mathsf{pred} \odot_! [\![M]\!] \\
[\![\mathbf{iszero}\ M]\!] & = & \mathsf{iszero} \odot_! [\![M]\!] \\
[\![\mathbf{let}\ x = N\ \mathbf{in}\ M]\!] & = & \mathsf{let}_{\mathbf{X},\mathbf{Y}} \odot_! \langle [\![N]\!], \Lambda^!([\![M]\!]) \rangle
\end{array}
$$

Figure 9.2: Basic interpretation clauses

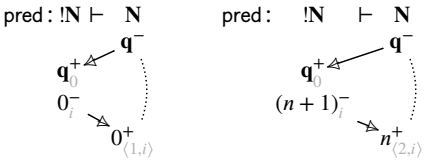


Figure 9.3: Strategy for **pred**

in contravariant position) so that the model conveys information about the behaviour of programs under a context that answers several times.

We define the **large** ground boards as $\mathbf{U} = \mathsf{ground}(\{\checkmark\})$, $\mathbf{B} = \mathsf{ground}(\{\mathbf{tt}, \mathbf{ff}\})$ and $\mathbf{N} = \mathsf{ground}(\mathbb{N})$ (the set of all natural numbers). As explained above, this extends to all *types* of PCF by setting $[\![A \to B]\!] = ![\![A]\!] \multimap [\![B]\!]$ using the constructions from Section 8.2.2, yielding for any type A a strict $-$-board $[\![A]\!]$. Finally, a *context* $\Gamma = x_1 : A_1, \ldots, x_n : A_n$ is interpreted as $[\![\Gamma]\!] = \&_{1 \le i \le n} [\![A_i]\!]$. We insist that $\mathtt{IA}_{/\!/}$ can be interpreted with the ground types set as $\mathbf{U}, \mathbf{B}$ and $\mathbf{N}$; or as U, B and N. We set the large boards as our default choice so as to match with non-alternating games, but we may sometimes refer to the small as well, because it allows for more succinct examples.

Interpretation of terms. A term $\Gamma \vdash M : A$ of PCF is interpreted as a strategy

$$[\![M]\!] \in \mathbf{NTCG}_!([\![\Gamma]\!], [\![A]\!]) .$$

We invoke the cartesian closed structure of $\mathbf{NTCG}_!$ – writing $\odot_!$ for Kleisli composition, and $\Lambda^!$ for Kleisli currying. We skip the details of the interpretation of the λ-calculus combinators, which follows the standard lines of the interpretation of the simply-typed λ-calculus in a cartesian closed category [Lambek and Scott, 1988].

We must specify strategies for the term constructions that PCF adds to the simply-typed λ-calculus. For constants, we set $[\![\mathbf{skip}]\!] : \mathbf{U}$, $[\![\mathbf{tt}]\!] : \mathbf{B}$, $[\![\mathbf{ff}]\!] : \mathbf{B}$ and $[\![n]\!] : \mathbf{N}$ as the obvious strategies replying immediately the corresponding value (with copy index 0). For all other combinators, the interpretation is in Figure 9.2. The strategies used are described in Figures 9.1, 9.3 and 9.4. The notation we employ is symbolic: the strategies comprise a branch for each instantiation of the copy index parameters i, j and the value parameters $v, w, \ldots$. For instance, the strategy if has as configuration that shown in Figure 9.5 as nothing prevents Opponent from answering a question multiple times – this is in contrast with $\circlearrowleft$-**Strat**, where plays were always assumed to be well-bracketed. Recall also from Section 8.3.1 that $\langle -, \ldots, - \rangle : \mathbb{N}^n \simeq \mathbb{N}$ denotes any bijection. Strategies specified symbolically as above carry a specific choice determining the copy indices of Player moves from those of Opponent moves. This choice does not matter up to positive isomorphism; the only thing that matters is that there is never any collision so that the display maps do satisfy the conditions of a concurrent strategy.

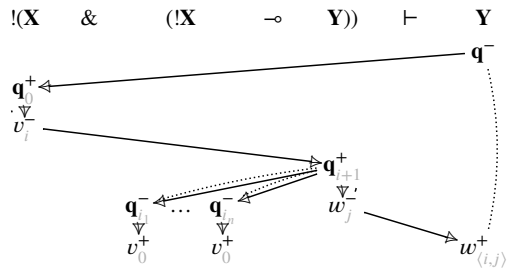


Figure 9.4: Typical configuration of **let**

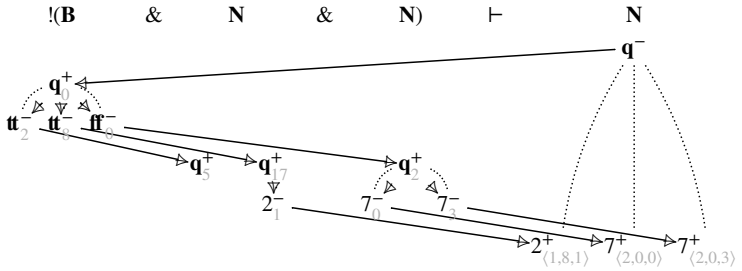


Figure 9.5: Example of a configuration of if

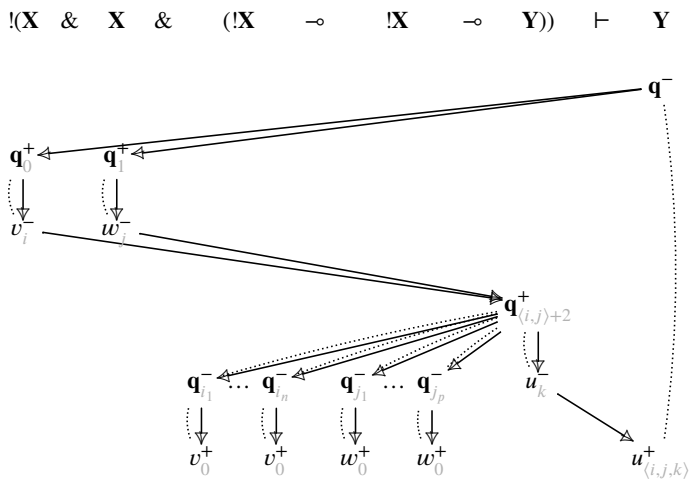


Figure 9.6: Typical configuration of **plet**

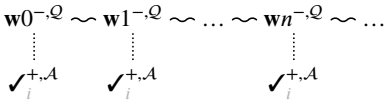


Figure 9.7: $\mathbf{V}_w$

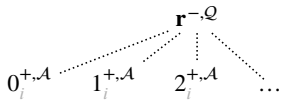


Figure 9.8: $\mathbf{V}_r$

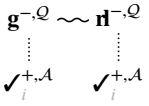


Figure 9.9: $\mathbf{S}$

For **let**, the strategy implements a *memoization* mechanism: it evaluates on $\mathbf{X}$ first obtaining a value v, which it feeds to the function argument each call. The interpretation is completed with $[\![\mathcal{Y}_A\ M]\!] = \mathcal{Y}_{\Gamma,A} \odot_! [\![M]\!]$ for $\Gamma \vdash M : A \to A$, using (8.9).

Semantics of parallelism**.** This is extended to PCF$_{/\!/}$ simply by adding the clause

$$[\![\Gamma \vdash \mathbf{let} \begin{pmatrix} x_1 & = & N_1 \\ x_2 & = & N_2 \end{pmatrix} \mathbf{in}\ M : \mathbb{Y}]\!] = \mathbf{plet}_{\mathbf{X},\mathbf{Y}} \odot_! \langle [\![N_1]\!], [\![N_2]\!], \Lambda^!(\Lambda^!([\![M]\!])) \rangle$$

via the strategy $\mathbf{plet}_{\mathbf{X},\mathbf{Y}}$ with typical configurations as presented in Figure 9.6. It is interesting to compare this definition with the non-alternating plays in Figure 5.11: the diagram avoids listing all interleavings, hence staying much more faithful to the intuition behind the behaviour of this primitive.

9.1.2 Semantics of state

The interpretation of references and semaphores closely follows that in (⇃↾-**Strat** and) ↺-**Strat**. For the types $\mathbb{V}$ and $\mathbb{S}$, we first introduce strict $-$-boards $\mathbf{V}_w$, $\mathbf{V}_r$ and $\mathbf{S}$ as

$$\mathbf{V}_w = \bigwith_{n \in \mathbb{N}} \mathbf{U}, \qquad \mathbf{V}_r = \mathbf{N}, \qquad \mathbf{S} = \mathbf{U} \& \mathbf{U}$$

with moves renamed as illustrated in Figures 9.7 (where in $\mathbf{V}_w$, all write requests are in pairwise conflict), 9.8 and 9.9. We set $\mathbf{V} = \mathbf{V}_w \& \mathbf{V}_r$, and $[\![\mathbb{V}]\!] = \mathbf{V}$, $[\![\mathbb{S}]\!] = \mathbf{S}$.

Stateless primitives. As in Section 4.2.2, only the interpretation of the primitives **newref** and **newsem** is effectful – other primitives only send various queries to references and semaphores, and as such are simple innocent strategies.

Accordingly, we set the interpretation of memory and semaphore accesses as:

$$\begin{aligned} [\![M := N]\!] &= \mathsf{assign} \odot_! \langle [\![N]\!], [\![M]\!] \rangle \\ [\![!M]\!] &= \mathsf{deref} \odot_! [\![M]\!] \\ [\![\mathbf{grab}(M)]\!] &= \mathsf{grab} \odot_! [\![M]\!] \\ [\![\mathbf{release}(M)]\!] &= \mathsf{release} \odot_! [\![M]\!] \end{aligned}$$

using the strategies of Figures 9.10, 9.11, 9.12, and 9.13. Finally, we set

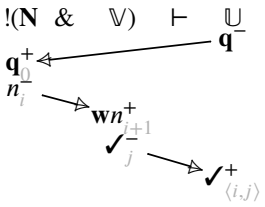


Figure 9.10: assign

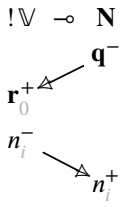


Figure 9.11: deref

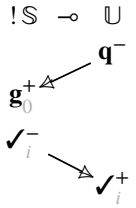


Figure 9.12: grab

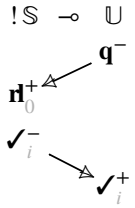


Figure 9.13: release

$$[\![\mathbf{mkvar}\ M\ N]\!] = \langle \langle [\![M]\!]\, n \mid n \in \mathbb{N} \rangle, [\![N]\!] \rangle\,, \qquad [\![\mathbf{mksem}\ M\ N]\!] = \langle [\![M]\!], [\![N]\!] \rangle\,,$$

where $[\![M]\!]\, n$ is $[\![M]\!]$ applied to the constant strategy n (using the cartesian closed structure of $\mathbf{NTCG}_!$). It remains to define the semantics of **newref** and **newsem**.

Creation of reference and semaphores. For **newref** and **newsem**, again we follow in the footsteps of Section 4.2.2. However, in contrast with respect to Section 4.2.2, we must manage copy indices so as to satisfy the constraints of concurrent prestrategies.

For each $I \subseteq_f \mathbb{N}$ and $n \in \mathbb{N}$, we define languages of infinite words on $!\mathbf{V}$ and $!\mathbf{S}$:

$$\begin{array}{rcll} \mathsf{cell}_n^I & = & \mathbf{r}_i^- \cdot n_0^+ \cdot \mathsf{cell}_n^{I \uplus \{i\}} \mid \mathbf{w}k_i^- \cdot \checkmark_0^+ \cdot \mathsf{cell}_k^{I \uplus \{i\}} & (i \notin I) \\ \mathsf{lock}_0^I & = & \mathbf{g}_i^- \cdot \checkmark_0^+ \cdot \mathsf{lock}_1^{I \uplus \{i\}} \mid \mathbf{rl}_i^- & (i \notin I) \\ \mathsf{lock}_n^I & = & \mathbf{g}_i^- \mid \mathbf{rl}_i^- \cdot \checkmark_0^+ \cdot \mathsf{lock}_0^{I \uplus \{i\}} & (i \notin I,\ n > 0) \end{array}$$

where symbols are moves in $!\mathbf{V}$ and $!\mathbf{S}$ respectively, separated via $\cdot$ for readability. Note that positive moves should really include *two* copy indices: one for the outer bang, one for the value – we show only the latter, the former is always the same as the previous negative move. Here, $I \subseteq_f \mathbb{N}$ collects the copy indices already used so that each new copy index is fresh, ensuring that no single move appears twice. We set cell_n as (the prefix language of) $\mathsf{cell}_n^\emptyset$ and lock_n as (the prefix language of) $\mathsf{lock}_n^\emptyset$. They may be viewed as event structures by setting events to be finite words, causality the prefix ordering, and setting any incomparable words to be in conflict. The two display maps

$$\begin{array}{rcccc} \partial_{\mathsf{cell}} & : & |\mathsf{cell}| & \to & !\mathbf{V} \\ & & sa & \mapsto & a \end{array} \qquad \begin{array}{rcccc} \partial_{\mathsf{lock}} & : & |\mathsf{lock}| & \to & !\mathbf{S} \\ & & sa & \mapsto & a \end{array}$$

only keep the last move. As configurations of cell and lock are sets of prefixes of a branch which is an alternating play in $!\mathbf{V}$ or $!\mathbf{S}$, we set $\mathcal{S}(\mathsf{cell}_n)$ to comprise those bijections induced by plays $s_1 \cong_A s_2$ symmetric on A (for A being $!\mathbf{V}$ or $!\mathbf{S}$, see Definition 7.1.14).

Altogether, we get:

Proposition 9.1.2. *We have two concurrent prestrategies* $\mathsf{cell} : !\mathbf{V}$ *and* $\mathsf{lock} : !\mathbf{S}$.

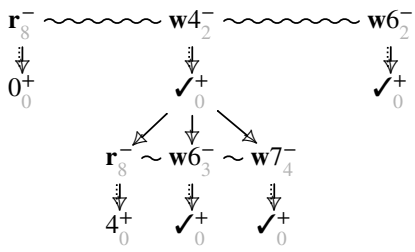


Figure 9.14: Beginning of cell_0

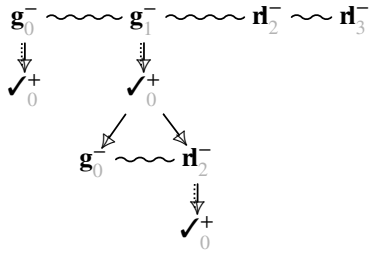


Figure 9.15: Beginning of lock_0

We display a few early moves, of cell in Figure 9.14 and lock in Figure 9.15, with the convention that all moves in the same row are in pairwise conflict. We observe that both prestrategies fail *courtesy*: the immediate causal link $\checkmark^+_0 \rightarrowtriangle \mathbf{r}^-_8$, for instance, would be illegal for a strategy. Note that for semaphores, in Figure 9.15, trailing Opponent moves are indeed maximal, Player has no response – an attempt to release a lock that has not been grabbed, or to grab a lock that has not been released, triggers no response.

Note that although $\mathsf{cell} : !\mathbf{V}$ is only a prestrategy and not a strategy, we have:

Proposition 9.1.3. *Consider* $\sigma : !\mathbf{V} \vdash A$ *and* $\tau : !\mathbf{S} \vdash A$.
Then, the two prestrategies $\sigma \odot \mathsf{cell} : A$ *and* $\tau \odot \mathsf{lock} : A$ *are strategies.*

Proof. We show it for cell, the proof for lock is the same. First, while cell is not courteous, it is *span-courteous* in the sense of Definition 7.3.10. Thus, by Proposition 6.2.22, $\sigma \odot \mathsf{cell}$ is well-defined and a prestrategy. It remains to prove that $\sigma \odot \mathsf{cell}$ is receptive and courteous. But by Proposition 6.4.12, that is the case if and only if it is invariant (up to strong iso) under composition with copycat. Thus we compute:

$$\begin{aligned} \mathfrak{C}_A \odot (\sigma \odot \mathsf{cell}) &\cong (\mathfrak{C}_A \odot \sigma) \odot \mathsf{cell} \\ &\cong \sigma \odot \mathsf{cell}\,, \end{aligned}$$

so by Proposition 6.4.12, $\sigma \odot \mathsf{cell}$ is receptive and courteous as required. □

It is then easy to deduce the following fact:

Corollary 9.1.4. *Consider* $\sigma \in \mathbf{NTCG}(\mathbf{V}, A)$ *and* $\tau \in \mathbf{NTCG}(\mathbf{S}, A)$.
Then, $\sigma \odot \mathsf{cell} \in \mathbf{NTCG}(\mathbf{1}, A)$ *and* $\tau \odot \mathsf{lock} \in \mathbf{NTCG}(\mathbf{1}, A)$.

Proof. Again, we prove it for $\sigma \odot \mathsf{cell}$. According to Proposition 8.2.15, we must show that it is *negative* and *winning* in the sense of Definition 8.2.14.

Negative is trivial as A is negative. For *winning*, take $x^\sigma \odot x^{\mathsf{cell}} \in \mathscr{C}^+(\sigma \odot \mathsf{cell})$. Then, $x^\sigma \in \mathscr{C}(\sigma)$ must be $+$-maximal, otherwise it immediately contradicts $+$-maximality of $x^\sigma \odot x^{\mathsf{cell}}$ or minimality of $(x^{\mathsf{cell}}, x^\sigma)$ in the sense of Proposition 7.3.9. Now in x^{cell},

maximal events (if any exists) must be positive, or contradict minimality of $(x^{\mathsf{cell}}, x^{\sigma})$. But thus $\kappa_{!\mathbf{V}}(x^{\mathsf{cell}}_{!\mathbf{V}}) = 0$, hence $\kappa_A(x^{\sigma}_A) \geq 0$ follows by $\boldsymbol{\sigma}$ winning. □

We now complete the interpretation. Consider $\Gamma, x : \mathbb{V} \vdash M : \mathbb{X}$ with

$$[\![M]\!] \in \mathbf{NTCG}_!(\Gamma \& \mathbb{V}, \mathbb{X})$$

omitting some brackets. Using the cartesian closed structure of $\mathbf{NTCG}_!$, we consider

$$\Lambda^!_\Gamma([\![M]\!]) \in \mathbf{NTCG}(!\,\mathbb{V}, !\,\Gamma \multimap \mathbb{X})$$

which we compose with the memory cell. Summing up, for references and semaphores,

$$\begin{array}{rcl} [\![\mathbf{newref}\ x := n\ \mathbf{in}\ M]\!] & = & {\Lambda^!_\Gamma}^{-1}(\Lambda^!_\Gamma([\![M]\!]) \odot \mathsf{cell}_n) \quad \in \quad \mathbf{NTCG}_!(\Gamma, \mathbb{X}) \\ [\![\mathbf{newsem}\ x := n\ \mathbf{in}\ M]\!] & = & {\Lambda^!_\Gamma}^{-1}(\Lambda^!_\Gamma([\![M]\!]) \odot \mathsf{lock}_n) \quad \in \quad \mathbf{NTCG}_!(\Gamma, \mathbb{X})\,. \end{array}$$

concluding the interpretation of $\mathtt{IA}_{/\!/}$ in **NTCG**. It shall follow from this chapter that:

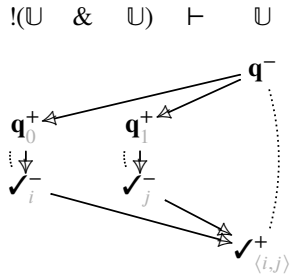
Theorem 9.1.5. $\mathbf{NTCG}_!$ *is adequate for* $\mathtt{IA}_{/\!/}$.

Proof. A consequence of the adequacy of $\circlearrowleft$-**Strat** for $\mathtt{IA}_{/\!/}$ with Theorem 9.3.9. □

9.1.3 Discussion on the Interpretation

What does the interpretation of $\mathtt{IA}_{/\!/}$ programs as concurrent strategies look like?

At first sight, the information on programs of $\mathtt{IA}_{/\!/}$ that the interpretation exposes is conceptually close to that in $\circlearrowleft$-**Strat**: it is a representation of its interactive behaviour. But the causal information makes it more explicit: for instance, the concurrent strategy for the parallel composition operation is simply described by the diagram



which is very close to the *intention* behind the many interleavings of the corresponding non-alternating strategy. Of course that example is somewhat expected: it is the canonical example illustrating what is gained by moving from an *interleaving* to a *truly concurrent* model of concurrency. So beyond this example, let us consider various fragments of $\mathtt{IA}_{/\!/}$ and provide examples illustrating the expressiveness of the model.

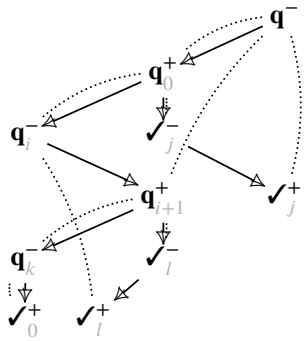
Sequential innocent strategies. Though the model is designed to support the interpretation of rich, effectful concurrent programs, it is informative to consider first its interpretation of simple, pure sequential programs.

For instance, the interpretation of the following simple program:

$$\vdash \lambda f^{\mathbb{U}\to\mathbb{U}}.f\,(f\,\mathbf{skip}) : (\mathbb{U}\to\mathbb{U})\to\mathbb{U}\to\mathbb{U}$$

yields the strategy whose behaviour is described, in symbolic notation, by

$$!(!\mathbf{U} \multimap \mathbf{U}) \multimap \mathbf{U}$$



Ignoring indices, one may recognize the shape of *meager innocent strategy* from Definition 3.2.12, *i.e.* the forest of P-views – this is behind the link with sequential innocent strategies in Section 11.3. Accounting for the replications implicit in the symbolic notation, the full strategy has branches for all values of i, j, k, l, making the strategy analogous to a normal form with replicated branches as in the *resource calculus*.

This shows that the traditional notion of *P-view* is really a way to recover the causal structure for sequential, purely functional programs. Concurrent games push this handle on causality way beyond purely functional sequential programs.

Causality of effectful programs. For pure sequential programs, causality simply conveys syntactic dependency in the sense of imbrication within the syntax tree. But what does it express for effectful programs? We consider the program

$$\vdash \mathbf{newref}\ x\ \mathbf{in}\ \lambda f^{\mathbb{U}\to\mathbb{U}}.\, f\,(x := 1);\ \mathbf{not}\,(\mathbf{iszero}\,!x) : (\mathbb{U}\to\mathbb{U})\to\mathbb{B}$$

that implements a sort of *strictness testing*: if f answers without evaluating its argument, then the read is triggered and returns the default value of the memory cell, 0 – hence the program returns **ff**. If f does evaluate its argument before returning, then the program returns **tt**. The interpretation yields a strategy which comprises as induced sub-event structure that in Figure 9.16 where copy indices are indicative – unfolding the exact definition of the interpretation might yield different choices. The full strategy is of course infinite, taking into account all possible replications by Opponent.

This showcases a few important phenomena:

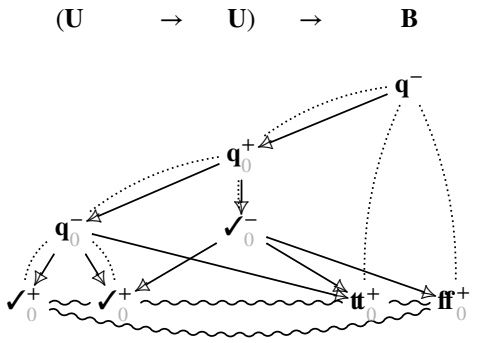
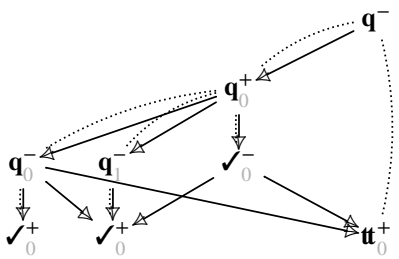


Figure 9.16: A prefix of the strategy for an effectful program

First of all, the example has conflict, *i.e.* non-determinism. Why is that, since the program above is in IA, a deterministic language? In fact, IA is only deterministic when observed by a sequential context: here, the model is more permissive and allows the argument function to simultaneously return and call its argument. This triggers a race in memory, which may be resolved in two incompatible ways: the write may win (leading to value **tt**), or the read may win (leading to value **ff**). Accordingly, the diagram has two maximal configurations, corresponding to the two resolutions of this race.

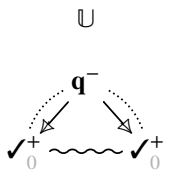
Secondly, beyond merely describing the syntactic tree, causality also flows through memory: $\mathbf{tt}^+_0$ depends on $\mathbf{q}^-_0$ as it can only happen after reading a value written by the subterm whose evaluation is triggered by $\mathbf{q}^-_0$. In contrast, $\mathbf{ff}^+_0$ only depends on $\checkmark^-_0$. This may seem strange, because it does not seem to imply that the read *must* happen before the write for us to observe $\mathbf{ff}^+_0$. But $\mathbf{ff}^+_0$ is only compatible with the copy of the write acknowledgement $\checkmark^+_0$ that *does* depend on $\checkmark^-_0$ (which triggers the read).

When drawing strategies for impure programs we avoid the symbolic representation used before, as it is very misleading. If Opponent duplicates $\mathbf{q}^-_0$ in the diagram above, this is not going to "only" duplicate the moves that depend on it: there will now be two read requests and one write requests, leading to $3! = 6$ ways to resolve this three-party race. This will give rise to a complex pattern of conflicts too unwieldy to be pleasantly drawn in a single diagram, allowing 6 maximal configurations. One of them will be

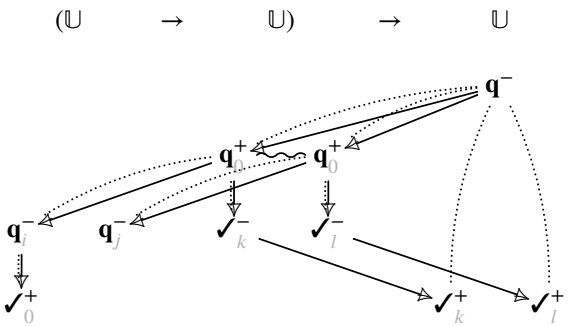


in which by squinting appropriately, the trained eye will recognize an execution where the write triggered by $\mathbf{q}^+_0$ was treated first, then the read, then the write triggered by $\mathbf{q}^+_1$ – the only possible alternating linearization of this diagram follows this order.

Non-determinism. The example above also shows that a strategy may have two non-deterministic events with the same label. It is crucial that in this model, *conflict is not idempotent*: for instance, $\vdash M = \mathbf{if\ choice\ then\ skip\ else\ skip} : \mathbb{U}$ yields a strategy



with two completely indistinguishable copies of the same computational event $\checkmark^+_0$. Intuivively, there are two events because there are two derivations of $M \Downarrow \mathbf{skip}$ – each event carries its full causal history. Keeping these events separate is sometimes important, because they may have different futures. For instance, in the strategy



the two events $\mathbf{q}^+_0$ look similar, but one will feed a value to the argument function while the other will not, as in the term $\vdash \lambda f.\, \mathbf{if\ choice\ then}\ f\ \mathbf{skip\ else}\ f\ \bot$.

One may wonder if this very fine-grained non-deterministic branching information makes the model sound with respect to must-testing. The answer is: not directly, because non-deterministic branches not yielding any observable result are erased when performing the hiding. However, hiding can be amended to correct that: this is done in [Castellan et al., 2018a] for affine games – we shall however not follow that route here.

Observable behaviour. It should be clear from the examples above that the model is extremely intensional; it displays information that reflects the internal workings of the program and is by no means observable. A natural question is then: what parts of the concurrent strategy interpreting an $\mathtt{IA}_{/\!/}$ program is observable within $\mathtt{IA}_{/\!/}$? We shall provide an answer to that next, by unfolding to non-alternating game semantics.

9.2 Unfolding Causal to Non-Alternating Strategies

For the first time, we must simultaneously consider two kinds of strategies: *concurrent* strategies, as in, event structures; and non-alternating strategies – which are, of course, also concurrent! To avoid confusion, we shall say *causal* (pre)strategies for event structure based (pre)strategies, and *non-alternating* (pre)strategies otherwise.

Play-based strategies present programs by listing all their observable behaviour, presented as chronologically ordered *plays*. Causal strategies tell us more: *why* an action was performed, which actions are *independent*, also keeping track separately of nondeterministic branches. Intuitively, it is clear how to go from the former to the latter: simply forget the causal structure and retain only the induced linear orderings.

While conceptually clear, doing this concretely will require us to address a number of small but significant mismatches between the two settings.

9.2.1 A First Unfolding

Plays on games with symmetry. We start with linearizations of event structures.

Definition 9.2.1. *Consider E an event structure.*
*A **linearization** on E is a sequence $s = s_1 \dots s_n$ which is:*

$$\begin{array}{ll} \text{valid:} & \forall 1 \leq i \leq n,\ \{s_1, \dots, s_i\} \in \mathscr{C}(A)\,, \\ \text{non-repetitive:} & \forall 1 \leq i, j \leq n,\ s_i = s_j \ \Rightarrow\ i = j\,. \end{array}$$

We write $\mathscr{L}(E)$ for the set of linearizations on E.

If A is a board, then we also call $s \in \mathscr{L}(A)$ a **non-alternating play** on A, and write $\circlearrowleft$-Plays$(A) = \mathscr{L}(A)$. Now recall that a board also carries a notion of *symmetry* (see Section 7.1.1). As in Definition 7.1.14, this induces an equivalence on linearizations:

Definition 9.2.2. *Let E be an ess and $s, t \in \mathscr{L}(E)$. We say that s and t are **symmetric**, written $s \cong_E t$, if s and t have the same length, and we have*

$$\theta^j_{s,t} = \{(s_i, t_i) \mid 1 \leq i \leq j\} : \{s_1, \dots, s_j\} \cong_E \{t_1, \dots, t_j\}$$

a symmetry in $\mathscr{S}(E)$ for all $1 \leq j \leq n$; writing $s = s_1 \dots s_n$ and $t = t_1 \dots t_n$.

It is straightforward to establish:

Lemma 9.2.3. *Consider A a board.*
Then, the relation $\cong_A$ on $\mathscr{L}(A)$ is a length-preserving equivalence satisfying:

$$\begin{array}{rl} \text{preservation:} & \text{if } s_1 \dots s_n \cong_A t_1 \dots t_p, \text{ then for all } 1 \leq i \leq n,\ \mathrm{pol}_A(s_i) = \mathrm{pol}_A(t_i), \\ \text{restriction:} & \text{if } s_1 \dots s_n \cong_A t_1 \dots t_n, \text{ then for all } 1 \leq i \leq n,\ s_1 \dots s_i \cong_A t_1 \dots t_i, \\ \text{simulation:} & \text{if } s \cong_A t \text{ and } sa \in \mathscr{L}(A), \\ & \text{then there is } tb \in \mathscr{L}(A) \text{ such that } sa \cong_A tb. \end{array}$$

Proof. Straightforward from Definitions 7.1.5 and 9.2.2. □

We shall rely on this to define our unfolding.

Linearizing strategies. As a first step towards the unfolding, we shall extract from any causal strategy σ on board A a subset of $\circlearrowleft$-Plays(A) – a "non-alternating strategy on A", though no such formal object has been defined[2].

To do so, our main observation is the following:

Lemma 9.2.4. *Consider $\sigma : A$ a causal strategy on board A. Then we have:*

$$\begin{array}{rcccl} \partial_\sigma & : & \mathscr{L}(\sigma) & \to & \mathscr{L}(A) = \circlearrowleft\text{-Plays}(A) \\ & & \varepsilon & \mapsto & \varepsilon \\ & & sm & \mapsto & \partial_\sigma(s)\partial_\sigma(m)\,. \end{array}$$

a length-preserving monotone function.

Proof. It is immediate from *rule-abiding* and *locally injective* that this is well-defined. Monotonicity and preservation of length are by definition. □

The overload of notations for ∂_σ should not cause any confusion.

The *pre-unfolding* of σ is then obtained simply by projecting $\mathscr{L}(\sigma)$ – all linearizations of σ – through ∂_σ. More precisely, we prove the following proposition:

Proposition 9.2.5. *Consider $\sigma : A$ a causal prestrategy on board A. Then, the set*

$$\circlearrowleft\text{-Plays}(\sigma) = \partial_\sigma(\mathscr{L}(\sigma))$$

satisfies the following properties:

non-empty: $\varepsilon \in \circlearrowleft$-Plays$(\sigma)$,
prefix-closed: *for all $t \in \circlearrowleft$-Plays(σ), is $s \sqsubseteq t$, then $s \in \circlearrowleft$-Plays(σ),*

and furthermore, if σ is a strategy, then we additionally have:

receptive: *if $s \in \circlearrowleft$-Plays(σ) and $sa^- \in \mathscr{L}(A)$, then $sa \in \circlearrowleft$-Plays(σ),*
courteous: *if $sabt \in \circlearrowleft$-Plays$(\sigma)$ and* pol$(a) = +$ *or* pol$(b) = -$,
then if $sbat \in \mathscr{L}(A)$, $sbat \in \circlearrowleft$-Plays$(\sigma)$.

Proof. Non-empty, prefix-closed. Obvious since $\varepsilon \in \mathscr{L}(\sigma)$, $\mathscr{L}(\sigma)$ is prefix-closed and ∂_σ is monotone and length-preserving.

Receptive. Consider $s \in \circlearrowleft$-Plays(σ) and $sa^- \in \mathscr{L}(A)$. By definition, there is $u \in \mathscr{L}(\sigma)$ s.t. $s = \partial_\sigma(u)$. By definition of plays, $|u| \in \mathscr{C}(\sigma)$ and $\partial_\sigma(|u|) = |s|$ extends with a^-; so by *receptive* there is $m \in |\sigma|$ s.t. $|u|$ extends with m and $\partial_\sigma\, m = a$. Hence, $um \in \mathscr{L}(\sigma)$ and $\partial_\sigma(um) = sa$, so that $sa \in \circlearrowleft$-Plays(σ) as required.

Courteous. Consider $sabt \in \circlearrowleft$-Plays$(\sigma)$ and assume pol$(a) = +$ or pol$(b) = -$. By definition, there is $umnv \in \mathscr{L}(\sigma)$ such that $\partial_\sigma(u) = s$, $\partial_\sigma(m) = a$, $\partial_\sigma(n) = b$ and $\partial_\sigma(v) = t$. Now, of course we cannot have $n <_\sigma m$. If m, n are incomparable, then $unmv \in \mathscr{L}(\sigma)$ as well as needed. Otherwise, $m <_\sigma n$. But then, we must have $m \rightarrowtriangle_\sigma n$. Now, since pol$(m) = +$ or pol$(n) = -$, by *courteous* we have $a \rightarrowtriangle_A b$ as well, contradicting $sbat \in \mathscr{L}(A)$. □

[2] Recall that non-alternating strategies are sets of plays with pointers on arenas, not sets of plays on boards.

Note that these properties are essentially those defining a non-alternating strategy (Definition 5.1.5), though based on $\mathscr{L}(A)$ with A a board rather than based on plays with pointers on an arena, *i.e.* as in a non-alternating version of AJM games.

It is tempting to attempt defining such a non-alternating version of AJM games, and set it as target for the unfolding of causal strategies. After all, thin concurrent games and AJM games rely on a similar mechanism for handling replication, via copy indices and symmetry, whereas $\mathcal{O}$-**Strat** rests on plays with pointers. However, it turns out that the AJM approach to uniformity breaks down beyond alternating deterministic strategies. An appealing alternative is suggested by Melliès' *orbital games* [Melliès, 2003], but this has yet to be worked out. In this monograph, the non-alternating pre-unfolding above will only be an intermediary towards the unfolding to plays with pointers.

Notations. In this section, the application of the display map to plays

$$\partial_{\sigma} : \mathscr{L}(\sigma) \to \mathcal{O}\text{-}\mathsf{Plays}(\sigma) \subseteq \mathcal{O}\text{-}\mathsf{Plays}(A)\,,$$

for a causal $\sigma : A$, is a central concept – it is thus worth introducing some specialized notations (inspired from our notations on configurations) to reason on such situations.

As usual, plays are ranged over via s, t, u, v, etc; while single moves are ranged over by a, b, m, n etc. For plays in a game, we often add a tag in subscript as in $s_A, t_A \in \mathcal{O}\text{-}\mathsf{Plays}(A)$. For linearizations in a causal strategy, we add a similar tag as a superscript, with *e.g.* $s^{\sigma}, t^{\sigma} \in \mathscr{L}(\sigma)$. In that case, we write $s^{\sigma}_A \in \partial_{\sigma}(s^{\sigma}) \in \mathcal{O}\text{-}\mathsf{Plays}(A)$.

9.2.2 Mixed Boards and Pointifixion

The above already gives a notion of unfolding of a causal strategy $\sigma : A$ on a board A as a set of non-alternating plays on A. However, the non-alternating game semantics of Chapter 5 is based on *plays with pointers*. How shall we link the two?

A causal strategy plays on a *board*, whereas non-alternating plays operate on *arenas*, which represent types without duplications. In principle, the arena may be obtained by quotienting the board, but it is not clear how to formalize this elegantly. So instead, we interpret types with a structure that stores *both* the arena and the board, along with the links between the two. Besides connecting the two structures, this will serve as a description of the "concrete" boards, that arise as the interpretation of types.

Mixed boards. This is captured by the notion of *mixed boards*:

Definition 9.2.6. *A **mixed board** is $(A, \underline{A}, \mathsf{lbl}_A)$ with A a strict board, $\underline{A}$ an arena, and*

$$\mathsf{lbl}_A : |A| \to |\underline{A}|$$

*a **label** function preserving polarities, satisfying the following requirements:*

rigid: lbl_A *preserves and reflects minimality, and preserves* $\rightarrowtriangle$,
transparent: *for any* $x, y \in \mathscr{C}(A)$ *and bijection* $\theta : x \simeq y$,
then $\theta \in \mathcal{S}(A)$ *iff* θ *is an order-iso preserving* lbl_A,

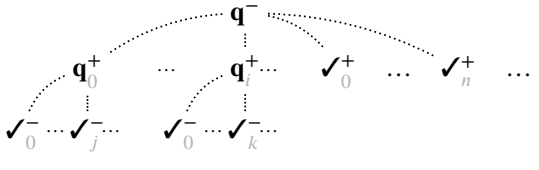


Figure 9.17: The strict board for $\mathbb{U} \to \mathbb{U}$

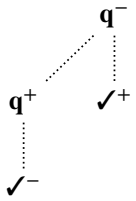


Figure 9.18: The arena

with additional components postponed until Section 12.1.1 (Definition 12.1.1).

Mixed boards make explicit how multiple moves in the board, kept apart by copy indices, match the same in the arena. Figures 9.17 and 9.18 together form a representative example, giving the mixed board interpreting the type $\mathbb{U} \to \mathbb{U}$, where the label function is clear. *Transparent* gives a concrete account of symmetries: as they must preserve the partial order and the label, they only exchange between interchangeable copies.

Constructions on mixed boards. All the basic arenas and boards used in the interpretation of ground types may be paired to yield mixed boards: for instance,

$$(\mathbf{X}, \underline{\mathbf{X}}, \mathsf{lbl}_{\mathbf{X}})$$

pairs the "large" boards for $\mathbb{U}, \mathbb{B}, \mathbb{N}$ from Section 9.1 with the basic arenas from Section 3.2.1, via the obvious labelling map – the mixed boards for references and semaphores are likewise straightforward. So as to lift this to the arrow type, we first need:

Lemma 9.2.7. *Consider A a mixed board. We extend lbl_A to*

$$\mathsf{lbl}_A : |!A| \to |\underline{A}|$$

by setting $\mathsf{lbl}_A(i, a) = \mathsf{lbl}_A(a)$. Then, we still have:

rigid: *lbl_A preserves minimality, and preserves immediate causality $\rightarrowtriangle$,*
transparent: *for any $x, y \in \mathscr{C}(!A)$ and bijection $\theta : x \simeq y$,*
then $\theta \in \mathscr{S}(!A)$ iff θ is an order-iso preserving lbl_A.

Proof. Rigid is obvious; we show *transparent*. *If.* Consider $x, y \in \mathscr{C}(!A)$ and a bijection $\theta : x \simeq y$, assume θ is an order-iso preserving lbl_A. Decompose x and y as

$$x = [x_i \mid i \in I] \qquad y = [y_j \mid j \in J]$$

via Lemma 8.3.1 – so that in particular for all $i \in I$, $x_i \neq \emptyset$ and for all $j \in J$, $y_j \neq \emptyset$. If I and J are empty then $\theta = \emptyset \in \mathscr{S}(!A)$. Otherwise, fix $i \in I$. As A is strict, x_i has exactly one initial move (i, a), take $\theta(i, a) = (j, b)$. Set $\pi(i) = j$; done for all $i \in I$ this yields $\pi : I \simeq J$, a bijection as θ induces a bijection between the initial moves of x and y. Now since θ is an order-iso, for all $(i, a') \in x$, we have $\theta(i, a') = (\pi(i), b')$ for

some $b' \in x_{\pi(i)}$. It is direct that θ restricts to an order-iso $\theta_i : x_i \simeq y_{\pi(i)}$ which still preserves labels. This concludes the proof that $\theta \in \mathcal{S}(!A)$ as required.

Only if. Straightforward. □

Via this lemma, all the remaining constructions on arenas extend transparently to mixed boards, pairing the arena with the matching board construction. In particular,

$$\begin{array}{rcl}(A, \underline{A}, \mathsf{lbl}_A) \,\&\, (B, \underline{B}, \mathsf{lbl}_B) & = & (A\&B, \underline{A}\&\underline{B}, \mathsf{lbl}_{A\&B}) \\ (A, \underline{A}, \mathsf{lbl}_A) \Rightarrow (B, \underline{B}, \mathsf{lbl}_B) & = & (!A \multimap B, \underline{A} \Rightarrow \underline{B}, \mathsf{lbl}_{A\Rightarrow B})\end{array}$$

where the labels for mixed board constructions are given by

$$\begin{array}{rcll}\mathsf{lbl}_{A_1\&A_2}(i,a) & = & (i, \mathsf{lbl}_{A_i}(a)) & \\ \mathsf{lbl}_{A\Rightarrow B}(2,b) & = & (2, \mathsf{lbl}_B(b)) & \\ \mathsf{lbl}_{A\Rightarrow B}(1,(i,a)) & = & (1, \mathsf{lbl}_A(a)) & (B \text{ well-opened}) \\ \mathsf{lbl}_{A\Rightarrow B}(1,(b,(i,a))) & = & (1,(b, \mathsf{lbl}_A(a))) & (B = \&_{i\in I} B_i \text{ with } B_i \text{ well-opened});\end{array}$$

we omit the immediate verifications that these satisfy the conditions for mixed boards.

Overall, this yields an interpretation of each $\mathtt{IA}_{/\!/}$ type (or context) A as a mixed board $[\![A]\!]$. As a convention, we shall still let A, B, C etc range over mixed boards, and silently coerce them to the board components (as in "let $a \in A$", etc). The arena component will be referred to explicitly via $\underline{A}$, $\underline{B}$, etc.

Pointifixion. Mixed boards are the adequate setting to link causal strategies with plays with pointers. First, any play on the board A yields a play with pointer on the arena $\underline{A}$:

Lemma 9.2.8. *Consider A a mixed board. Then, the function*

$$(\overset{\frown}{-}) : \circlearrowleft\text{-}\mathsf{Plays}(!A) \quad \to \quad \circlearrowleft\text{-}\mathsf{PrePlays}(\underline{A})$$

sending $s = s_1 \dots s_n \in \circlearrowleft\text{-}\mathsf{Plays}(A)$ to $\overset{\frown}{s}$ with components,

$$\begin{array}{rl}\text{moves:} & \textit{the sequence } t = t_1 \dots t_n \textit{ with } t_i = \mathsf{lbl}(s_i), \\ \text{pointers:} & \textit{for } 1 \le i < j \le n,\ t_j \textit{ points to } t_i \textit{ iff } s_i \rightarrowtriangle_A t_i.\end{array}$$

is well-defined.

Proof. For $s = s_1 \dots s_n \in \circlearrowleft\text{-}\mathsf{Plays}(!A)$, $\overset{\frown}{s}$ is $t_1 \dots t_n$, where t_j points to t_i iff $s_i \rightarrowtriangle_{!A} s_j$. We must show that every non-initial move has a pointer: if t_j is non-initial, then s_j is non-initial since lbl preserves minimality. So there is $a \rightarrowtriangle_{!A} s_j$. But since $s \in \circlearrowleft\text{-}\mathsf{Plays}(!A)$, $\{s_1, \dots, s_j\}$ is a configuration thus down-closed, so $a = s_i$ for some $1 \le i < j$. It follows that t_j has a pointer to t_i as required. □

This is illustrated in Figure 9.19: on the lhs, we show a non-alternating play on $!(!(!\mathbf{U} \multimap \mathbf{U}) \multimap \mathbf{B})$ – in this figure, each move carries a sequence of copy indices, from the outermost ! to the innermost[3]. On the rhs, we show the corresponding play

[3]This is different to the convention in most of this monograph where only the index for the innermost ! is shown, the others being recovered from the immediate dependency.

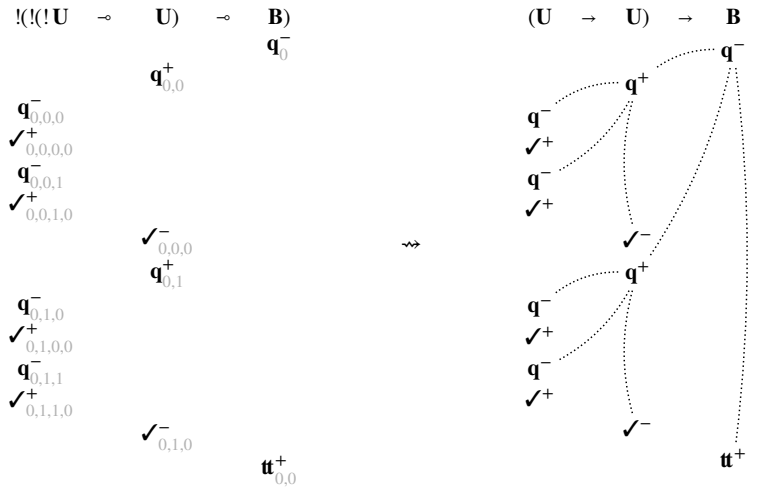


Figure 9.19: Recovering pointers

with pointers: it is obtained from the diagram on the lhs by making the immediate dependency from the game an explicit component of the play, then removing copy indices.

Crucially, this construction is invariant under symmetry:

Proposition 9.2.9. *Consider A a mixed board. Then, $(\overset{\curvearrowleft}{-})$ yields a function*

$$(\overset{\curvearrowleft}{-}) \quad : \quad \circlearrowleft\text{-Plays}(!A)/\cong_{!A} \quad \to \quad \circlearrowleft\text{-PrePlays}(\underline{A}),$$

injective and preserving length and prefix.

Proof. First, it preserves $\cong_{!A}$: if $s = s_1 \dots s_n \cong_{!A} t_1 \dots t_n = t$, then $\theta = \{(s_i, t_i) \mid 1 \leq i \leq n\} \in \mathcal{S}(!A)$; but by *transparent* (via Lemma 9.2.8), $\mathsf{lbl}(s_i) = \mathsf{lbl}(t_i)$ for all $1 \leq i \leq n$. Furthermore, by Lemma 7.1.6, θ is an order-isomorphism, so $s_i \rightarrowtriangle_A s_j$ iff $t_i \rightarrowtriangle_A t_j$ for all $1 \leq i < j \leq n$, hence $\overset{\curvearrowleft}{s}$ and $\overset{\curvearrowleft}{t}$ have the same pointers. Finally, the construction clearly preserves length and prefix as required.

For injectivity, take $s, t \in \circlearrowleft\text{-Plays}(!A)$ with $\overset{\curvearrowleft}{s} = \overset{\curvearrowleft}{t}$, writing $s = s_1 \dots s_n$ and $t = t_1 \dots t_n$. Then, for all $1 \leq i \leq n$, $\theta^i_{s,t} = \{(s_j, t_j) \mid 1 \leq j \leq i\}$ is a bijection between configurations of A. But since $\overset{\curvearrowleft}{s} = \overset{\curvearrowleft}{t}$, $\theta^i_{s,t}$ is actually an order-iso preserving lbl, so by *transparent* (via Lemma 9.2.8) we have $\theta_i \in \mathcal{S}(!A)$. Hence, $s \cong_{!A} t$. □

This proposition makes explicit the folklore idea that *plays with pointers* are concrete representatives for equivalence classes of plays with copy indices. Drawing inspiration from [Danos et al., 1996], we call $\overset{\curvearrowleft}{s}$ the **pointifixion** of s – note however that in [Danos et al., 1996] the pointers must be *reconstructed* from the copy indices, whereas here, they are already in s as the immediate causal links from the board.

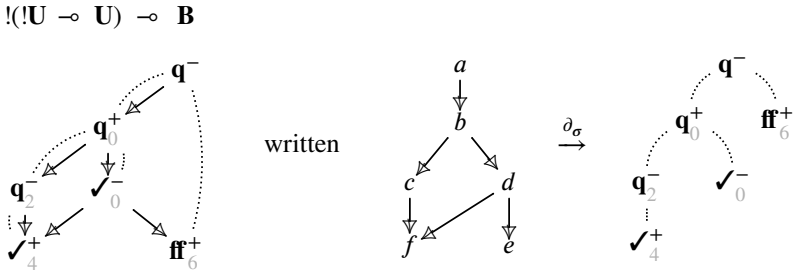


Figure 9.20: A configuration of $\sigma : !(!\mathbb{U} \multimap \mathbb{U}) \multimap \mathbf{B}$

Note that though pointifixion was defined as

$$(\overset{\curvearrowleft}{-}) : \circlearrowleft\text{-}\mathsf{Plays}(!A)/\cong_{!A} \quad \to \quad \circlearrowleft\text{-}\mathsf{PrePlays}(\underline{A}),$$

the same definition may be transparently applied to obtain, for all A,

$$(\overset{\curvearrowleft}{-}) : \circlearrowleft\text{-}\mathsf{Plays}(A)/\cong_{A} \quad \to \quad \circlearrowleft\text{-}\mathsf{PrePlays}_{\bullet}(\underline{A}),$$

used without further details. Additionally, extensions of plays lift along these functions:

Lemma 9.2.10. *Consider A a mixed board. We have the following properties:*

(1) *For all $ta \in \circlearrowleft\text{-}\mathsf{PrePlays}_{\bullet}(\underline{A})$, $s \in \circlearrowleft\text{-}\mathsf{Plays}(A)$ such that $t = \overleftarrow{s}$, there is $sa' \in \circlearrowleft\text{-}\mathsf{Plays}(A)$ such that $(\overset{\curvearrowleft}{-})(sa') = ta$;*
(2) *For all $ta \in \circlearrowleft\text{-}\mathsf{PrePlays}(\underline{A})$, $s \in \circlearrowleft\text{-}\mathsf{Plays}(!A)$ such that $t = \overleftarrow{s}$, there is $sa' \in \circlearrowleft\text{-}\mathsf{Plays}(!A)$ such that $(\overset{\curvearrowleft}{-})(sa') = ta$.*

Sketch. The proof exploits that since we have interpreted ground types duplicating answers, for any extension of $t \in \circlearrowleft\text{-}\mathsf{PrePlays}(\underline{A})$ one can find a "fresh" move in $!A$, with a new copy index, matching the extension. However, formalizing this requires the full definition of mixed boards which we prefer to postpone to Section 12.1.1. □

Proposition 9.2.9 and Lemma 9.2.10 entail that plays with pointers on $\underline{A}$ may be regarded *exactly* as non-alternating plays on the $-$-board $!A$ considered up to symmetry:

Corollary 9.2.11. *Consider A a mixed board. Then, $(\overset{\curvearrowleft}{-})$ yields a bijection*

$$(\overset{\curvearrowleft}{-}) \quad : \quad \circlearrowleft\text{-}\mathsf{Plays}(!A)/\cong_{!A} \quad \simeq \quad \circlearrowleft\text{-}\mathsf{PrePlays}(\underline{A}),$$

preserving length and prefix.

The above development concerns *non-alternating pre-plays with pointers* (Definition 5.1.1); recall that the *non-alternating plays with pointers* from which non-alternating game semantics are built are also (logically) well-bracketed (Definition 5.1.3).

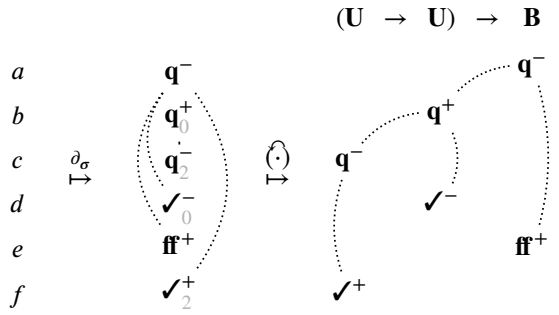


Figure 9.21: Constructing a play in the unfolding

9.2.3 Unfoldings of a Causal Strategy

Next, we describe how to *unfold* a causal strategy to a non-alternating strategy.

More precisely, we shall provide *three* different unfoldings:

(1) $\circlearrowleft$-$\mathsf{Unf}_\bullet$ unfolds $\sigma : A$ to a non-alternating thread-strategy on $\underline{A}$,
(2) $\circlearrowleft$-$\mathsf{Unf}_!$ unfolds $\sigma : !A$ to a non-alternating strategy on $\underline{A}$,
(3) $\circlearrowleft$-Unf unfolds $\sigma : !A \vdash B$ to a non-alternating strategy on $\underline{A} \Rightarrow \underline{B}$,

for all mixed boards A and B.

Unfolding to a thread-strategy. For A a mixed board, we may now *unfold* a causal strategy $\sigma : A$ to a set $\circlearrowleft$-$\mathsf{Unf}_\bullet(\sigma)$ of threads. Plays in $\circlearrowleft$-$\mathsf{Unf}_\bullet(\sigma)$ are obtained from σ in three steps. First *(1)* form the *linearizations of configurations* of σ, *i.e.* $\mathscr{L}(\sigma)$ (Definition 9.2.1). Then *(2)*, project those to $\circlearrowleft$-$\mathsf{Plays}(A)$ by applying ∂_σ event by event. Then finally, *(3)* apply pointifixion. More formally, the definition reads as:

Definition 9.2.12. *Consider A a mixed board, and $\sigma : A$ a causal strategy.*
*The **unfolding** of σ is the set* $\circlearrowleft$-$\mathsf{Unf}_\bullet(\sigma) = \{\overset{\circlearrowleft}{(\cdot)}(s^\sigma_A) \in \circlearrowleft\text{-}\mathsf{Plays}(\underline{A}) \mid s^\sigma \in \mathscr{L}(\sigma)\}$.

Observe that not all linearizations of σ contribute to the unfolding, only those which by display and pointifixion yield a (logically) well-bracketed play.

We illustrate this in Figures 9.20 and 9.21. Figure 9.20 displays a configuration of a causal σ on $!(!\mathbf{U} \multimap \mathbf{U}) \multimap \mathbf{B}$, first with our usual notational convention, then explicitely separating the internal events of σ from their display in the game. In Figure 9.21, we start with the sequence $abcdef \in \mathscr{L}(\sigma)$; then follow the two steps of the construction of the corresponding play of the unfolding: first by displaying it on A, then pointifixion. In Figure 9.21, we insist that though pointers appear in the intermediary step, they are not part of the structure of plays and only represent immediate causal dependency in the game. On the right hand side, pointers are now explicitly part of the play.

As expected, this unfolding yields a valid non-alternating thread-strategy on $\underline{A}$:

Proposition 9.2.13. *Consider* $\boldsymbol{\sigma} : A$ *a causal strategy.*
Then, $\circlearrowleft\text{-}\mathsf{Unf}_\bullet(\boldsymbol{\sigma}) : \underline{A}$ *is a thread-strategy in the sense of Definition 5.1.7.*

Proof. Clearly it is a set of threads. *Non-empty* and *prefix-closed* are immediate.

Receptive: consider $t = (\overset{\frown}{-})(s^\sigma_A)$ for $s^\sigma \in \mathscr{L}(\boldsymbol{\sigma})$, with $tb^- \in \circlearrowleft\text{-}\widehat{\mathsf{Plays}}_\bullet(\underline{A})$. By Lemma 9.2.10, there is $s^\sigma_A a^- \in \circlearrowleft\text{-}\mathsf{Plays}(A)$ such that $(\overset{\frown}{-})(s^\sigma_A a^-) = tb^-$. Now, by Proposition 9.2.5, there is $s^\sigma m \in \mathscr{L}(\boldsymbol{\sigma})$ displaying to $s^\sigma_A a^-$, and *receptive* follows.

Courteous: consider $uabv = (\overset{\frown}{-})(u^\sigma_A a' b' v^\sigma_A)$ where a is positive or b negative, and assume $ubav \in \circlearrowleft\text{-}\widehat{\mathsf{Plays}}_\bullet(\underline{A})$. It follows that $u^\sigma_A b' a' v^\sigma_A \in \circlearrowleft\text{-}\mathsf{Plays}(A)$: if we had $a' \rightarrowtriangle_A b'$, this would mean that b points on a by definition of pointifixion. Hence, $u^\sigma b' a' v^\sigma_A \in \circlearrowleft\text{-}\mathsf{Plays}(\boldsymbol{\sigma})$ by Proposition 9.2.5, whose pointifixion is $ubav$ as required. □

Unfolding to a non-alternating strategy. Above, we have defined the unfolding of $\boldsymbol{\sigma} : A$ as a thread-strategy $\circlearrowleft\text{-}\mathsf{Unf}_\bullet(\boldsymbol{\sigma})$. Next we unfold $\boldsymbol{\sigma} : !A$ to a full strategy:

Definition 9.2.14. *Consider* A *a mixed board, and* $\boldsymbol{\sigma} : !A$ *a causal strategy.*
The ***unfolding*** *of* $\boldsymbol{\sigma}$ *is the set* $\circlearrowleft\text{-}\mathsf{Unf}_!(\boldsymbol{\sigma}) = \{(\overset{\frown}{\cdot})(s^\sigma_{!A}) \mid s^\sigma \in \mathscr{L}(\boldsymbol{\sigma})\}$.

With respect to Definition 9.2.12, the only difference is that we start with a causal strategy on $!A$ rather than A. The immediate consequence is that the unfolding process will give us *plays*, rather than merely threads. As before, we have:

Proposition 9.2.15. *Consider* A *a mixed board and* $\boldsymbol{\sigma} : !A$ *a causal strategy.*
Then, $\circlearrowleft\text{-}\mathsf{Unf}_!(\boldsymbol{\sigma}) : \underline{A}$ *is a non-alternating strategy as in Definition 5.1.5.*

Proof. Same proof as for Proposition 9.2.13. □

If $\boldsymbol{\sigma} : A$ is a causal strategy, we may unfold it as a thread-strategy with $\circlearrowleft\text{-}\mathsf{Unf}_\bullet(\boldsymbol{\sigma}) : \underline{A}$; or we may unfold its promotion as a full non-alternating strategy with $\circlearrowleft\text{-}\mathsf{Unf}_!(\boldsymbol{\sigma}^!) : \underline{A}$. In the sequel we shall need to relate these two unfoldings: in particular, $\circlearrowleft\text{-}\mathsf{Unf}_!(\boldsymbol{\sigma}^!)$ is a single-threaded strategy whose generating thread-strategy is $\circlearrowleft\text{-}\mathsf{Unf}_\bullet(\boldsymbol{\sigma})$:

Lemma 9.2.16. *Consider causal* $\boldsymbol{\sigma} : A$*, and* $\boldsymbol{\sigma}^! : !A$ *its promotion.*
Then, $(\circlearrowleft\text{-}\mathsf{Unf}_\bullet(\boldsymbol{\sigma}))^! = \circlearrowleft\text{-}\mathsf{Unf}_!(\boldsymbol{\sigma}^!)$.

Proof. For $\subseteq$, we show by induction on $n \in \mathbb{N}$ that for all $n \in \mathbb{N}$, $\circlearrowleft\text{-}\mathsf{Unf}_\bullet(\boldsymbol{\sigma})^{(n)}$ as defined below Definition 5.1.7, is included in $\circlearrowleft\text{-}\mathsf{Unf}_!(\boldsymbol{\sigma}^!)$. This boils down to the fact that whenever $s \in \circlearrowleft\text{-}\mathsf{Unf}_!(\boldsymbol{\sigma}^!)$ and $t \in \circlearrowleft\text{-}\mathsf{Unf}_\bullet(\boldsymbol{\sigma})$, then $s \sqcup t \subseteq \circlearrowleft\text{-}\mathsf{Unf}_!(\boldsymbol{\sigma}^!)$.

Consider $u \in s \sqcup t$. We have $s = \hat{u}$ and $t = \hat{v}$ for $u = \partial_{\boldsymbol{\sigma}^!}(u')$ and $v = \partial_{\boldsymbol{\sigma}}(v')$ with

$$u' \in \circlearrowleft\text{-}\mathsf{Plays}(\boldsymbol{\sigma}^!)\,, \qquad v' \in \circlearrowleft\text{-}\mathsf{Plays}(\boldsymbol{\sigma})\,.$$

Now, u' visits only finitely many copies for the outermost !, so considering

$$w' = (i, v'_1) \dots (i, v'_n)$$

for $v' = v'_1 \dots v'_n$, with $i \in \mathbf{N}$ not visited by u', we get $w' \in \circlearrowleft\text{-}\mathsf{Plays}(\boldsymbol{\sigma}^!)$ satisfying that writing $w = \partial_{\boldsymbol{\sigma}^!}(w')$, we have $\hat{w} = \hat{v} = t$. But necessarily, u' and w' are disjoint; so we may form $z' \in \circlearrowleft\text{-}\mathsf{Plays}(\boldsymbol{\sigma}^!)$ the interleaving of u' and w' following the same scheduling

as $u \in s \sqcup t$, and by construction $u = \hat{z}$ for $z = \partial_{\sigma^!}(z')$.

For $\supseteq$, consider $s \in \mathcal{O}\text{-}\mathsf{Unf}_!(\sigma^!)$. By definition, $s = \hat{u}$ and $u = \partial_{\sigma^!}(u')$ such that $u' \in \mathcal{O}\text{-}\mathsf{Plays}(\sigma^!)$. By definition, u' visits a finite set I of copy indices; each $i \in I$ yields a restriction $u'_i \in \mathcal{O}\text{-}\mathsf{Plays}(\sigma)$. For each $i \in I$, we write $u_i = \partial_\sigma(u'_i)$ and $s_i = \hat{u_i}$ – by construction, $s_i \in \mathcal{O}\text{-}\mathsf{Unf}_\bullet(\sigma)$. Finally, s is an interleaving of all the s_i's. □

Unfolding from mixed board A to mixed board B. We wish to relate the Kleisli category $\mathbf{NTCG}_!$ to the cartesian closed category $\mathcal{O}$-**Strat** – this induces two complications in the unfolding, departing from the simple case described above.

Firstly, the way the two hom-sets are constructed differs slightly. Indeed, a morphism σ from A to B in $\mathbf{NTCG}_!$ is a causal strategy on $!A \vdash B$, *i.e.* $(!A)^\perp \parallel B$ where $!A, B$ are causally independent. In constrast, non-alternating strategies from $\underline{A}$ to $\underline{B}$ have plays in $\underline{A} \Rightarrow \underline{B}$, where the arrow construction imposes a dependency between (the initial moves of) the two components. We must thus consider the currying of σ:

$$\Lambda(\sigma) : !A \multimap B\,.$$

Secondly, the configurations of a causal strategy $\sigma : !A \multimap B$ have at most one initial move in B – in contrast with non-alternating plays on $\underline{A} \Rightarrow \underline{B}$, in which arbitrarily many initial moves in $\underline{B}$ may occur. This is solved by unfolding not $\Lambda(\sigma)$ but

$$\Lambda(\sigma)^! : !(!A \multimap B)\,,$$

its promotion, which will be the basis of the unfolding. This prompts:

Definition 9.2.17. *Consider $\sigma : !A \vdash B$ a causal strategy. Its **unfolding** is:*

$$\mathcal{O}\text{-}\mathsf{Unf}(\sigma) = \mathcal{O}\text{-}\mathsf{Unf}_!(\Lambda(\sigma)^!) : \underline{A} \Rightarrow \underline{B}\,.$$

It is a consequence of Proposition 9.2.15 that this is well-defined.

9.3 $\multimap\!\!\triangleright$-Strat and Full Abstraction

We aim to prove that the unfolding above is as expected compatible with all operations used in the interpretation of $\mathtt{IA}_{/\!/}$, which shall bridge the two interpretations.

But we shall soon see that this will require us to add one further condition refining $\mathbf{NTCG}_!$ to $\multimap\!\!\triangleright$-**Strat**, dealing with the fact that non-alternating plays with pointers are (logically) well-bracketed while arbitrary linearizations might not be.

9.3.1 Unfolding the Categorical Structure

We examine how unfolding preserves the categorical structure of $\mathbf{NTCG}_!$.

Unfolding of copycat. First, we unfold the identities of $\mathbf{NTCG}_!$. Here, not much is happening conceptually; however technically the respective definitions of the causal and non-alternating copycat strongly differ: while the causal copycat comes from an explicit, concrete description (Definition 6.4.1), the non-alternating copycat is in a sense the "free asynchronous completion" of the alternating copycat (Definition 5.2.5). This indirect definition makes it quite unwieldy to reason with.

Proposition 9.3.1. *Consider A a mixed board.*

Then, $\mathbf{der}_A : !A \vdash A$ *unfolds to the non-alternating copycat strategy,* $\circlearrowleft\text{-}\mathfrak{cc}_{\underline{A}}$.

Proof. Recall from Definition 5.2.5 and Lemma 5.1.10 that the plays of $\circlearrowleft\text{-}\mathfrak{cc}_A$ are those obtained from $\Updownarrow\text{-}\mathfrak{cc}_A$ by *receptivity* and *courtesy* in the sense of Definition 5.1.5.

By Lemma 9.2.16, it suffices to prove the equality

$$\circlearrowleft\text{-}\mathsf{Unf}_\bullet(\Lambda(\mathbf{der}_A)) = \circlearrowleft\text{-}\mathfrak{cc}^\bullet_{\underline{A}}\,,$$

where $\circlearrowleft\text{-}\mathfrak{cc}^\bullet_{\underline{A}}$ denotes the set of *threads* of $\circlearrowleft\text{-}\mathfrak{cc}_{\underline{A}}$.

For $\subseteq$ consider $s \in \circlearrowleft\text{-}\mathsf{Unf}_\bullet(\Lambda(\mathbf{der}_A))$, meaning that there are $s = \hat{t}$ with $t = \partial_{\Lambda(\mathbf{der}_A)}(u)$ for some $u \in \mathscr{L}(\mathfrak{cc}_A)$. Now, one can show that there is $v \in \Updownarrow\text{-}\mathscr{L}(\mathfrak{cc}_A)$ and w^- a sequence of moves of $\mathfrak{cc}_A$ negative in $A \vdash A$, such that

$$vw^- \in \circlearrowleft\text{-}\mathsf{Plays}(\mathfrak{cc}_A)$$

has the same moves as u, and s.t. u may be obtained from vw^- by *courtesy*, such that (the pointifixion of) all intermediate plays are logically well-bracketed. Considering

$$t' = \partial_{\Lambda(\mathbf{der}_A)}(v)\,, \qquad s' = \hat{t'}\,, \qquad t'' = \partial_{\Lambda(\mathbf{der}_A)}(vw^-)\,, \qquad s'' = \hat{t''}\,,$$

it is immediate that $t' \in \Updownarrow\text{-}\mathfrak{cc}_{\underline{A}}$. Thus, $t' \in \circlearrowleft\text{-}\mathfrak{cc}_{\underline{A}}$, and so $t'' \in \circlearrowleft\text{-}\mathfrak{cc}_{\underline{A}}$ by receptivity. But then, by construction s may be obtained from s'' by courtesy, thus $s \in \circlearrowleft\text{-}\mathfrak{cc}_{\underline{A}}$.

For $\supseteq$, consider $s \in \circlearrowleft\text{-}\mathfrak{cc}_{\underline{A}}$ a thread. By definition, s is obtained by courtesy and receptivity from some $s' \in \Updownarrow\text{-}\mathfrak{cc}_{\underline{A}}$. Writing $s' \in \circlearrowleft\text{-}\mathsf{Pla}\hat{\mathsf{y}}\mathsf{s}(\underline{A}_1 \Rightarrow \underline{A}_2)$, for all positive-ending $t' \sqsubseteq s'$, $t' \upharpoonright \underline{A}_1 = t' \upharpoonright \underline{A}_2$. By Corollary 9.2.11, there is $u \in \Updownarrow\text{-}\mathsf{Plays}(A)$ s.t. $t' \upharpoonright \underline{A}_1 = \hat{u}$. We may then define $v \in \Updownarrow\text{-}\mathscr{L}(\mathfrak{cc}_A)$ by interleaving two copies of u in the obvious way, s.t. $v \upharpoonright A_1 = v \upharpoonright A_2 = u$. From the definition, it is direct that writing $w = \partial_{\Lambda(\mathbf{der}_A)}(v)$, $s' = \hat{w}$. By Proposition 9.2.5, applying to w instances of courtesy and receptivity as for $s' \rightsquigarrow s$, we get $z \in \circlearrowleft\text{-}\mathsf{Plays}(\mathfrak{cc}_A)$ s.t. $s = \curvearrowleft(\partial_{\Lambda(\mathbf{der}_A)}(z))$. □

The definition of $\circlearrowleft\text{-}\mathfrak{cc}_{\underline{A}}$ via saturation in Definition 5.2.5 is unpleasant: to prove that a play is in $\circlearrowleft\text{-}\mathfrak{cc}_{\underline{A}}$ we must provide a sequence of applications of courtesy and receptivity from a play of $\Updownarrow\text{-}\mathfrak{cc}_{\underline{A}}$. It seems hard to avoid while staying in the context of pointer games. In contrast, $\overline{\mathbf{der}}_A$ offers via unfolding a concrete definition of $\circlearrowleft\text{-}\mathfrak{cc}_{\underline{A}}$.

Well-bracketing. Unfolding preserves the identity; but composition is *not* preserved: for $\boldsymbol{\sigma} \in \mathbf{NTCG}_!(A, B)$ and $\boldsymbol{\tau} \in \mathbf{NTCG}_!(B, C)$, we may not have

$$\circlearrowleft\text{-}\mathsf{Unf}(\boldsymbol{\tau} \odot_! \boldsymbol{\sigma}) \subseteq \circlearrowleft\text{-}\mathsf{Unf}(\boldsymbol{\tau}) \odot \circlearrowleft\text{-}\mathsf{Unf}(\boldsymbol{\sigma})\,.$$

The issue is that unfolding only keeps (logically) well-bracketed plays, but well-bracketed plays in the unfolding of $\boldsymbol{\tau} \odot_! \boldsymbol{\sigma}$ may very well originate in pre-plays in the unfoldings of $\boldsymbol{\sigma}$ and $\boldsymbol{\tau}$ that fail well-bracketing. In order to obtain this inclusion, we shall have to put an additional restriction on causal strategies:

Definition 9.3.2. *Consider $\boldsymbol{\sigma} : !A \vdash B$ a causal strategy. It is **well-bracketed** iff for all $sm^+ \in \mathscr{L}(\boldsymbol{\sigma})$, if $\overset{\frown}{(\cdot)}(\partial_{\Lambda(\boldsymbol{\sigma})}(s))$ is (logically) well-bracketed, then so is $\overset{\frown}{(\cdot)}(\partial_{\Lambda(\boldsymbol{\sigma})}(sm))$.*

Alternatively, $\boldsymbol{\sigma}$ is well-bracketed iff the unfolding only "cuts at Opponent moves". We skip the routine but lengthy verification of the following fact:

Theorem 9.3.3. *There is $\multimap\!\!\triangleright$-**Strat**, a cartesian closed $\sim$-category with objects mixed boards and morphisms those well-bracketed $\boldsymbol{\sigma} \in \mathbf{NTCG}_!(A, B)$, which supports the interpretation of $\mathtt{IA}_{/\!/}$. Moreover there is an obvious forgetful $\sim$-functor*

$$\multimap\!\!\triangleright\text{-}\mathbf{Strat} \to \mathbf{NTCG}_!$$

sending a mixed board to the underlying $-$-board, leaving morphisms unchanged.

Preservation of composition. The models $\multimap\!\!\triangleright$-**Strat** and $\circlearrowleft$-**Strat** differ fundamentally in that one is causal while the other is based on interleavings, but that in itself is not hard to manage, via the mechanism of linearizations.

The real challenge is that the compositional mechanisms of these two models follow significantly different principles: *(1)* $\multimap\!\!\triangleright$-**Strat** is constructed via the Kleisli category of the linear model **NTCG**, while $\circlearrowleft$-**Strat** is inherently non-linear; *(2)* $\multimap\!\!\triangleright$-**Strat** uses explicit copy indices, while $\circlearrowleft$-**Strat** uses pointers; and *(3)* the hom-set of $\multimap\!\!\triangleright$-**Strat** is the board $!A \vdash B$ with no causal dependency between A and B, whereas $\circlearrowleft$-**Strat** uses the arrow construction $\underline{A} \Rightarrow \underline{B}$ (which seems unavoidable for a pointer-based model). None of these are particularly challenging, but the three are deeply intertwinned and must be managed simultaneously in the compatibility of unfolding with composition[4].

We address the two directions of the unfolding separately. First:

Lemma 9.3.4. *Consider $\boldsymbol{\sigma} \in \multimap\!\!\triangleright\text{-}\mathbf{Strat}(A, B)$ and $\boldsymbol{\tau} \in \multimap\!\!\triangleright\text{-}\mathbf{Strat}(B, C)$.*
Then, we have the inclusion $\circlearrowleft\text{-}\mathsf{Unf}_\bullet(\Lambda(\boldsymbol{\tau} \odot \boldsymbol{\sigma}^!)) \subseteq \circlearrowleft\text{-}\mathsf{Unf}(\boldsymbol{\tau}) \odot \circlearrowleft\text{-}\mathsf{Unf}(\boldsymbol{\sigma})$.

Proof. For $s \in \circlearrowleft\text{-}\mathsf{Unf}_\bullet(\Lambda(\boldsymbol{\tau} \odot \boldsymbol{\sigma}^!))$, by definition $s = \overset{\frown}{(\cdot)}(\partial_{\Lambda(\boldsymbol{\tau} \odot \boldsymbol{\sigma}^!)}(s'))$ for some $s' \in \circlearrowleft\text{-}\mathsf{Plays}(\boldsymbol{\tau} \odot \boldsymbol{\sigma}^!)$. There is $u' \in \circlearrowleft\text{-}\mathsf{Plays}(\boldsymbol{\tau} \circledast \boldsymbol{\sigma}^!)$ with visible restriction s', and

$$u'_{\boldsymbol{\sigma}^!} \in \circlearrowleft\text{-}\mathsf{Plays}(\boldsymbol{\sigma}^!) = \circlearrowleft\text{-}\mathsf{Plays}(!\boldsymbol{\sigma}), \qquad u'_{\boldsymbol{\tau}} \in \circlearrowleft\text{-}\mathsf{Plays}(\boldsymbol{\tau}).$$

are projections obtained applying move-by-move the (partial) projections of moves of the interaction to moves of the compound strategies as defined just above Lemma 6.2.14.

Now, u' is displayed to $v' \in \circlearrowleft\text{-}\mathsf{Plays}(!A \parallel !B \parallel C)$ via $\partial_{\boldsymbol{\tau} \circledast \boldsymbol{\sigma}}$. There is an implicit dependency between moves minimal in the game: moves minimal in $!B$ depend on the minimal move in C. Likewise, there is a unique way to assign moves minimal in $!A$ to

[4] Ideally, one would deal with them separately by introducing a separate model: a non-alternating play-based games model based on copy indices rather than pointers – but we leave this out of scope.

those minimal in $!B$ in a way compatible with $\partial_{!\Lambda(\boldsymbol{\sigma})}(u'_{\boldsymbol{\sigma}^!}) \in \mathcal{O}\text{-}\mathsf{Plays}(!(!A \multimap B))$. We form $w' \in \mathcal{O}\text{-}\mathtt{I}(\underline{A}, \underline{B}, \underline{C})$ by first applying labels to v' as in Proposition 9.2.9, extended with this assignment. Now from the constructions, direct verifications entail

$$w' \upharpoonright \underline{A}, \underline{B} \in \mathcal{O}\text{-}\mathsf{Unf}_!(\Lambda(\boldsymbol{\sigma})^!)\,, \quad w' \upharpoonright \underline{B}, \underline{C} \in \mathcal{O}\text{-}\mathsf{Unf}_!(\Lambda(\boldsymbol{\tau})^!)\,, \quad w' \upharpoonright \underline{A}, \underline{C} = s\,,$$

thus w' is a witness for $s \in \mathcal{O}\text{-}\mathsf{Unf}(\boldsymbol{\tau}) \odot \mathcal{O}\text{-}\mathsf{Unf}(\boldsymbol{\sigma})$ as required. □

These "direct verifications" include, in particular, that the restrictions of w' are (logically) well-bracketed. That w' is itself well-bracketed is established by immediate induction, using that $\boldsymbol{\sigma}$ and $\boldsymbol{\tau}$ are well-bracketed. Now, we focus on the other inclusion:

Lemma 9.3.5. *Consider $\boldsymbol{\sigma} \in \mathbf{NTCG}_!(A, B)$, $\boldsymbol{\tau} \in \mathbf{NTCG}_!(B, C)$.*
Then, threads in $\mathcal{O}\text{-}\mathsf{Unf}(\boldsymbol{\tau}) \odot \mathcal{O}\text{-}\mathsf{Unf}(\boldsymbol{\sigma})$ are in $\mathcal{O}\text{-}\mathsf{Unf}_\bullet(\Lambda(\boldsymbol{\tau} \odot \boldsymbol{\sigma}^!))$.

Proof. Consider $s \in \mathcal{O}\text{-}\mathsf{Unf}(\boldsymbol{\tau}) \odot \mathcal{O}\text{-}\mathsf{Unf}(\boldsymbol{\sigma})$ a thread, so there is $u \in \mathcal{O}\text{-}\mathtt{I}(\underline{A}, \underline{B}, \underline{C})$ s.t. $u \upharpoonright \underline{A}, \underline{B} \in \mathcal{O}\text{-}\mathsf{Unf}(\boldsymbol{\sigma})$, $u \upharpoonright \underline{B}, \underline{C} \in \mathcal{O}\text{-}\mathsf{Unf}(\boldsymbol{\tau})$, and $u \upharpoonright \underline{A}, \underline{C} = s$. By hypothesis,

$$u \upharpoonright \underline{A}, \underline{B} = \overset{\frown}{(\cdot)}(\partial_{\Lambda(\boldsymbol{\sigma})^!}(v^{\boldsymbol{\sigma}}))\,, \qquad u \upharpoonright \underline{B}, \underline{C} = \overset{\frown}{(\cdot)}(\partial_{\Lambda(\boldsymbol{\tau})^!}(v^{\boldsymbol{\tau}}))$$

for $v^{\boldsymbol{\sigma}} \in \mathcal{O}\text{-}\mathsf{Plays}(!\boldsymbol{\sigma}) = \mathcal{O}\text{-}\mathsf{Plays}(\boldsymbol{\sigma}^!)$ and $v^{\boldsymbol{\tau}} \in \mathcal{O}\text{-}\mathsf{Plays}(!\boldsymbol{\tau})$; but as $v^{\boldsymbol{\sigma}}$ has one initial move, we consider $v^{\boldsymbol{\tau}} \in \mathcal{O}\text{-}\mathsf{Plays}(\boldsymbol{\tau})$ with $u \upharpoonright \underline{B}, \underline{C} = \overset{\frown}{(\cdot)}(\partial_{\Lambda(\boldsymbol{\tau})}(v^{\boldsymbol{\tau}}))$. We also have:

$$w^{\boldsymbol{\sigma}} = \partial_{\boldsymbol{\sigma}^!}(v^{\boldsymbol{\sigma}}) \in \mathcal{O}\text{-}\mathsf{Plays}(!A \vdash !B)\,, \qquad w^{\boldsymbol{\tau}} = \partial_{\boldsymbol{\tau}}(v^{\boldsymbol{\tau}}) \in \mathcal{O}\text{-}\mathsf{Plays}(!B \vdash C)\,,$$

and since $u \upharpoonright \underline{A}, \underline{B}$ and $u \upharpoonright \underline{B}, \underline{C}$ coincide on $\underline{B}$, it is a direct verification that

$$\overset{\frown}{(\cdot)}(w^{\boldsymbol{\sigma}} \upharpoonright !B) \quad = \quad u \upharpoonright \underline{B} \quad = \quad \overset{\frown}{(\cdot)}(w^{\boldsymbol{\tau}} \upharpoonright !B)\,,$$

so that $w^{\boldsymbol{\sigma}} \upharpoonright !B \cong_{!B} w^{\boldsymbol{\tau}} \upharpoonright !B$ by Proposition 9.2.9. Now, writing configurations

$$x^{\boldsymbol{\sigma}} = |v^{\boldsymbol{\sigma}}|\,, \qquad x^{\boldsymbol{\tau}} = |v^{\boldsymbol{\tau}}|\,,$$

this informs a symmetry $\theta : x^{\boldsymbol{\sigma}}_{!B} \cong_{!B} x^{\boldsymbol{\tau}}_{!B}$. The difficulty is that this symmetry might not be the identity, preventing us from synchronizing $x^{\boldsymbol{\sigma}}$ and $x^{\boldsymbol{\tau}}$.

So additionally, we note that u informs a linearization of the composite bijection

$$\varphi[x^{\boldsymbol{\sigma}}, \theta, x^{\boldsymbol{\tau}}] : x^{\boldsymbol{\sigma}} \parallel x^{\boldsymbol{\tau}}_C \overset{\partial_{\boldsymbol{\sigma}} \parallel x^{\boldsymbol{\tau}}_C}{\simeq} x^{\boldsymbol{\sigma}}_{!A} \parallel x^{\boldsymbol{\sigma}}_{!B} \parallel x^{\boldsymbol{\tau}}_C \overset{x^{\boldsymbol{\sigma}}_{!A} \parallel \theta \parallel x^{\boldsymbol{\tau}}_C}{\simeq} x^{\boldsymbol{\sigma}}_{!A} \parallel x^{\boldsymbol{\tau}}_{!B} \parallel x^{\boldsymbol{\tau}}_C \overset{\partial^{-1}_{\boldsymbol{\tau}} \parallel x^{\boldsymbol{\tau}}_C}{\simeq} x^{\boldsymbol{\sigma}}_{!A} \parallel x^{\boldsymbol{\tau}}\,,$$

witnessing the fact that the triple $(x^{\boldsymbol{\sigma}}, \theta, x^{\boldsymbol{\tau}})$ is causally compatible in the sense of Definition 7.4.3. We write $r \in \mathcal{O}\text{-}\mathsf{Plays}(\varphi[x^{\boldsymbol{\sigma}}, \theta, x^{\boldsymbol{\tau}}])$ for this linearization; it projects to $\partial(r) \in \mathcal{O}\text{-}\mathsf{Plays}(!A \parallel !B \parallel C)$ in the obvious way, and we have

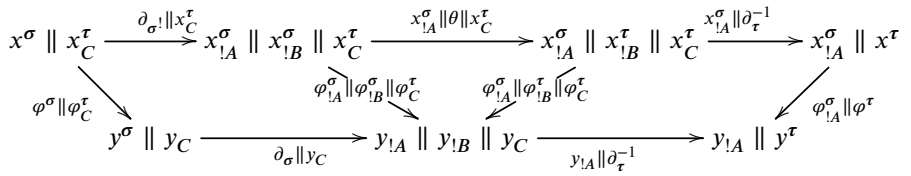
$$\overset{\frown}{(\cdot)}(\partial(r)) = \bar{u} \in \mathcal{O}\text{-}\mathsf{Plays}(\underline{A} \times \underline{B} \times \underline{C})$$

where $\bar{u}$ is u without the pointers of initial moves in $\underline{A}$ and $\underline{B}$.

Thanks to causal compatibility, Proposition 7.4.4 provides us with $y^{\boldsymbol{\sigma}} \in \mathscr{C}(\boldsymbol{\sigma}^!)$ and $y^{\boldsymbol{\tau}} \in \mathscr{C}(\boldsymbol{\tau})$ causally compatible configurations, along with symmetries

$$\varphi^{\boldsymbol{\sigma}} : x^{\boldsymbol{\sigma}} \cong_{\boldsymbol{\sigma}^!} y^{\boldsymbol{\sigma}}\,, \qquad \varphi^{\boldsymbol{\tau}} : x^{\boldsymbol{\tau}} \cong_{\boldsymbol{\tau}} y^{\boldsymbol{\tau}}\,,$$

such that we have $\varphi^{\sigma}_{!A} \in \mathcal{S}_{-}(!A)$, $\varphi^{\tau}_{C} \in \mathcal{S}_{+}(C)$, and the following diagram commutes



following which the linearization $r \in \mathcal{O}\text{-}\mathsf{Plays}(\varphi[x^{\sigma}, \theta, x^{\tau}])$ transports to a linearization $r' \in \mathcal{O}\text{-}\mathsf{Plays}(\varphi[y^{\sigma}, y^{\tau}])$ satisfying $\partial(r) \cong_{!A \| !B \| C} \partial(r')$, so that $\overset{\frown}{(\cdot)}(\partial(r')) = \bar{u}$.

Now by Proposition 6.2.7, $\varphi[y^{\tau} \circledast y^{\sigma}]$ is order-isomorphic to $y^{\tau} \circledast y^{\sigma} \in \mathscr{C}(\tau \circledast \sigma^{!})$. Accordingly, r' transports to $r'' \in \mathcal{O}\text{-}\mathsf{Plays}(y^{\tau} \circledast y^{\sigma})$ such that $\overset{\frown}{(\cdot)}(\partial_{\tau \circledast \sigma^{!}}(r'')) = \bar{u}$, which projects to $z \in \mathcal{O}\text{-}\mathsf{Plays}(\tau \odot \sigma^{!})$ such that $\overset{\frown}{(\cdot)}(\partial_{\tau \odot \sigma^{!}}(z)) = \bar{s}$. But s is a thread so that the pointers of initial moves in A are forced; accordingly $\overset{\frown}{(\cdot)}(\partial_{\Lambda(\tau \odot \sigma^{!})}(z)) = s$. □

Note that this inclusion does not require well-bracketing. Altogether:

Corollary 9.3.6. *Consider* $\sigma \in \multimap\!\!\triangleright\text{-}\mathbf{Strat}(A, B)$ *and* $\tau \in \multimap\!\!\triangleright\text{-}\mathbf{Strat}(B, C)$.
Then, $\mathcal{O}\text{-}\mathsf{Unf}(\tau \odot_{!} \sigma) = \mathcal{O}\text{-}\mathsf{Unf}(\tau) \odot \mathcal{O}\text{-}\mathsf{Unf}(\sigma)$.

Proof. For the first inclusion, we compute:

$$\begin{aligned}
\mathcal{O}\text{-}\mathsf{Unf}(\tau \odot_{!} \sigma) &= \mathcal{O}\text{-}\mathsf{Unf}(\tau \odot \sigma^{!}) \\
&= \mathcal{O}\text{-}\mathsf{Unf}_{!}(\Lambda(\tau \odot \sigma^{!})^{!})^{!}) \\
&= \mathcal{O}\text{-}\mathsf{Unf}_{\bullet}(\Lambda(\tau \odot \sigma^{!}))^{!} \\
&\subseteq (\mathcal{O}\text{-}\mathsf{Unf}(\tau) \odot \mathcal{O}\text{-}\mathsf{Unf}(\sigma))^{!} \\
&= \mathcal{O}\text{-}\mathsf{Unf}(\tau) \odot \mathcal{O}\text{-}\mathsf{Unf}(\sigma)
\end{aligned}$$

using definition of Kleisli composition, definition of $\mathcal{O}\text{-}\mathsf{Unf}$, Lemma 9.2.16, Lemma 9.3.4. Finally, we use that by Lemma 9.2.16, $\mathcal{O}\text{-}\mathsf{Unf}(\sigma)$ and $\mathcal{O}\text{-}\mathsf{Unf}(\tau)$ are single-threaded strategies so that their composition is, so it is stable under interleavings.

The other inclusion holds on threads by Lemma 9.3.5 and extends to all plays by Lemma 9.2.16 since the non-alternating strategies involved are single-threaded. □

Now, we may finally state:

Theorem 9.3.7. *Unfolding yields a* $\sim$*-functor* $\mathcal{O}\text{-}\mathsf{Unf} : \multimap\!\!\triangleright\text{-}\mathbf{Strat} \to \mathcal{O}\text{-}\mathbf{Strat}$.

Proof. For any $\sigma \in \multimap\!\!\triangleright\text{-}\mathbf{Strat}(A, B)$, $\mathcal{O}\text{-}\mathsf{Unf}(\sigma)$ is a non-alternating strategy on $\underline{A} \Rightarrow \underline{B}$, single-threaded by Lemma 9.2.16, thus $\mathcal{O}\text{-}\mathsf{Unf}(\sigma) \in \mathcal{O}\text{-}\mathbf{Strat}(A, B)$. The only thing remaining to check is that $\mathcal{O}\text{-}\mathsf{Unf}$ preserves equivalence, *i.e.*

$$\mathcal{O}\text{-}\mathsf{Unf}(\sigma) = \mathcal{O}\text{-}\mathsf{Unf}(\tau)$$

for all positive iso $\varphi : \sigma \approx \tau \in \multimap\!\!\triangleright\text{-}\mathbf{Strat}(A, B)$. To prove that, consider $s \in \mathcal{O}\text{-}\mathsf{Unf}(\sigma)$, *i.e.* there must be $s^{\sigma} \in \mathscr{L}(\sigma)$ such that $\overset{\frown}{(\cdot)}(\partial_{\Lambda(\sigma)^{!}}(s^{\sigma})) = s$. But then

$$\varphi(s^{\sigma}) \in \mathscr{L}(\tau)$$

and from the commutation in the game up to symmetry, it follows immediately that

$$\partial_{\Lambda(\sigma)^!}(s^\sigma) \quad \cong_{!(!A \multimap B)} \quad \partial_{\Lambda(\tau)^!}(\varphi(s^\sigma))$$

which entails that their pointifixion is the same by Proposition 9.2.9. Thus we have found $\varphi(s^\sigma) \in \mathscr{L}(\tau)$ such that $\hat{(\cdot)}(\partial_{\Lambda(\tau)^!}(\varphi(s^\sigma))) = s$, so that $s \in \mathcal{O}\text{-}\mathsf{Unf}(\tau)$. □

9.3.2 Unfolding the Interpretation of $\mathsf{IA}_{/\!/}$

Next, we show that $\mathcal{O}$-Unf preserves the interpretation of $\mathsf{IA}_{/\!/}$.

Cartesian closed $\sim$-functor. Recall that on objects, $\mathcal{O}\text{-}\mathsf{Unf}(A, \underline{A}, \mathsf{lbl}_A) = \underline{A}$ is a simple forgetful operation. By definition of constructions on mixed boards it follows that $\mathcal{O}\text{-}\mathsf{Unf}(-)$ preserves on the nose the interpretation of ground types, products and arrows, and the terminal object. In order to preserve the cartesian closed structure, from the universal property it suffices to preserve projections and the evaluation morphism. For projections, a direct variation of Proposition 9.3.1 shows that

$$\mathcal{O}\text{-}\mathsf{Unf}(\pi^!_A) = \pi_{\underline{A}}\,, \qquad \mathcal{O}\text{-}\mathsf{Unf}(\pi^!_B) = \pi_{\underline{B}}\,,$$

so that the cartesian structure is preserved in the strict sense. For the evaluation,

$$\mathcal{O}\text{-}\mathsf{Unf}(\Lambda^!(\sigma)) = \Lambda(\mathcal{O}\text{-}\mathsf{Unf}(\sigma))$$

follows from a direct verification for $\sigma : !(\Gamma \& A) \vdash B$ with Γ, A and B mixed boards: currying is defined on both sides with essentially the same relabeling. It follows that $\mathcal{O}$-Unf preserves Λ^{-1} as well, and thus the evaluation (which is $\Lambda^{-1}(\mathrm{id}_{A \Rightarrow B})$).

Overall, we get:

Theorem 9.3.8. *We have a cartesian closed $\sim$-functor:*

$$\mathcal{O}\text{-}\mathsf{Unf}(-) : \multimap\text{-}\mathbf{Strat} \to \mathcal{O}\text{-}\mathbf{Strat}$$

It is also straightforward that unfolding is continuous with respect to the dcpo structure of hom-sets, so that $\mathcal{O}\text{-}\mathsf{Unf}(\mathcal{Y}_A) = \mathcal{Y}_{\underline{A}}$ for any mixed board A.

Unfolding $\mathsf{PIA}_{/\!/}$ primitives. For all the basic primitives of $\mathsf{PIA}_{/\!/}$ (constants, conditional, sequential composition, operation, sequential and parallel let bindings, reference and semaphore queries, reference and semaphore creation, and bad variable and semaphores), it follows by inspection that they are preserved by unfolding.

Altogether, we get:

Theorem 9.3.9. *Consider $\Gamma \vdash M : A$ any term of $\mathsf{IA}_{/\!/}$. Then,*

$$\mathcal{O}\text{-}\mathsf{Unf}([\![M]\!]_{\multimap\text{-}\mathbf{Strat}}) = [\![M]\!]_{\mathcal{O}\text{-}\mathbf{Strat}}\,.$$

Along with intentional full abstraction of $\multimap$-**Strat**:

Theorem 9.3.10. *The interpretation of* $\mathtt{IA}_{/\!/}$ *in* $\multimap\!\triangleright$-**Strat** *is intensionally fully abstract.*

Proof. Consider $\vdash M, N : A$ observationally equivalent and assume $[\![M]\!]_{\multimap\triangleright\text{-}\mathbf{Strat}}$ and $[\![N]\!]_{\multimap\triangleright\text{-}\mathbf{Strat}}$ are not, *i.e.* there is $\alpha \in \multimap\!\triangleright$-$\mathbf{Strat}([\![A]\!], \mathbf{U})$ which distinguishes them. Then,

$$\circlearrowleft\text{-}\mathsf{Unf}(\alpha) \in \circlearrowleft\text{-}\mathbf{Strat}([\![A]\!], \mathbf{U})$$

which, by unfolding, distinguishes $[\![M]\!]_{\circlearrowleft\text{-}\mathbf{Strat}}$ and $[\![N]\!]_{\circlearrowleft\text{-}\mathbf{Strat}}$. But since $\circlearrowleft$-**Strat** is intensionally fully abstract for $\mathtt{IA}_{/\!/}$, this contradicts the obs. equivalence $M \simeq N$. □

In fact, from Theorems 5.3.2 and 9.3.9 we may immediately deduce that

$$M \simeq N \quad \Leftrightarrow \quad \mathsf{comp}(\circlearrowleft\text{-}\mathsf{Unf}([\![M]\!]_{\multimap\triangleright\text{-}\mathbf{Strat}})) = \mathsf{comp}(\circlearrowleft\text{-}\mathsf{Unf}([\![N]\!]_{\multimap\triangleright\text{-}\mathbf{Strat}}))\,.$$

9.4 History and Related Work

This interpretation of higher-order concurrency into concurrent games (and its non-alternating unfolding) was first introduced in [Castellan and Clairambault, 2016], although for an affine language. The interpretation of non-affine $\mathtt{IA}_{/\!/}$ in full concurrent games was first skecthed in [Castellan et al., 2019], though without any adequacy or full abstraction result. The overall results presented here, with the unfolding to $\circlearrowleft$-**Strat**, were first presented in [Castellan and Clairambault, 2021].

Overall, this forms the first "truly concurrent" model of a higher-order concurrent language with shared memory. Around the same time this was developed, a truly concurrent model of the asychronous π-calculus was proposed [Sakayori and Tsukada, 2017].

Chapter 10

Parallel Innocence

As we have seen, $\multimap\!\triangleright$-**Strat** (and already **NTCG**) is a wide semantic realm, with strategies witnessing a wealth of computational phenomena: they reflect higher-order concurrent programs, accessing shared state or semaphores. In this permissive universe, can we capture which causal patterns are definable via programs without state, *i.e.* with only pure parallel higher-order programming? Answering that is the purpose of this chapter, under the form of a condition on strategies called *parallel innocence*.

The outline is as follows. First, in Section 10.1 we introduce the definition of parallel innocence, motivating the different ingredients in sequence – in particular we introduce *visibility*, a first approximation of parallel innocence that already enjoys a number of its properties. Sections 10.2 and 10.3 are devoted to the stability under composition of parallel innocence: first visibility, and then innocence. In Section 10.4 we show that visible strategies admit an interpretation-preserving collapse to the relational model. Finally, in Section 10.5 we refine parallel innocence into *globularity*, that further constrains the causal shape of strategies accounting for Questions and Answers. This is less fundamental, but will be helpful later on for proving finite definability.

Remark. In this chapter, the interpretation is used mostly to provide examples. For succinctness, we shall refer in examples to the interpretation using *small* ground boards rather than *large* ground boards (see the beginning of Section 9.1.1). All formal statements on the interpretation hold for both, based on either small or large ground boards.

Besides, though we eventually must consider parallel innocence on the cartesian closed $\sim$-category $\multimap\!\triangleright$-**Strat**, this development is completely independent from mixed boards and (logical) well-bracketing – thus, we prefer to develop the theory of parallel innocence for the linear structure, *i.e.* based on the relative Seely $\sim$-category **NTCG**. Of course, all the constructions given here will seamlessly apply to $\multimap\!\triangleright$-**Strat** as well.

10.1 Defining Parallel Innocence

We now embark on our quest to understand the causal shape of pure parallel programs.

10.1.1 Causal Determinism

In $\mathsf{PCF}_{/\!/}$, the *parallel let* lets us write programs like

$$\Gamma \vdash \textbf{let} \begin{pmatrix} x & = & f\,\textbf{skip} \\ y & = & g\,\textbf{skip} \end{pmatrix} \textbf{in}\ x + y : \mathbb{Y}$$

where the execution of f **skip** and g **skip** are in parallel. The *execution* of this program is non-deterministic: the scheduler must run one of f **skip** and g **skip** first. Nevertheless, one may argue that in $\mathsf{PCF}_{/\!/}$, this order does not matter. Intuitively, we have

$$f\,\textbf{skip} \cdot g\,\textbf{skip} \quad \sim \quad g\,\textbf{skip} \cdot f\,\textbf{skip}$$

if f **skip** and g **skip** perform no side effect, in the sense that the environment has no way to distinguish between the two: whatever the order, the result of the computation will be the same. We say that this program is *causally deterministic*. This is unlike the behaviour of a true non-deterministic primitive such as **choice**.

Of course, $\mathsf{IA}_{/\!/}$ *is* a truly non-deterministic language (we have seen in Section 2.2.3 how to define **choice** via a race in the memory). However, this true non-determinism emerges from the combination of shared memory and parallelism. Hence, if *parallel innocence* is to remove shared memory, it must also reinstate causal determinism[1].

Definition. While *causal determinism* may be hard to express in traditional, play-based game semantics, it is straightforward to do so for causal strategies: it simply amounts to asking that Player does not impose any minimal conflicts.

Definition 10.1.1. *Consider $\sigma : A \vdash B$ a causal strategy.*
*We say σ is **causally deterministic** if for all $s \sim\!\!\sim_\sigma s'$ in σ,* $\mathrm{pol}(s) = \mathrm{pol}(s') = -$.

This rejects a truly non-deterministic strategy such as in Figure 10.1, but accepts strategies such as that in Figure 10.2 where the conflict originates in Opponent moves. Keep in mind that by Lemma 6.1.17, such an immediate conflict between negative events in the strategy *must* correspond to an immediate conflict in the game.

Copycat strategies. We show that copycat strategies are deterministic. First:

Proposition 10.1.2. *For any $-$-board A, $\mathfrak{cc}_A \in \mathbf{NTCG}(A, A)$ is causally deterministic.*

Proof. Consider $m_1 \sim\!\!\sim_{\mathfrak{cc}_A} m_2$ in minimal conflict. In particular, $m_1 \#_{\mathfrak{cc}_A} m_2$ meaning that by definition, there is $m_1' \leq_{\mathfrak{cc}_A} m_1$ and $m_2' \leq_{\mathfrak{cc}_A} m_2$ such that

$$m_1' \#_{A \vdash A} m_2'$$

[1] The status of determinism in concurrent games is unsatisfactory – see Section 14.2.2 for a discussion.

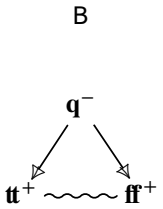


Figure 10.1: A non-deterministic strategy

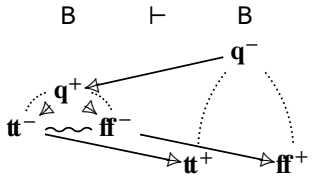


Figure 10.2: Deterministic copycat

hence there are $m_1'' \leq_{A\vdash A} m_1'$ and $m_2'' \leq_{A\vdash A} m_2'$ such that $m_1'' \sim\!\!\sim_{A\vdash A} m_2''$. Now, by Lemma 6.1.8, $m_1'' \leq_{\mathfrak{C}_A} m_1'$ and $m_2'' \leq_{\mathfrak{C}_A} m_2'$, so that $m_1'' \leq_{\mathfrak{C}_A} m_1$ and $m_2'' \leq_{\mathfrak{C}_A} m_2$; and clearly $m_1'' \#_{\mathfrak{C}_A} m_2''$ as they conflict in the game. Thus by minimality, $m_1'' = m_1$ and $m_2'' = m_2$; hence we have shown that for any $m_1 \sim\!\!\sim_{\mathfrak{C}_A} m_2$, $m_1 \sim\!\!\sim_{A\vdash A} m_2$ as well.

Now assume, seeking a contradiction, that $\mathrm{pol}_{A\vdash A}(m_1) = +$. By condition *race-free* of Definition 8.2.3, it follows that $\mathrm{pol}_{A\vdash A}(m_2) = +$ as well. But then there are unique

$$m_1' \rightarrowtriangle_{\mathfrak{C}_A} m_1\,, \qquad m_2' \rightarrowtriangle_{\mathfrak{C}_A} m_2$$

both negative and corresponding to the same event in A. So, $m_1' \sim\!\!\sim_{A\vdash A} m_2'$, contradicting the minimality of $m_1 \sim\!\!\sim_{\mathfrak{C}_A} m_2$. Hence, $\mathrm{pol}_{A\vdash A}(m_1) = \mathrm{pol}_{A\vdash A}(m_2) = -$. □

For the first time, we have used the condition *race-free* from the definition of boards. Without this condition, a minimal conflict in the game of mixed polarity, as in

$$\oplus \quad \sim\!\!\sim \quad \ominus\,,$$

causes copycat to be non-deterministic (as the reader may check).

It is straightforward that this applies to all structural morphisms involved in the categorical structure of **NTCG**, obtained via lifting of renamings – as the renaming operation $g \cdot \boldsymbol{\sigma} \cdot f$ from Section 8.1.2 does not change the strategy, only its display map.

Composition. Next, we show preservation under composition. First, we must be able to attribute the responsibility of a minimal conflict to one of the two players.

Lemma 10.1.3. *Consider $\boldsymbol{\sigma} : A \vdash B$ and $\boldsymbol{\tau} : B \vdash C$ causal strategies.*
Then, for each $p \sim\!\!\sim_{\boldsymbol{\tau}\circledast\boldsymbol{\sigma}} p'$, one of the two propositions hold:

(1) *$p_{\boldsymbol{\sigma}}, p'_{\boldsymbol{\sigma}}$ are defined and $p_{\boldsymbol{\sigma}} \sim\!\!\sim_{\boldsymbol{\sigma}} p'_{\boldsymbol{\sigma}}$,*
(2) *$p_{\boldsymbol{\tau}}, p'_{\boldsymbol{\tau}}$ are defined and $p_{\boldsymbol{\tau}} \sim\!\!\sim_{\boldsymbol{\tau}} p'_{\boldsymbol{\tau}}$.*

Proof. An equivalent statement of the hypothesis is that we have

$$x^{\boldsymbol{\tau}} \circledast x^{\boldsymbol{\sigma}} \overset{p}{\multimap} y^{\boldsymbol{\tau}} \circledast y^{\boldsymbol{\sigma}}\,, \qquad x^{\boldsymbol{\tau}} \circledast x^{\boldsymbol{\sigma}} \overset{p'}{\multimap} z^{\boldsymbol{\tau}} \circledast z^{\boldsymbol{\sigma}}$$

with $y^\tau \circledast y^\sigma, z^\tau \circledast z^\sigma$ incompatible. By Lemma 6.2.12, this means that we have

$$\varphi[x^\sigma, x^\tau] \overset{(m_1,m_2)}{\rightarrowtail} \varphi[y^\sigma, y^\tau], \qquad \varphi[x^\sigma, x^\tau] \overset{(m_1',m_2')}{\rightarrowtail} \varphi[z^\sigma, z^\tau]$$

with $\varphi[y^\sigma, y^\tau] \cup \varphi[z^\sigma, z^\tau]$ not a secured bijection. This may be because it is not a bijection – say $m_1 = m_1'$ while $m_2 \neq m_2'$. But then $m_2 = (2,t)$ and $m_2' = (2,t')$ with $\partial_\tau(t) = \partial_\tau(t')$ and thus $t \#_\tau t'$ by local injectivity. Overall, $p_\tau = t$ and $p'_\tau = t'$ are defined and conflicting. Or, this may be because $m_1 \#_{\sigma \| C} m_1'$ or $m_2 \#_{A \| \tau} m_2'$, say the former. If $m_1 = (1,s)$ and $m_1' = (1,s')$, then $p_\sigma = s$ and $p'_\sigma = s'$ are defined and conflicting. Otherwise, $m_1 = (2,c)$ and $m_2 = (2,c')$ with $c \#_C c'$. But then, necessarily $m_2 = (2,t)$ and $m_2' = (2,t')$ with $t \#_\tau t'$, thus $p_\tau = t$ and $p'_\tau = t'$ defined and conflicting.

It remains to prove that the resulting conflict is minimal, but that is obvious: if $p_\sigma \#_\sigma p'_\sigma$, all their strict dependency lie in x^σ – and symmetrically if $p_\tau \#_\tau p'_\tau$. □

It remains to link minimal conflicts in the composition with the interaction:

Lemma 10.1.4. *Consider $\sigma : A \vdash B$ and $\tau : B \vdash C$ causal strategies.*
If $p \sim\!\!\sim_{\tau \odot \sigma} p'$, then we have a diagram in $\tau \circledast \sigma$:

$$\begin{array}{ccccccccc} q_1 & \longrightarrow & q_2 & \longrightarrow & \cdots & \longrightarrow & q_m & \longrightarrow & p \\ \wr & & & & & & & & \\ q_1' & \longrightarrow & q_2' & \longrightarrow & \cdots & \longrightarrow & q_n' & \longrightarrow & p' \end{array}$$

with all q_i's synchronized (if $m > 0$) and q_i''s synchronized (if $n > 0$).

Proof. Obvious by definition of $\tau \odot \sigma$. □

Proposition 10.1.5. *Consider $\sigma : A \vdash B$ and $\tau : B \vdash C$ causally deterministic.*
Then, $\tau \odot \sigma : A \vdash C$ is causally deterministic.

Proof. Consider $p \sim\!\!\sim_{\tau \odot \sigma} p'$ a minimal conflict in $\tau \odot \sigma$, and its originating immediate conflict in $\tau \circledast \sigma$ as guaranteed by Lemma 10.1.4. First, assume that $p \sim\!\!\sim_{\tau \circledast \sigma} p'$, *i.e.* $m = n = 0$. By Lemma 10.1.3, this corresponds to an immediate conflict for σ or τ – assume the former, so that p_σ and p'_σ are defined and $p_\sigma \sim\!\!\sim_\sigma p'_\sigma$. But then since σ is causally deterministic, those are negative moves, hence negative in $\tau \odot \sigma$ as well.

Otherwise, we have $m > 0$ or $n > 0$, say the former so that the immediate conflict originates in $q \sim\!\!\sim_{\tau \circledast \sigma} q'$ with q synchronized. By Lemma 10.1.3, this corresponds to an immediate conflict for σ or τ – assume the former, so that q_σ and q'_σ are defined and $q_\sigma \sim\!\!\sim_\sigma q'_\sigma$. Since σ is causally deterministic, q_σ and q'_σ must be negative and by Lemma 6.1.17, their display to the game are conflicting. But this implies that q_τ and q'_τ are defined and conflicting as well. Additionally we must have $q_\tau \sim\!\!\sim_\tau q'_\tau$, or we would get a contradiction with the minimality of $q \sim\!\!\sim_{\tau \circledast \sigma} q'$. But since τ is causally deterministic this means that q_τ and q'_τ are negative, contradiction. □

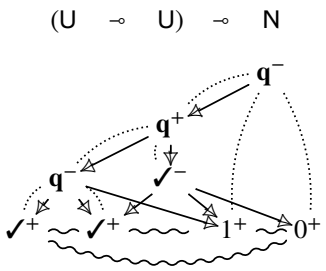


Figure 10.3: A causal strategy

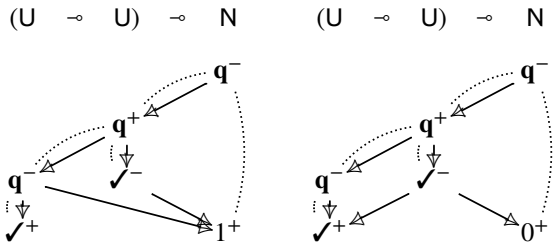


Figure 10.4: Two configurations of Figure 10.3

Full structure. For remaining operations on strategies (tensor, pairing, currying, promotion), it is immediate by construction that they preserve causal determinism. So:

Theorem 10.1.6. *The relative Seely $\sim$-category* **NTCG** *admits a lluf relative Seely sub-$\sim$-category* **NTCG-Det** *with morphisms restricted to causally deterministic strategies.*

Additionally, the least upper bound of a chain of causally deterministic strategies is still causally deterministic, hence the fixpoint combinator is causally deterministic. It also follows by direct inspection that the strategies used in the interpretation of $\mathsf{PCF}_{/\!/}$ are all causally deterministic; hence for any $\Gamma \vdash M : A$ in $\mathsf{PCF}_{/\!/}$, it follows that

$$[\![M]\!] \in \mathbf{NTCG\text{-}Det}_!([\![\Gamma]\!], [\![A]\!]) .$$

But causal determinism is not sufficient to characterise pure parallel computation.

10.1.2 Pre-innocence

So, what causal shapes are distinctive of pure parallel computation?

Pure parallel programs may spawn parallel threads, which remain independent in the absence of interference. Once they terminate the program may take new actions that depend on their results, causally "merging" them. A typical causal strategy featuring this behaviour, for $x : \mathbb{U}, y : \mathbb{U} \vdash x \parallel y : \mathbb{U}$, appears in Figure 10.6. The slogan is:

"Player may merge threads than he himself has spawned".

In contrast, both diagrams of Figure 10.4 bear signs of interference. In the first, the answer 1^+ depends on $\mathbf{q}^-$: the program somehow observes if the function has called its argument, which is only possible if the argument performs some side-effect that the program observes. In the second, $\checkmark^+$ depends on $\checkmark^-$; but likewise this can only occur if the termination of the function triggers a side-effect. In both cases, this is witnessed by Player "merging" causal chains which forked at Opponent moves.

To ban such interference, the slogan is:

"Player may not merge threads spawned by Opponent".

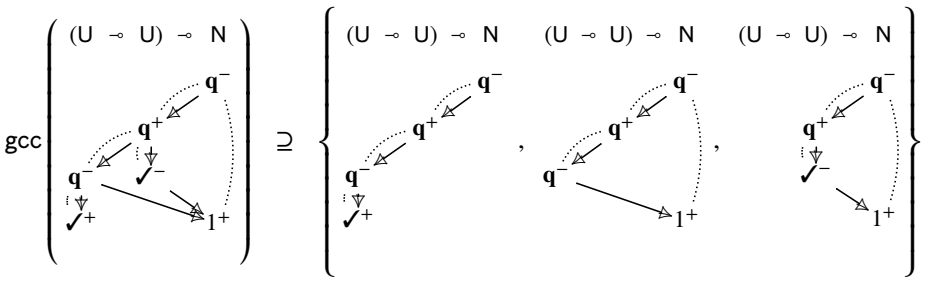


Figure 10.5: Maximal grounded causal chains of a causal strategy

To define parallel innocence, our first step is to introduce a formal notion of "thread":

Definition 10.1.7. *Consider A a board, and $\sigma : A$ a causal strategy.*
*A **grounded causal chain (gcc)** in σ is $\rho = \{\rho_1, \dots, \rho_n\} \subseteq |\sigma|$ forming*

$$\rho_1 \rightarrowtriangle_\sigma \cdots \rightarrowtriangle_\sigma \rho_n$$

a chain with ρ_1 minimal with respect to $\leq_\sigma$. We write $\mathsf{gcc}(\sigma)$ for the gccs in σ.

A gcc is just a set, written $\rho = \rho_1 \rightarrowtriangle_\sigma \cdots \rightarrowtriangle_\sigma \rho_n \in \mathsf{gcc}(\sigma)$ with the causal ordering from $\leq_\sigma$. If also $\rho_1 \rightarrowtriangle_\sigma \cdots \rightarrowtriangle_\sigma \rho_n \rightarrowtriangle_\sigma m \in \mathsf{gcc}(\sigma)$, then we write $\rho \rightarrowtriangle m = \rho \cup \{m\}$. Gccs are not necessarily down-closed: we show in Figure 10.5 all maximal gccs of a causal strategy. Of those, the second and third omit some dependencies of 1^+.

We may now make formal the idea of "only merging threads forked by Player".

Definition 10.1.8. *For A a board, a causally deterministic $\sigma : A$ is **pre-innocent** iff*

pre-innocent: *If $m^+ \in \sigma$ and $\rho_1 \rightarrowtriangle m, \rho_2 \rightarrowtriangle m \in \mathsf{gcc}(\sigma)$ are distinct, then $\min(\rho_1) = \min(\rho_2)$ and their least distinct moves are positive.*

The strategy of Figure 10.6 is pre-innocent. In contrast, that of Figure 10.3 is not – both configurations of Figure 10.4 fail pre-innocence. For instance, the second and third gccs of Figure 10.5 arrive at 1^+ but before that, the greatest common event is $\mathbf{q}^+$, which is positive: Player is merging (via 1^+) two gccs forked by Opponent.

It will follow later on that the *sequential pre-innocent* causal strategies exactly match the standard alternating innocent strategies of Definition 3.2.9; see Section 11.3.

However, for non-sequential strategies, pre-innocence is still incomplete.

10.1.3 Visibility

The problem arises as non-stability of pre-innocence under composition.

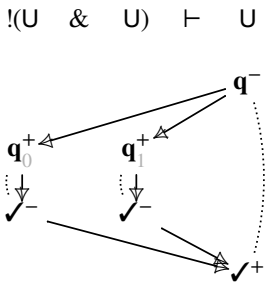


Figure 10.6: A typical pre-innocent strat.

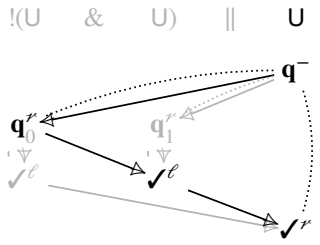


Figure 10.7: Partiality of views

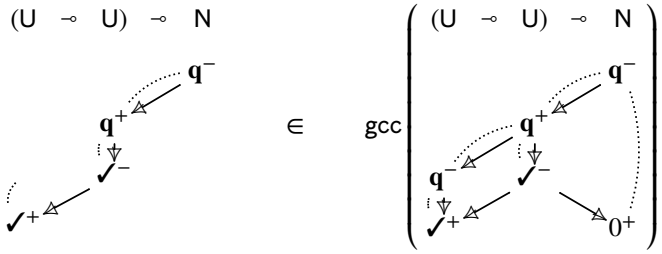


Figure 10.8: A gcc of a non-visible strategy, losing its pointer

A counter-example appears in Figure 10.9, which we postpone to the section devoted to compositionality of innocence. However, we can already explain the issue intuitively: the definition of pre-innocence relies on gccs which formalize a notion of *thread*. If that intuition is to be taken seriously, gccs should be valid executions of standalone sequential programs. But this is not the case: Figure 10.8 shows a gcc where the last move answers a question that *was not asked* within this gcc. This could not be a valid state of a sequential program, because the last move *loses its pointer*.

Visible strategies are simply those such that this does not happen.

Definition 10.1.9. *A causal strategy $\sigma : A$ is **visible** if it is:*

pointed: *for any $s \in \sigma$, there is a unique* $\mathrm{init}(s) \leq_\sigma$ *s minimal in σ,*
valid-gccs: *for all $\rho \in \mathsf{gcc}(\sigma)$, $\partial_\sigma(\rho) \in \mathscr{C}(A)$.*

We smuggle in *pointed*, which ensures that separate initial Opponent moves explore causally independent parts of the strategy. It is necessary for our proof of composition of visibility, but it is clearly less important than *valid-gccs*, mainly because it is already implied by pre-innocence (and also, automatic for $\sigma : A \vdash B$ with B strict). Hence, we shall tend in the future to brush over *pointed*, and identify *visible* and *valid-gccs*.

Valid-gccs means that every move in ρ *points* within ρ. This phrasing highlights the analogy with Definition 3.2.8, *i.e.* "Player always points in the P-view". It is indeed this

analogy that inspired the name[2]. But one must be wary: the alternating interpretation of sequential programs with state yields sequential P-visible strategies, but their causal interpretation (as in Figure 10.3) may not be visible. Visibility is very restrictive, it is not clear what sensible primitive would satisfy visibility but not pre-innocence.

The following lemma captures how a gcc may be regarded as a standalone thread.

Lemma 10.1.10. *Consider A a mixed board, and $\sigma : A$ a visible causal strategy.*
If $\rho = \rho_1 \rightarrowtriangle_\sigma \cdots \rightarrowtriangle_\sigma \rho_n \in \mathsf{gcc}(\sigma)$, $\overset{\frown}{(\cdot)}(\partial_\sigma(\rho)) = \overset{\frown}{(\cdot)}(\partial_\sigma(\rho_1) \ldots \partial_\sigma(\rho_n))$ is a P-view.

Proof. By Lemma 6.1.16, $\partial_\sigma(\rho)$ is alternating. By visibility, its prefixes are configurations of A. So, $\partial_\sigma(\rho) \in \Updownarrow\text{-}\mathsf{Plays}(A)$. By Lemma 6.1.16 again, the predecessor of $a^- \in \partial_\sigma(\rho)$ for $\rightarrowtriangle_A$ is its predecessor in $\partial_\sigma(\rho)$, *i.e.* Opponent always points to the previous move. The statement follows by definition of pointifixion. □

We may now define *parallel innocent* causal strategies, or just *innocent* for short.

Definition 10.1.11. *Consider $\sigma : A$ causally deterministic on board A.*
*It is **parallel innocent** if it is pre-innocent and **visible**.*

A standard innocent strategy as in Section 3.2.3, under its "causal" presentation, is a forest of P-views, *i.e.* a forest of (displayed) gccs. In that light the definition of parallel innocent strategies seems natural: they are generated no longer by a forest of P-views, but by a directed acyclic graph of P-views with additional conflict relation. This graph describes how threads are spawned, and then may merge, following the innocence discipline ensuring that Player may not create interference between Opponent's threads.

One of the main hurdles, in traditional game semantics, is to prove that innocent strategies compose. We now tackle this problem for parallel innocent strategies.

10.2 Composition of Visibility

First, we establish compositionality of visibility.

10.2.1 Justifiers in causal strategies

We introduce some machinery on *justifiers*. If $\sigma : A$ is a causal strategy on A some $-$-board, then as for plays, the immediate causality in A endows moves in σ with a notion of *justifier*. This extends to $\sigma : A \vdash B$ where A and B are $-$-boards:

Definition 10.2.1. *Consider A and B $-$-boards, and $\sigma : A \vdash B$ pointed.*
*We define the **justifiers** in σ by setting, for all $m, m' \in \sigma$,*

$$\begin{array}{lcll} \mathrm{j}(m) & = & m' & \text{if } \partial_\sigma(m') \rightarrowtriangle_{A \vdash B} \partial_\sigma(m), \\ \mathrm{j}(m) & = & \mathrm{init}(m) & \text{if } \partial_\sigma(m) \text{ minimal in } A, \end{array}$$

and undefined otherwise.

[2]This, plus as in traditional game semantics, visibility is a prerequisite for a working notion of innocence.

Since A and B are *forestial* and σ is pointed, $\mathsf{j}(-)$ is well-defined as a partial function on σ, only left undefined for $m \in \sigma$ minimal in σ. Note that assigning the justifier of m minimal in A to $\mathrm{init}(m)$ ensures that the assignment of justifiers is invariant under currying. It might be helpful to the reader to observe that a pointed causal strategy $\sigma : A \vdash B$ is visible iff for all $\rho \in \mathsf{gcc}(\sigma)$, for all $m \in \rho$, $\mathsf{j}(m) \in \rho$ as well: all gccs are closed under justifiers. We mention in passing this lemma:

Lemma 10.2.2. *Consider A, B $-$-boards and $\sigma : A \vdash B$ a pointed causal strategy.*

Then, for any non-initial $m \in \sigma$, we have $\mathsf{j}(m) <_\sigma m$. Moreover, if $\mathrm{pol}_\sigma(m) = -$, then $\mathsf{j}(m) \rightarrowtriangle_\sigma m$ is its (unique) immediate predecessor in σ.

Proof. As a map of es, ∂_σ locally reflects causality (Lemma 6.1.8), so $\mathsf{j}(m) <_\sigma m$ if the first clause of Definition 10.2.1 applies; for the other we clearly have $\mathrm{init}(m) <_\sigma m$.

The second fact is simply by Lemma 6.1.16. □

10.2.2 Justifiers in interactions

So as to prove visibility stable under composition, we first extend justifiers to *interactions* – consider A, B and C three $-$-boards, and $\sigma : A \vdash B$ and $\tau : B \vdash C$ pointed.

Definition 10.2.3. *We define $\mathsf{j} : |\tau \circledast \sigma| \rightharpoonup |\tau \circledast \sigma|$ as $\mathsf{j}(m) = m'$ if:*

(1) $\partial_{\tau\circledast\sigma}(m') \rightarrowtriangle_{A\|B\|C} \partial_{\tau\circledast\sigma}(m)$, *or*
(2) $\partial_{\tau\circledast\sigma}(m)$ *is minimal in A and* $m'_\sigma = \mathrm{init}(m_\sigma)$, *or*
(3) $\partial_{\tau\circledast\sigma}(m)$ *is minimal in B and* $m'_\tau = \mathrm{init}(m_\tau)$,

and undefined otherwise. We say that m' is the ***justifier*** *of m in $\tau \circledast \sigma$.*

This leaves $\mathsf{j}(m)$ undefined exactly if it corresponds to a minimal move in C. Clearly the two notions of justifier are compatible, in the sense that for all $m \in \tau \circledast \sigma$, if m_σ is defined then $\mathsf{j}(m)_\sigma$ is defined and equal to $\mathsf{j}(m_\sigma)$, and likewise for τ.

10.2.3 Views of gccs

We introduce the main technical device on visible causal interactions.

We use polarities in interactions as in Lemma 6.2.15, and annotate events accordingly. We also write *e.g.* $a^{-,\nearrow}$ to indicate that a has polarity $-$ *or* $\nearrow$. If $\rho \in \mathsf{gcc}(\tau \circledast \sigma)$ with last event m, we say that ρ **ends in** σ if m_σ is defined, and likewise for τ. We now define *views* of gccs, used to project a gcc of the interaction to gccs for both strategies.

Definition 10.2.4. *Take $\sigma : A \vdash B$ and $\tau : B \vdash C$ pointed with A, B, C $-$-boards.*

If $\rho \in \mathsf{gcc}(\tau \circledast \sigma)$ ends in σ, we (partially) define $\ulcorner \rho \urcorner^\sigma \in \mathsf{gcc}(\sigma)$ by:

$$\begin{array}{rcll}
\ulcorner \rho_0 \rightarrowtriangle \ldots \rightarrowtriangle \rho_n \rightarrowtriangle \rho_{n+1}^{\swarrow} \urcorner^\sigma & = & \ulcorner \rho_0 \rightarrowtriangle \ldots \rightarrowtriangle \rho_n \urcorner^\sigma \cup \{(\rho_{n+1})_\sigma\}\,, & \\
\ulcorner \rho_0 \rightarrowtriangle \ldots \rightarrowtriangle \rho_i \rightarrowtriangle \ldots \rightarrowtriangle \rho_{n+1}^{-,\nearrow} \urcorner^\sigma & = & \ulcorner \rho_0 \rightarrowtriangle \ldots \rightarrowtriangle \rho_i \urcorner^\sigma \cup \{(\rho_{n+1})_\sigma\} & \text{if } \mathsf{j}(\rho_{n+1}) = \rho_i \text{ in } A \text{ or } B\,, \\
\ulcorner \rho_0 \rightarrowtriangle \ldots \rightarrowtriangle \rho_i \rightarrowtriangle \ldots \rightarrowtriangle \rho_{n+1}^{\nearrow} \urcorner^\sigma & = & \{(\rho_{n+1})_\sigma\} & \text{if } \rho_{n+1} \text{ minimal in } B\,,
\end{array}$$

undefined otherwise. If $\rho \in \mathsf{gcc}(\tau \circledast \sigma)$ ends in τ, we (partially) define $\ulcorner \rho \urcorner^\tau \in \mathsf{gcc}(\tau)$:

$$\begin{array}{rcll}
\ulcorner \rho_0 \rightarrowtriangle \ldots \rightarrowtriangle \rho_n \rightarrowtriangle \rho_{n+1}^{\nearrow} \urcorner^\tau & = & \ulcorner \rho_0 \rightarrowtriangle \ldots \rightarrowtriangle \rho_n \urcorner^\tau \cup \{(\rho_{n+1})_\tau\}\,, & \\
\ulcorner \rho_0 \rightarrowtriangle \ldots \rightarrowtriangle \rho_i \rightarrowtriangle \ldots \rightarrowtriangle \rho_{n+1}^{-,\swarrow} \urcorner^\tau & = & \ulcorner \rho_0 \rightarrowtriangle \ldots \rightarrowtriangle \rho_i \urcorner^\tau \cup \{(\rho_{n+1})_\tau\} & \text{if } \mathsf{j}(\rho_{n+1}) = \rho_i\,;
\end{array}$$

when defined, $\ulcorner \rho \urcorner^{\sigma} \in \mathsf{gcc}(\sigma)$ *is the* ***σ-view*** *of* ρ *and* $\ulcorner \rho \urcorner^{\tau} \in \mathsf{gcc}(\tau)$ *the* ***τ-view*** *of* ρ.

These definitions closely follow Definition 3.2.7. The last clause is only needed for $\ulcorner - \urcorner^{\sigma}$ and not $\ulcorner - \urcorner^{\tau}$, because an initial event in C must be the first event of ρ anyway.

That this yields gccs of σ and τ rests on Lemma 6.2.15, and courtesy of σ and τ. The σ-view and the τ-view are in principle only partially defined, because it may be, when attempting to follow the opponent's pointer, that that justifier lies outside the gcc. For instance $\ulcorner \rho \urcorner^{\tau}$, for ρ in Figure 10.7, is not well-defined: when attempting to compute $\ulcorner \mathbf{q}^{-}\mathbf{q}_0^{\nearrow}\checkmark_1^{\ell} \urcorner^{\tau}$, none of the clauses apply as $\mathsf{j}(\checkmark_1^{\ell}) = \mathbf{q}_1^{\nearrow}$ is outside ρ. The bulk of the proof of stability of visibility under composition, is to show that this cannot happen:

Proposition 10.2.5. *Let* $\sigma : A \vdash B$ *and* $\tau : B \vdash C$ *be* visible *causal strategies. Then, the views of gccs of* $\tau \circledast \sigma$ *as in Definition 10.2.4 are always well-defined.*

Proof. We prove by induction on ρ that, for all prefixes of ρ,

(1) if ρ ends in σ, then $\ulcorner \rho \urcorner^{\sigma} \in \mathsf{gcc}(\sigma)$ is well-defined,
(2) if ρ ends in τ, then $\ulcorner \rho \urcorner^{\tau} \in \mathsf{gcc}(\tau)$ is well-defined.

Assume ρ ends in τ. If the last move has polarity $-$, then either it is initial and there is nothing to prove, or by Lemma 6.2.15 its justifier is its predecessor in ρ, so $\ulcorner \rho \urcorner^{\tau} \in \mathsf{gcc}(\tau)$ follows by induction hypothesis (in that case ρ does not end in σ).

If the last move has pol. $\nearrow$, write $\rho = \rho' \rightarrowtriangle m \rightarrowtriangle n^{\nearrow}$. By Lemma 6.2.15, $m_{\tau} \rightarrowtriangle_{\tau} n_{\tau}$, so m is in τ. By induction hypothesis, $\kappa = \ulcorner \rho' \rightarrowtriangle m \urcorner^{\tau} \in \mathsf{gcc}(\tau)$, so

$$\kappa \rightarrowtriangle n_{\tau} = \ulcorner \rho \urcorner^{\tau} \in \mathsf{gcc}(\tau)$$

as well. But if ρ ends in σ *and* τ (*i.e.* in B), we must further prove that $\ulcorner \rho \urcorner^{\sigma} \in \mathsf{gcc}(\sigma)$. In that case, since $\ulcorner \rho \urcorner^{\tau} \in \mathsf{gcc}(\tau)$ and τ is visible, it follows that $\mathsf{j}(n_{\tau}) \in \ulcorner \rho \urcorner^{\tau}$, but this entails $\mathsf{j}(n) \in \rho$. Hence the second clause of Definition 10.2.4 applies, and we conclude by induction hypothesis. If ρ finishes in σ, the reasoning is symmetric. □

From this, we are now ready to conclude:

Proposition 10.2.6. *Let* $\sigma : A \vdash B$ *and* $\tau : B \vdash C$ *be visible causal strategies. Then,* $\tau \odot \sigma : A \vdash C$ *is also visible.*

Proof. First, we show that $\tau \odot \sigma$ is pointed. there are distinct $q, q' \in \tau \odot \sigma$ minimal such that $q \leq_{\tau \odot \sigma} p$ and $q' \leq_{\tau \odot \sigma} p$. It is a direct consequence of courtesy (via Lemma 6.1.16) that then, q, q' must still be minimal in $\tau \circledast \sigma$. Then there must be some r in $\tau \circledast \sigma$ minimal such that there is $m \rightarrowtriangle_{\tau \circledast \sigma} r$ and $n \rightarrowtriangle_{\tau \circledast \sigma} r$ with $[m]_{\tau \circledast \sigma}$ and $[n]_{\tau \circledast \sigma}$ disjoint. Now, we distinguish cases depending on the polarity of r. If it is ℓ, then by Lemma 6.2.15, m_{σ} and n_{σ} are defined and $m_{\sigma} \rightarrowtriangle_{\sigma} r_{\sigma}$ and $n_{\sigma} \rightarrowtriangle_{\sigma} r_{\sigma}$. But since σ is pointed, $[m_{\sigma}]_{\sigma}$ and $[n_{\sigma}]_{\sigma}$ cannot be disjoint, contradiction. The case for $\nearrow$ is symmetric.

Now, we show visibility. For that, we prove by induction on ρ that for all $\rho \in \mathsf{gcc}(\tau \circledast \sigma)$, $\partial_{\tau \circledast \sigma}(\rho) \in \mathscr{C}(A \parallel B \parallel C)$. If ρ is empty it is clear; take $\rho \rightarrowtriangle m \in \mathsf{gcc}(\tau \circledast \sigma)$. By induction hypothesis, $\partial_{\tau \circledast \sigma}(\rho) \in \mathscr{C}(A \parallel B \parallel C)$, we only need that the justifier of m is in ρ. We reason by cases on the polarity of m: if it is ℓ, then by Proposition 10.2.5 $\ulcorner \rho \rightarrowtriangle m \urcorner^{\sigma} \in \mathsf{gcc}(\sigma)$. But since σ is visible, the justifier of m_{σ} appears in $\ulcorner \rho \urcorner^{\sigma}$, hence

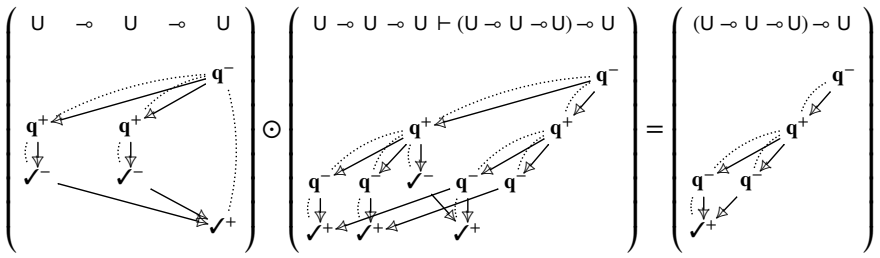


Figure 10.9: Failure of preservation of pre-innocence under composition

the justifier of m appears in ρ. The other cases are symmetric or trivial.

Now, take $\rho_{\odot} \in \mathsf{gcc}(\boldsymbol{\tau} \odot \boldsymbol{\sigma})$. By definition of $\boldsymbol{\tau} \odot \boldsymbol{\sigma}$, there is a (non-necessarily unique) $\rho_{\circledast} \in \mathsf{gcc}(\boldsymbol{\tau} \circledast \boldsymbol{\sigma})$ s.t. $\rho_{\odot}$ comprises exactly those events of $\rho_{\circledast}$ in A or C. By the observation above, $\partial_{\boldsymbol{\tau} \circledast \boldsymbol{\sigma}}(\rho_{\circledast}) \in \mathscr{C}(A \parallel B \parallel C)$, hence $\partial_{\boldsymbol{\tau} \odot \boldsymbol{\sigma}}(\rho_{\odot}) \in \mathscr{C}(A \parallel C)$. □

10.2.4 The Relative Seely ∼-Category NTCG-Vis

In this monograph, we are particularly interested in visibility as a means to ensure that innocence is preserved under composition. However, it is noteworthy that visibility indeed supports all the constructions involved in the relative Seely category structure.

Theorem 10.2.7. *There is* **NTCG-Vis**, *a relative Seely sub-∼-category of* **NTCG**, *with morphisms restricted to visible causal strategies.*

Proof. As causality in boards is forestial, so is the causality of copycat – it follows immediately that copycat is visible, and this immediately extends to lifted strategies. By Proposition 10.2.6, visibility is stable under composition. For all other operations (tensor, pairing, currying, promotions), it is immediate that visibility is preserved. □

This extends to the fixpoint operation and – by inspection – to all primitive strategies involved in the interpretation of $\mathsf{PCF}_{/\!/}$, so that any term $\Gamma \vdash M : A$ of $\mathsf{PCF}_{/\!/}$ yields a visible causal strategy $[\![M]\!] \in \mathbf{NTCG\text{-}Vis}_!([\![\Gamma]\!], [\![A]\!])$. In contrast, the interpretation of $\mathsf{IA}_{/\!/}$ does *not* satisfy visibility: the non-visible behaviour from Figure 10.8 comes from

$$\vdash \mathbf{newref}\ x\ \mathbf{in}\ \lambda f^{\mathbb{U}\to\mathbb{U}}.\ f\ (x := \mathbf{tt});\ !r : (\mathbb{U} \to \mathbb{U}) \to \mathbb{B}\,, \tag{10.1}$$

a term of IA. Though visibility does not match tightly any of our programming languages, it will play in important role in this work. But next, we show that in the presence of visibility, innocence becomes stable under composition.

10.3 Composition of Innocence

We start by showing "what could go wrong". In Figure 10.9, we show a counter-

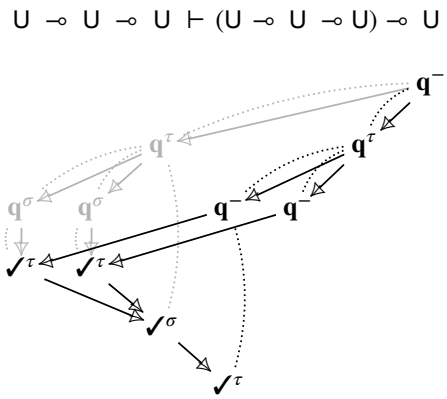


Figure 10.10: Illegal causal merge

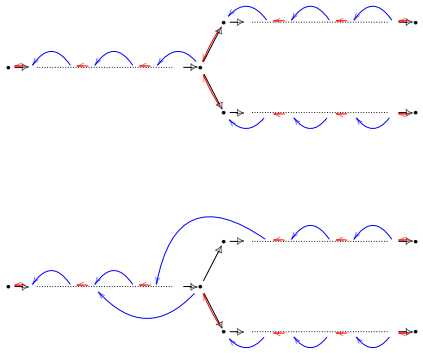


Figure 10.11: Merging paths in G

example to the stability under composition of pre-innocence without visibility, with the corresponding interaction appearing as Figure 10.10. Let us attempt to explain the phenomenon, calling $\boldsymbol{\sigma}$ the left hand side strategy (parallel composition) and $\boldsymbol{\tau}$ the right hand side one – observe that the dotted lines include the justification relations from Definition 10.2.1 rather than just those coming from the arena. Imagine that $\boldsymbol{\tau}$ wants to perform an illegal causal merge between the two argument calls of its argument of type $\mathsf{U} \multimap \mathsf{U} \multimap \mathsf{U}$. By pre-innocence it cannot do so directly. However, it can outsource the merge to $\boldsymbol{\sigma}$ by linking (legally with respect to pre-innocence, but illegally with respect to visibility) the arguments of the parallel composition to those that it wants to merge.

We shall prove that this cannot happen in the presence of visibility. Let us fix, until the end of the section, two visible causal strategies $\boldsymbol{\sigma} : A \vdash B$ and $\boldsymbol{\tau} : B \vdash C$.

10.3.1 The "forking lemma"

Taking a closer look at Figure 10.10, we highlight the two illegally merging gccs in the interaction: while $\boldsymbol{\sigma}$ is responsible for the merge, the point where these gccs forked is external, outside the scope of $\boldsymbol{\sigma}$! The next lemma, dubbed the "forking lemma", forbids this: it implies that visible strategies cannot unknowingly close an Opponent fork.

If $\rho = \rho_1 \rightarrowtriangle \ldots \rightarrowtriangle \rho_n$ is a gcc and $1 \leq i \leq n$, $\rho_{\leq i}$ is $\rho_1 \rightarrowtriangle \ldots \rightarrowtriangle \rho_i$. Two gccs ρ, κ are **forking** iff $\rho \cap \kappa \neq \emptyset$, and for all i, j, if $\rho_i = \kappa_j$ then $\rho_{\leq i} = \kappa_{\leq j}$. If ρ, κ are two forking gccs, we write $\mathsf{gce}(\rho, \kappa)$ for their **greatest common event**. Notice that despite the terminology, two forking gccs can be prefix of one another and never truly go separate ways; but if they do diverge, their remainder must be independent.

Lemma 10.3.1 (Forking lemma)**.** *Consiser $\boldsymbol{\sigma} : A \vdash B$ and $\boldsymbol{\tau} : B \vdash C$ visible strategies. Let $\rho, \kappa \in \mathsf{gcc}(\boldsymbol{\tau} \circledast \boldsymbol{\sigma})$ be forking gccs ending in $\boldsymbol{\sigma}$, s.t. $\ulcorner \rho \urcorner^{\boldsymbol{\sigma}} \cap \ulcorner \kappa \urcorner^{\boldsymbol{\sigma}} \neq \emptyset$ and $\mathsf{gce}(\ulcorner \rho \urcorner^{\boldsymbol{\sigma}}, \ulcorner \kappa \urcorner^{\boldsymbol{\sigma}})$ negative (the least distinct events, if any, are positive).*

Then, $\mathsf{gce}(\ulcorner\rho\urcorner^{\sigma}, \ulcorner\kappa\urcorner^{\sigma}) = \mathsf{gce}(\rho,\kappa)_{\sigma}$. *The symmetric property holds for* τ.

Proof. We only detail the proof for σ, the proof for τ is exactly the same. We build a directed graph G with vertices $\rho \cup \kappa$, and edges the (disjoint) union of the sets:

$$\begin{array}{rcl} O\text{-edges} & = & \{(m_1, m_2^{\ell}) \mid \mathsf{j}(m_1) = m_2\} \\ P\text{-edges} & = & \{(m_1^{\ell}, m_2) \mid m_2 \rightarrowtriangle_{\tau \circledast \sigma} m_1\} \end{array}$$

where $(-)^{\ell}$ indicates the polarity. Each vertex is source of at most one edge, and following edges consists exactly in computing the σ-view. If ρ and κ have the same final move, then $\rho = \kappa$. Otherwise, consider the two paths in G starting with these distinct final moves. Since $\ulcorner\rho\urcorner^{\sigma} \cap \ulcorner\kappa\urcorner^{\sigma} \neq \emptyset$, these paths must intersect – Figure 10.11 represents a typical G with O-edges in blue and P-edges in red with the two typical cases.

These paths meet at a vertex of incoming degree at least 2; but vertices receive only O-edges, or only P-edges. For the former (as in the bottom of Figure 10.11), then $\mathsf{gce}(\ulcorner\rho\urcorner^{\sigma}, \ulcorner\kappa\urcorner^{\sigma})$ is positive, which contradicts the hypothesis. For the latter (as in the top of Figure 10.11), we remark that P-edges are immediate causal links in $\tau \circledast \sigma$; and there is at most one event in $\rho \cup \kappa$ causing two distinct events: $\mathsf{gce}(\rho,\kappa)$. □

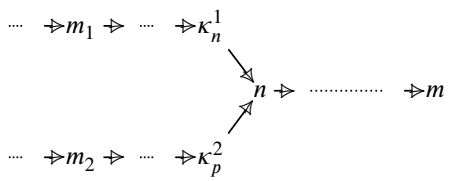
This provides the core argument for the compositionality of pre-innocence: if a pre-innocent strategy merges two threads, by pre-innocence its *views* of these two threads fork positively. But then the forking lemma ensures that this strategy sees the actual forking point for these threads – which therefore cannot be due to the external Opponent.

10.3.2 Stability of pre-innocence

Now, much of the proof consists in restricting the causal shapes in $\tau \circledast \sigma$ corresponding to a causal merge in $\tau \odot \sigma$, so that the forking lemma applies.

Proposition 10.3.2. *Consider* $\sigma : A \vdash B$ *and* $\tau : C \vdash C$ *visible causal strategies.*
If $\sigma : A \vdash B$ *and* $\tau : B \vdash C$ *are pre-innocent, then so is* $\tau \odot \sigma$.

Proof. Consider $m \in \tau \odot \sigma$ and distinct $\rho^1 \rightarrowtriangle m, \rho^2 \rightarrowtriangle m \in \mathsf{gcc}(\tau \odot \sigma)$. *W.l.o.g.* assume that whenever $\rho^1_i = \rho^2_j$, $\rho^1_{\leq i} = \rho^2_{\leq j}$ – or we can ensure this by changing m and ρ^i, keeping the same least distinct events. Likewise, since ρ^1, ρ^2 are distinct, *w.l.o.g.* their last moves $m_1 \in \rho^1, m_2 \in \rho^2$ are distinct – or we may replace m with an earlier causal merge. These gccs ρ^1 and ρ^2 may be completed to $\kappa^1 \rightarrowtriangle m, \kappa^2 \rightarrowtriangle m \in \mathsf{gcc}(\tau \circledast \sigma)$ such that ρ^i consists exactly of the events of κ^i occurring in A or C. Necessarily, the greatest visible events of κ^1 and κ^2 are m_1 and m_2 respectively. Call n the least common event of $\kappa^1 \rightarrowtriangle m$ and $\kappa^2 \rightarrowtriangle m$ above m_1 and m_2 (which might not be m). The situation is:

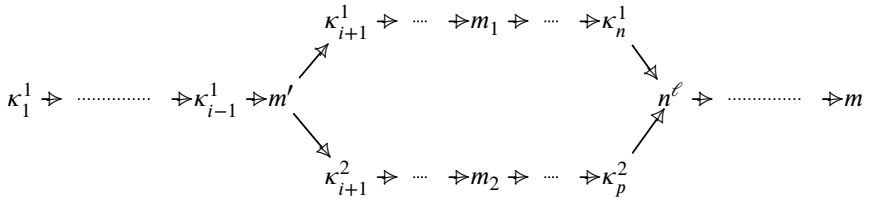


with m_1, m_2 and m visible, and no one visible in between. We reason on the polarity of n in $\tau \circledast \sigma$. By Lemma 6.2.15 and since arenas are forestial, it cannot be negative, so its polarity is either ℓ or $\nearrow$. Assume it is ℓ – the other case is symmetric.

The polarity of n is positive for σ, so we compute the σ-*views*:

$$\ulcorner \kappa^1_{\leq n} \urcorner^\sigma \twoheadrightarrow n^{\urcorner\sigma}, \qquad \ulcorner \kappa^2_{\leq p} \urcorner^\sigma \twoheadrightarrow n^{\urcorner\sigma} \quad \in \quad \mathsf{gcc}(\sigma)\,,$$

respectively $\ulcorner \kappa^1_{\leq n} \urcorner^\sigma \twoheadrightarrow n_\sigma$ and $\ulcorner \kappa^2_{\leq p} \urcorner^\sigma \twoheadrightarrow n_\sigma$, with $\ulcorner \kappa^1_{\leq n} \urcorner^\sigma$ and $\ulcorner \kappa^2_{\leq p} \urcorner^\sigma$ distinct as they respectively contain κ^1_n and κ^2_p. Since σ is pointed, $\ulcorner \kappa^1_{\leq n} \urcorner^\sigma \cap \ulcorner \kappa^2_{\leq p} \urcorner^\sigma \neq \emptyset$, so $\kappa^1 \cap \kappa^2 \neq \emptyset$ as well. Call m' the gce of κ^1 and κ^2, necessarily below m_1 and m_2, then:



assuming *w.l.o.g.* that $\kappa^1_{\leq i} = \kappa^2_{\leq i}$ (changing the beginning of κ^2 if required). Summing up some properties, $\xi^1 = \kappa^1_{i+1} \dots \kappa^1_n$ and $\xi^2 = \kappa^2_{i+1} \dots \kappa^2_p$ are disjoint. Thus the σ-views

$$\ulcorner \kappa^1_{\leq n} \urcorner^\sigma\,, \quad \ulcorner \kappa^2_{\leq p} \urcorner^\sigma \quad \in \quad \mathsf{gcc}(\sigma)$$

are *forking*: they coincide on a prefix and disjoint after. Indeed, since ξ^1 and ξ^2 are disjoint, any common event is in $\kappa^1_1 \twoheadrightarrow \dots \twoheadrightarrow \kappa^1_i$, before which the σ-views coincide.

Since σ is pre-innocent, the least distinct moves m'_1 and m'_2 of $\ulcorner \kappa^1_{\leq n} \urcorner^\sigma$ and $\ulcorner \kappa^2_{\leq p} \urcorner^\sigma$ are positive. Thus their common immediate predecessor is negative – but it is also their greatest common event, since $\ulcorner \kappa^1_{\leq n} \urcorner^\sigma$ and $\ulcorner \kappa^2_{\leq p} \urcorner^\sigma$ are forking. Thus by Lemma 10.3.1,

$$\mathsf{gce}(\ulcorner \kappa^1_{\leq n} \urcorner^\sigma, \ulcorner \kappa^2_{\leq p} \urcorner^\sigma) = \mathsf{gce}(\kappa^1_{\leq n}, \kappa^2_{\leq p})_\sigma = m'_\sigma\,,$$

so m' is negative for σ. In $\tau \circledast \sigma$, m' is negative or in B – in both cases the least visible events in ξ^1 and ξ^2 are positive, but those are our least distinct events of ρ_1 and ρ_2. □

10.3.3 The Relative Seely Category NTCG-Inn

As for visibility, we have:

Theorem 10.3.3. *There is* **NTCG-Inn**, *a relative Seely sub-$\sim$-category of* **NTCG-Det**, *with morphisms restricted to parallel innocent causal strategies.*

Proof. For composition, this is via Propositions 10.1.5, 10.2.6 and 10.3.2. Parallel innocence of copycat is immediate, and that extends to lifted strategies. For all other operations on strategies, preservation of innocence is direct. □

Again this extends to the fixpoint operation and to all primitive strategies involved in the interpretation of $\mathsf{PCF}_{/\!/}$, so that any term $\Gamma \vdash M : A$ of $\mathsf{PCF}_{/\!/}$ yields

$$[\![M]\!] \in \mathbf{NTCG\text{-}Inn}_!([\![\Gamma]\!], [\![A]\!])$$

a parallel innocent strategy.

Note that though we bundled causal determinism with parallel innocence in this monograph, they are actually independent in the sense that stability under composition of parallel innocence does not need causal determinism. However, without determinism, parallel innocence as presented here is insufficient to obtain a definability result for a non-deterministic extension of $\mathsf{PCF}_{/\!/}$. We have not yet been able to give a working notion of non-deterministic parallel innocence – see Section 14.2.1 for a discussion.

10.3.4 Complement: the "Bang Lemma"

Within the relative Seely $\sim$-category **NTCG-Inn**, one can prove the "bang lemma" from AJM games [Abramsky et al., 2000] – it is an important technical lemma expressing that essentially, all innocent strategies on $!A \vdash !B$ are the promotion of some strategy on $!A \vdash B$. Note that this does not hold without innocence: a non-innocent strategy on $!B$ may very well impose constraints accross its different copies. In AJM games, the bang lemma depends on *history-freeness*; it does not exist in HO games in the absence of a linear decomposition. Fix A and B two strict boards, and $\sigma : !A \vdash !B$ innocent.

A minimal $m \in \sigma$ displays to $\partial_\sigma(m) = (2, (i, b))$ for some $i \in \mathbb{N}$ – we say that m **is in copy** i. Being innocent, σ is *pointed* and every event has a unique minimal cause; so we say that arbitrary $m \in \sigma$ **is in copy** i if its unique minimal cause is. Let us write

$$|\sigma_i| = \{m \in |\sigma| \mid m \text{ is in copy } i\}$$

for the set of events of σ in copy i. By construction, for $i, j \in \mathbb{N}$ distinct, $|\sigma_i|$ and $|\sigma_j|$ are disjoint, and each inherit from σ the structure of an ess. Furthermore, $\sigma_i : !A \vdash B$ with the display map ∂_{σ_i} defined in the obvious way. Moreover, we have:

Lemma 10.3.4. *As event structures with symmetries, we have* $\sigma \cong \|_{i\in\mathbb{N}}\sigma_i$.

Proof. The non-trivial point is that no immediate conflict in σ can span accross copies: if $s \sim\!\!\!\!\sim_\sigma s'$ then $\mathrm{pol}(s) = \mathrm{pol}(s') = -$ by *causal determinism*, and $\partial_\sigma(s) \sim\!\!\!\!\sim \partial_\sigma(s')$ as well by Lemma 6.1.17. By definition of bang, this entails that $\partial_\sigma(s)$ and $\partial_\sigma(s')$ are in a common copy of B, *or* of A. If it is B then we are done. If it is A then from polarity reasons they cannot be minimal in A, and since A is strict this entails that they are above (for $\leq_{!A}$) the *same* minimal event in A. So s and s' are above a common move in σ, which entails that they are in the same copy. □

They key argument, in the bang lemma, is that these copies are interchangeable:

Lemma 10.3.5. *For any* $i, j \in \mathbb{N}$*, we have* $\sigma_i \approx \sigma_j$.

Proof. We exploit Lemma 7.1.9 and build an iso between the domains of configurations, compatible with symmetry. Consider $x_i \in \mathscr{C}(\boldsymbol{\sigma}_i)$, displaying to

$$\partial_{\boldsymbol{\sigma}}(x_i) = x_A \parallel \{i\} \times x_B \,.$$

By definition of the board constructions, there is a symmetry exchanging i and j:

$$\theta^-_{i,j} : x_A \parallel \{i\} \times x_B \cong^-_{!A \vdash !B} x_A \parallel \{j\} \times x_B \,,$$

and our isomorphism will transport x_i along this negative symmetry. Indeed by Lemma 7.2.6, there are unique $x_j \in \mathscr{C}(\boldsymbol{\sigma})$ and $\psi : x_i \cong_{\boldsymbol{\sigma}} x_j$ s.t. $\partial_{\boldsymbol{\sigma}}\psi = \theta^+ \circ \theta^-_{i,j}$ for some

$$\theta^+ : x_A \parallel \{j\} \times x_B \cong^+_{!A \parallel !B} y_A \parallel \{j\} \times y_B$$

where we know that j is unchanged by definition of the positive symmetries of the bang. Therefore, $x_j \in \mathscr{C}(\boldsymbol{\sigma}_j)$ as required. Monotonicity of this operation and the fact that it is a bijection between configurations follow from the *uniqueness* clause for Lemma 7.2.6; compatibility with symmetry follows from composition with the symmetry ψ.

This induces an isomorphism of ess $\varphi : \boldsymbol{\sigma}_i \approx \boldsymbol{\sigma}_j$ which we must still check is a positive isomorphism. But for $x_i \in \mathscr{C}(\boldsymbol{\sigma}_i)$, the symmetry θ^+ above entails

$$\partial_{\boldsymbol{\sigma}_i}(x_i) = x_A \parallel x_B \quad \overset{\varphi^+}{\cong^+_{!A \vdash B}} \quad y_A \parallel y_B = \partial_{\boldsymbol{\sigma}_j}(x_j)$$

ensuring that the triangle commutes up to positive symmetry as required. □

Next we lift a positive isomorphism on one copy index to the whole strategy:

Lemma 10.3.6. *Consider A and B strict boards, and $\boldsymbol{\sigma}, \boldsymbol{\tau} : !A \vdash !B$ parallel innocent. If $\boldsymbol{\sigma}_0 \approx \boldsymbol{\tau}_0$, then $\boldsymbol{\sigma} \approx \boldsymbol{\tau}$.*

Proof. By Lemma 10.3.5, for $i \in \mathbb{N}$, $\boldsymbol{\sigma}_i \approx \boldsymbol{\sigma}_0 \approx \boldsymbol{\tau}_0 \approx \boldsymbol{\tau}_i$, and thus we have

$$\|_{i \in \mathbb{N}} \boldsymbol{\sigma}_i \approx \|_{i \in \mathbb{N}} \boldsymbol{\tau}_i$$

lifted to an isomorphism of ess $\boldsymbol{\sigma} \cong \boldsymbol{\tau}$ by Lemma 10.3.4. It is then straightforward that this isomorphisms is compatible with display maps up to positive symmetries. □

We may finally deduce the bang lemma:

Lemma 10.3.7. *For A, B strict boards and $\boldsymbol{\sigma} \in$ **NTCG**-**Inn**$(!A, !B)$ parallel innocent,*

$$(\mathsf{der}_B \odot \boldsymbol{\sigma})^! \approx \boldsymbol{\sigma} \,.$$

Proof. From Proposition 7.3.1, it is direct to establish $\mathsf{der}_B \odot \boldsymbol{\sigma} \approx \boldsymbol{\sigma}_0$. Therefore, by Lemma 10.3.6, for any $\boldsymbol{\sigma}, \boldsymbol{\tau} : !A \vdash !B$, if $\mathsf{der}_B \odot \boldsymbol{\sigma} \approx \mathsf{der}_B \odot \boldsymbol{\tau}$, then $\boldsymbol{\sigma} \approx \boldsymbol{\tau}$. Now as

$$\mathsf{der}_B \odot (\mathsf{der}_B \odot \boldsymbol{\sigma})^\dagger \approx \mathsf{der}_B \odot \boldsymbol{\sigma}$$

by the laws of relative Seely categories, the lemma follows. □

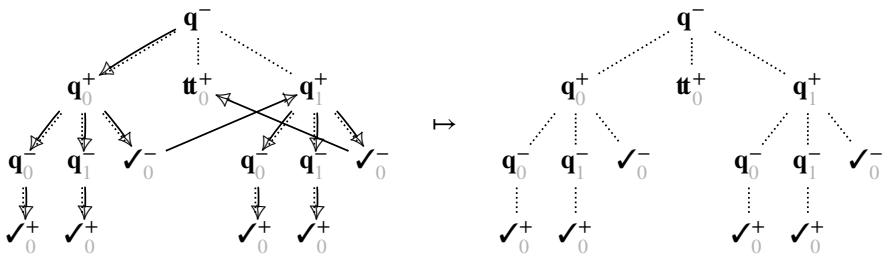


Figure 10.12: A witness and its display

10.4 Relational Collapse

Most consequences of parallel innocence will be presented in Chapter 12, when we focus on finite definability for those and intensional full abstraction for $\mathsf{PCF}_{/\!/}$. But we explore now a crucial consequence of parallel innocence (more precisely, of visibility): it allows a simple, transparent translation to the relational model. This will be useful technically later on, but we also regard this as an important contribution in its own right.

10.4.1 Stopping Positions and Witnesses

Intuitively, a causal strategy records observable computational events, along with the causal dependencies between them, but we may also regard it as presenting all positions reached in the sense of the relational model, enriching them with causal information.

Forgetting causality. Recall that a causal strategy $\sigma : A$ on $-$-board A is specified by an internal event structure σ, along with a *display map*

$$\partial_\sigma : \sigma \to A\,.$$

A configuration $x_A \in \mathscr{C}(A)$ is **reached by** σ iff there is $x^\sigma \in \mathscr{C}(\sigma)$, its **witness**, such that $\partial_\sigma x^\sigma = x_A$. Intuitively, and as a first approximation, the relational model only remembers which configurations $x_A \in \mathscr{C}(A)$ have a witness (though we shall see below that the relational model only tracks certain configurations considered as *completed* executions, and that *symmetry* entails that points in the relational models should be certain symmetry classes rather than mere configurations).

As an example, we show in Figure 10.12 a configuration of the strategy σ for

$$\vdash \lambda f^{\mathbb{U}\to\mathbb{U}}.\, f\,\mathbf{skip};\, f\,\mathbf{skip};\, \mathbf{tt} : (\mathbb{U}\to\mathbb{U})\to\mathbb{B}\,;$$

where the configuration appears on the left hand side. This diagram follows our long-standing convention to draw $x^\sigma \in \mathscr{C}(\sigma)$ with events displayed as their image via ∂_σ,

with the diagram showing immediate causality both in σ (drawn as $\multimap$) and in A (drawn as dotted lines). Thus, its display via ∂_σ is obtained diagrammatically simply as the erasure of the $\multimap$ relation, moving towards the right hand side of the diagram.

Our claim is that the diagram on the right hand side of Figure 10.12 corresponds (up to symmetry, see below) to an element in the interpretation of the type $(\mathbb{U} \to \mathbb{U}) \to \mathbb{B}$ in the relational model: that point resulting in $\mathbf{tt}$, where the argument is called twice, and where in turn each call of the argument calls *its* argument twice – written

$$\alpha = ([([\checkmark, \checkmark], \checkmark), ([\checkmark, \checkmark], \checkmark)], \mathbf{tt})$$

in standard relational model notation, but (ignoring copy indices) the reader should be able to see how the same information is conveyed on the right hand side of Figure 10.12: each "atom" in α corresponds to a pair of a question and an answer. Then, moving from concurrent games to the relational model essentially consists in forgetting the relation $\multimap$ in all configurations – so that we may even regard concurrent games as the relational model enriched with causal information. We now aim to make this more formal.

Positions. We wish to extract from a board a set matching the relational interpretation. To that end, we start by examining the most constraining constructor, which is

$$!X = \mathscr{M}_f(X)\,,$$

the *bang* of X a set, defined as the set of *finite multisets*. In contrast, the *configurations* of a $-$-board $!A$ correspond to the set $\mathrm{Fam}(A)$ of *families* of non-empty configurations indexed by finite sets of natural numbers (see Proposition 7.1.3). As $\mathscr{M}_f(X)$ may be presented as a quotient of finitely indexed families of elements of X, it is natural to define the positions of a $-$-board A as configurations up to symmetry:

Definition 10.4.1. *Consider A a board.*
*The set $\mathscr{Pos}(A)$ of **positions** of A is simply defined as the quotient set $\mathscr{C}(A)/\cong_A$.*

By convention, we use symbols $\mathsf{x}, \mathsf{y}, \mathsf{z}$, etc to range over positions – *i.e.* the same letters as for configurations, but with a different font.

As an example, there are *four* positions on the $-$-board B of Section 8.2.2: the (symmetry classes corrresponding to the) configurations $\emptyset$, $\{\mathbf{q}^-\}$, $\{\mathbf{q}^-, \mathbf{tt}^+\}$, and $\{\mathbf{q}^-, \mathbf{ff}^+\}$. This shows that *positions* do not quite match the relational model just yet, as the relational interpretation of $\mathbb{B}$ only has *two* elements, for the two complete computations.

Stopping positions of boards. Fortunately, boards come equipped with exactly the structure needed to eliminate partial computations: the *payoff function*.

Recall from Definition 8.2.3 that boards come equipped with a *payoff function* $\kappa_A : \mathscr{C}(A) \to \{-1, 0, +1\}$, which was motivated precisely by the need to capture which configurations are complete, and to designate a responsible for non-complete configurations. By condition *invariant*, the payoff function automatically extends to

$$\kappa_A : \mathscr{Pos}(A) \to \{-1, 0, +1\}$$

a payoff function for positions. This lets us define:

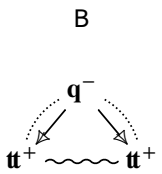


Figure 10.13: Non-deterministic witnesses

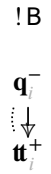


Figure 10.14: Symmetric witnesses

Definition 10.4.2. *Consider A a board. Its set of **stopping positions** is the set*

$$(\!|A|\!) = \{\mathsf{x}_A \in \mathscr{P}os(A) \mid \kappa_A(\mathsf{x}_A) = 0\}$$

This is indeed the right definition, and we shall see in Section 10.4.4 that it makes the various constructions on boards agree with their relational counterparts.

Stopping positions of strategies. To a board A, the relational collapse will associate a set $(\!|A|\!)$. If $\sigma : A$ is a strategy on A, we should assign it a subset of $(\!|A|\!)$. We have seen with Proposition 7.3.1 that causal strategies almost compose relationally already with respect to $+$-covered configurations, so the next definition is almost self-evident:

Definition 10.4.3. *Consider A a board and $\sigma : A$ a causal strategy. Then,*

$$(\!|\sigma|\!) = \{\mathsf{x}_A \in (\!|A|\!) \mid \exists x^\sigma \in \mathscr{C}^+(\sigma),\ \partial_\sigma\, x^\sigma \in \mathsf{x}_A\}$$

*is its **relational collapse**.*

For $x^\sigma \in \mathscr{C}^+(\sigma)$ such that $\partial_\sigma\, x^\sigma \in \mathsf{x}_A \in (\!|A|\!)$, we say that x^σ is a **witness** for x_A; the relational collapse only keeps stopping positions for which there is a witness. In general, the witness is not unique. We show two examples: in Figure 10.13, the (symmetry class of) $\{\mathbf{q}^-, \mathbf{tt}^+\}$ has two witnesses, illustrating the two distinct non-deterministic branches realizing the value $\mathbf{tt}$. But the number of witnesses does not necessarily match the number of non-deterministic branches leading to an observable result: in Figure 10.14, there is an *infinite* number of witnesses for the stopping position corresponding to value $\mathbf{tt}$ – even though all those witnesses are symmetric with each other[3].

10.4.2 Composition and Deadlocks

Now, take A and B two $-$-boards, and let us consider the case of a strategy *from A to B*. Of course the definition above applies, instantiated on the board $A \vdash B$. But in order to get a relation from $(\!|A|\!)$ to $(\!|B|\!)$, we shall make use of the following fact:

Lemma 10.4.4. *Consider A, B two $-$-boards. Then, there is a bijection*

$$s^{\vdash}_{A,B} : (\!|A \vdash B|\!) \simeq (\!|A|\!) \times (\!|B|\!)$$

[3]It is a good intuition that witnesses *should* somehow count non-deterministic branches leading to an observable result, but it requires a much finer definition of witnesses – see Section 13.3.1.

Proof. We know from Lemma 7.1.22 that configurations of $A \vdash B$ all have the form $x_A \vdash x_B$ where $x_A \in \mathscr{C}(A)$ and $x_B \in \mathscr{C}(B)$; this extends to symmetry classes since symmetries enjoy the same unique decomposition. By Definition 8.2.5, $\mathsf{x}_A \vdash \mathsf{x}_B$ has null payoff exactly when both x_A and x_B have (recall that $A \vdash B = A^\perp \parr B$ by definition, below Definition 8.2.5). The bijection follows this decomposition. □

We use this bijection implicitely when defining the collapse of $\boldsymbol{\sigma} : A \vdash B$ as

$$\boldsymbol{(\!|}\boldsymbol{\sigma}\boldsymbol{|\!)} = \{(\mathsf{x}_A, \mathsf{x}_B) \in \boldsymbol{(\!|}A\boldsymbol{|\!)} \times \boldsymbol{(\!|}B\boldsymbol{|\!)} \mid \exists x^{\boldsymbol{\sigma}} \in \mathscr{C}^+(\boldsymbol{\sigma}),\ \partial_{\boldsymbol{\sigma}} x^{\boldsymbol{\sigma}} \in \mathsf{x}_A \vdash \mathsf{x}_B\}$$

a morphism in $\mathbf{Rel}(\boldsymbol{(\!|}A\boldsymbol{|\!)}, \boldsymbol{(\!|}B\boldsymbol{|\!)})$ – now, does this preserve the categorical structure?

Copycat. We start the analysis for copycat, which is straightforward:

Proposition 10.4.5. *Consider A a $-$-board. Then, we have the equality*

$$\boldsymbol{(\!|}\boldsymbol{\mathfrak{a}}_A\boldsymbol{|\!)} = \mathrm{id}_{\boldsymbol{(\!|}A\boldsymbol{|\!)}} \in \mathbf{Rel}(\boldsymbol{(\!|}A\boldsymbol{|\!)}, \boldsymbol{(\!|}A\boldsymbol{|\!)})\,.$$

Proof. $\subseteq$. Consider $\mathsf{x}_A \vdash \mathsf{y}_A \in \boldsymbol{(\!|}\boldsymbol{\mathfrak{a}}_A\boldsymbol{|\!)}$. By definition it has a witness in $\boldsymbol{\mathfrak{a}}_A$, which by Lemma 6.4.4 has form $x_A \parallel x_A \in \mathscr{C}^+(\boldsymbol{\mathfrak{a}}_A)$ with $x_A \vdash x_A \in \mathsf{x}_A \vdash \mathsf{y}_A$, so $\mathsf{x}_A = \mathsf{y}_A$.

$\supseteq$. Consider $(\mathsf{x}_A, \mathsf{x}_A) \in \mathrm{id}_{\boldsymbol{(\!|}A\boldsymbol{|\!)}}$ and take any representative $x_A \in \mathsf{x}_A$. Again by Lemma 6.4.4, $x_A \parallel x_A \in \mathscr{C}^+(\boldsymbol{\mathfrak{a}}_A)$, and it is immediate that it is a witness for $(\mathsf{x}_A, \mathsf{x}_A)$. □

Indeed from Lemma 6.4.4, and with respect to $+$-covered configurations, copycat already acts like the relational identity! In the light of Proposition 7.3.1, we may hope that the situation for composition will just as simple.

Composition: oplax preservation. For one of the two directions, it *is* simple:

Lemma 10.4.6. *Consider $-$-boards A, B, C, and $\boldsymbol{\sigma} : A \vdash B, \boldsymbol{\tau} : B \vdash C$.*
Then, we have $\boldsymbol{(\!|}\boldsymbol{\tau} \odot \boldsymbol{\sigma}\boldsymbol{|\!)} \subseteq \boldsymbol{(\!|}\boldsymbol{\tau}\boldsymbol{|\!)} \circ \boldsymbol{(\!|}\boldsymbol{\sigma}\boldsymbol{|\!)}$.

Proof. Consider $(\mathsf{x}_A, \mathsf{x}_C) \in \boldsymbol{(\!|}\boldsymbol{\tau} \odot \boldsymbol{\sigma}\boldsymbol{|\!)}$. By definition – and inlining Proposition 7.3.1 – there is $x^{\boldsymbol{\tau}} \odot x^{\boldsymbol{\sigma}} \in \mathscr{C}^+(\boldsymbol{\tau} \odot \boldsymbol{\sigma})$ s.t. $\partial_{\boldsymbol{\tau}\odot\boldsymbol{\sigma}}(x^{\boldsymbol{\tau}} \odot x^{\boldsymbol{\sigma}}) = x^{\boldsymbol{\sigma}}_A \vdash x^{\boldsymbol{\tau}}_C \in \mathsf{x}_A \vdash \mathsf{x}_C$. Thus

$$\partial_{\boldsymbol{\sigma}}\, x^{\boldsymbol{\sigma}} = x^{\boldsymbol{\sigma}}_A \vdash x^{\boldsymbol{\sigma}}_B \in \mathsf{x}_A \vdash \mathsf{x}_B\,, \qquad \partial_{\boldsymbol{\tau}}\, x^{\boldsymbol{\tau}} = x^{\boldsymbol{\tau}}_B \vdash x^{\boldsymbol{\tau}}_C \in \mathsf{x}_B \vdash \mathsf{x}_C$$

where $x^{\boldsymbol{\sigma}} \in \mathscr{C}^+(\boldsymbol{\sigma})$ and $x^{\boldsymbol{\tau}} \in \mathscr{C}^+(\boldsymbol{\tau})$ and for x_B the symmetry class of $x^{\boldsymbol{\sigma}}_B = x^{\boldsymbol{\tau}}_B = x_B$.

The only thing we must check is that $\mathsf{x}_B \in \boldsymbol{(\!|}B\boldsymbol{|\!)}$, *i.e.* $\kappa_B(x_B) = 0$. But if we had $\kappa_B(x_B) = -1$, we would have $\kappa_{A\vdash B}(x^{\boldsymbol{\sigma}}_A \vdash x_B) = -1$, contradicting that $\boldsymbol{\sigma}$ is winning. Likewise, $\kappa_B(x_B) = 1$ would contradict that $\boldsymbol{\tau}$ is winning. Hence, $\kappa_B(x_B) = 0$. □

This is not by chance: the mechanism of payoff and winning strategies has been set up specifically with this collapse in mind. Now, examine the other direction.

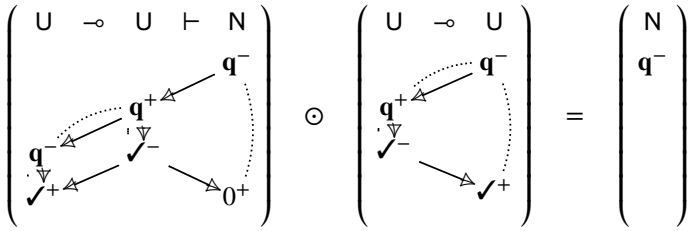


Figure 10.15: Deadlocking composition of causal strategies

Composition: non lax preservation. In the other direction, consider $\sigma : A \vdash B, \tau : B \vdash C$, and $(\mathsf{x}_A, \mathsf{x}_C) \in (\!|\tau|\!) \circ (\!|\sigma|\!)$. This means that there is $\mathsf{x}_B \in (\!|B|\!)$ a stopping position such that $(\mathsf{x}_A, \mathsf{x}_B) \in (\!|\sigma|\!)$ and $(\mathsf{x}_B, \mathsf{x}_C) \in (\!|\tau|\!)$; meaning that there are two witnesses

$$x^\sigma \in \mathscr{C}^+(\sigma), \qquad x^\tau \in \mathscr{C}^+(\tau)$$

such that $x^\sigma_A \in \mathsf{x}_A, x^\tau_C \in \mathsf{x}_C$ and $x^\sigma_B, x^\tau_B \in \mathsf{x}_B$ – thus there is $\theta_B : x^\sigma_B \cong_B x^\tau_B$. So x^σ and x^τ may not match on B on the nose, but only up to symmetry. This is an issue, but one that we are prepared to handle just as we already did in the proof of Lemma 9.3.5, using Proposition 7.4.4, reindexing both x^σ and x^τ to find a true synchronization. Thus let us ignore that for now, and pretend that we are lucky enough to have $x^\sigma_B = x^\tau_B$.

To sum up, relational composition provides us with a *witness position* that both strategies agree (up to symmetry) is *reachable*. In contrast, composition of strategies is more rigid: not only should the projections of x^σ and x^τ on B match, they should also arrive at this position *in a compatible chronological ordering*. This is not always possible: these two notions of composition differ when interaction triggers a *causal deadlock*, *i.e.* pairs of configurations that are *matching* but not *secured* as in Definition 6.2.2. Figure 10.15 displays an example: the strategy obtained by composition has no response to the initial Opponent move, while relational composition authorizes 0^+.

This strikes at the heart of the difference between game and relational semantics: the former is dynamic hence sensitive to deadlocks, while the latter is static. This of course is what lets game semantics model languages with non-commutative effects, but for us, very concretely, it means that the relational model is not lax functorial.

10.4.3 The Deadlock-Free Lemma

Our *deus ex machina* is visibility. A powerful – and at first unexpected – consequence of visibility is that any interaction between visible strategies is always deadlock-free. The consequence of visibility that our proof will exploit repeatedly is:

Lemma 10.4.7. *Take A, B $-$-boards, $\sigma : A \vdash B$ visible, and $m, m' \in \sigma$ s.t. $m <_\sigma m'$. Then, $\mathsf{j}(m')$ is comparable with m with respect to $\leq_\sigma$.*

Proof. Since $m <_\sigma m'$, there is $\rho \rightarrowtriangle m' \in \mathsf{gcc}(\sigma)$ s.t. $m \in \rho$. If $\partial_\sigma(m')$ is minimal in A, $\partial_\sigma(\mathsf{j}(m'))$ is minimal in B, so $\mathsf{j}(m')$ is minimal for $\leq_\sigma$ by courtesy. But since σ is

pointed, $\mathrm{j}(m')$ is initial in ρ, obviously comparable with m as ρ is totally ordered.
Else, by visibility $\mathrm{j}(m') \in \rho$. But ρ is totally ordered, so $m, \mathrm{j}(m')$ comparable. □

We shall prove that the composition of visible causal strategies is deadlock-free. But first, we recall the basic mechanisms of interactions between causal strategies. Consider $\sigma : A \vdash B$ and $\tau : B \vdash C$, and configurations $x^\sigma \in \mathscr{C}(\sigma)$, and $x^\tau \in \mathscr{C}(\tau)$ s.t., writing $\partial_\sigma x^\sigma = x^\sigma_A \parallel x^\sigma_B$ and $\partial_\tau x^\tau = x^\tau_B \parallel x^\tau_C$, we have $x^\sigma_B = x^\tau_B = x_B$, *i.e.* x^σ and x^τ are *matching*. Recall from Definition 6.2.2 the bijection arising from their synchronization:

$$\varphi \quad : \quad x^\sigma \parallel x^\tau_C \quad \overset{\partial_\sigma \parallel x^\tau_C}{\simeq} \quad x^\sigma_A \parallel x_B \parallel x^\tau_C \quad \overset{x^\sigma_A \parallel \partial_\tau^{-1}}{\simeq} \quad x^\sigma_A \parallel x^\tau \,,$$

whose graph is equipped with a relation importing the causal constraints from σ and τ:

$$(l, r) \lhd (l', r') \qquad \Leftrightarrow \qquad l <_{\sigma \parallel C} l' \quad \vee \quad r <_{A \parallel \tau} r' \,.$$

We saw in Definition 6.2.2 and Proposition 7.3.6 that (x^σ, x^τ) corresponds to a configuration of the interaction $\tau \circledast \sigma$ when this bijection is *secured*, *i.e.* $\lhd$ is acyclic.

If $\sigma : A \vdash B$ and $\tau : B \vdash C$ are visible, we claim that this is always the case.

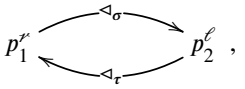
Proof sketch. Before giving the formal proof, we showcase the reasoning on a simplified case. The basic idea is by contradiction: starting with a putative deadlock, we repeatedly push it down the causal dependency of the board, until it reaches a minimal event – but those cannot appear in a cycle.

Consider a **simple deadlock** in φ, given by $p_1 = (l_1, r_1)$ and $p_2 = (l_2, r_2) \in \varphi$ s.t.

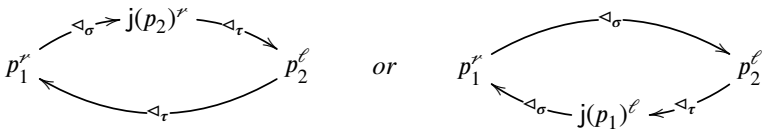
$$l_1 <_{\sigma \parallel C} l_2 \,, \qquad r_2 <_{A \parallel \tau} r_1 \,,$$

an immediate incompatibility between p_1 and p_2. In other words we have $p_1 \lhd p_2$ and $p_2 \lhd p_1$; we use $p_1 \lhd_\sigma p_2$ and $p_2 \lhd_\tau p_1$ to indicate the origin of the causal constraint. We apply the same conventions for polarity of elements of φ as for Lemma 6.2.15.

The first observation (skipped here) is that *w.l.o.g.*, the polarities are as in

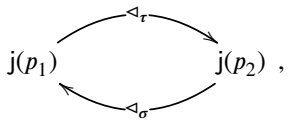


where both occur in B but not minimal in B – so we may take $\mathrm{j}(p_i) = (\mathrm{j}(l_i), \mathrm{j}(r_i))$. By Lemma 10.4.7, l_1 and $\mathrm{j}(l_2)$ are comparable for σ; while r_2 and $\mathrm{j}(r_1)$ are comparable for τ. If $p_1 \lhd_\sigma \mathrm{j}(p_2)$ or $p_2 \lhd_\tau \mathrm{j}(p_1)$, then we respectively have one of the cycles:



so simple deadlocks between p_1 and $\mathsf{j}(p_2)$; or between $\mathsf{j}(p_1)$ and p_2. Those simple deadlocks are smaller, in the sense that their cumulative *depth* in B has decreased – justifying the inductive reasoning. The case $p_1 = \mathsf{j}(p_2)$ or $p_2 = \mathsf{j}(p_1)$ is easily discarded.

The last case has $\mathsf{j}(p_2) \lhd_\sigma p_1$ and $\mathsf{j}(p_1) \lhd_\tau p_2$. But p_1 has polarity $\nearrow$, so by Lemma 10.2.2 the only immediate dependency in σ of l_1^- is $\mathsf{j}(l_1)$. So $\mathsf{j}(p_2) \lhd_\sigma p_1$ factors as $\mathsf{j}(p_2) \lhd_\sigma \mathsf{j}(p_1) \lhd_\sigma p_1$. Symmetrically $\mathsf{j}(p_1) \lhd_\tau \mathsf{j}(p_2)$, so we have:



closer to the root of the board. Repeating this we eventually hit an impossible simple deadlock with a minimal event in B, finally exposing the contradiction.

Detailed proof. The proof of the deadlock-free lemma is the same in essence, but challenging in form. Firstly, cycles in $\lhd$ in Definition 6.2.2 may have arbitrary length. Secondly, in relational composition strategies synchronize on *symmetry classes* rather than configurations; so we must account for synchronization through symmetry.

Lemma 10.4.8. *Consider A, B, C −-boards, $\sigma : A \vdash B$ and $\tau : B \vdash C$ visible causal strategies, $x^\sigma \in \mathscr{C}(\sigma)$ and $x^\tau \in \mathscr{C}(\tau)$ with a symmetry $\theta : x^\sigma_B \cong_B x^\tau_B$.*

Then, the composite bijection

$$\varphi \;:\; x^\sigma \parallel x^\tau_C \;\overset{\partial_\sigma \parallel x^\tau_C}{\simeq}\; x^\sigma_A \parallel x^\sigma_B \parallel x^\tau_C \;\overset{x^\sigma_A \parallel \theta \parallel x^\tau_C}{\simeq}\; x^\sigma_A \parallel x^\tau_B \parallel x^\tau_C \;\overset{x^\sigma_A \parallel \partial_\tau^{-1}}{\simeq}\; x^\sigma_A \parallel x^\tau\,,$$

is secured, in the sense that the relation $\lhd$, defined on the graph of φ with

$$(l, r) \lhd (l', r')$$

whenever l $(<_\sigma \parallel <_C)$ l' or r $(<_A \parallel <_\tau)$ r', is acyclic; i.e. *$(x^\sigma, \theta, x^\tau)$ causally compatible.*

Proof. We use *polarities* $\ell, \nearrow$ or $-$ for elements of φ (*i.e.* pairs (l, r)) as above. We say (l, r) *occurs in A, B or C* in the obvious sense. We use a notion of *justifier* of a pair (l, r) non-minimal in B: as θ is an order-iso, $(\partial_\sigma \parallel C)(l)$ is minimal in B iff $(A \parallel \partial_\tau)(r)$ is. If not, then $\mathsf{j}(l)$ and $\mathsf{j}(r)$ also match up to θ and $(\mathsf{j}(l), \mathsf{j}(r))$ must be in φ as well – we write it $\mathsf{j}(l, r)$. Suppose now $\lhd$ is *not* secured, *i.e.* there is $((l_1, r_1), \ldots, (l_n, r_n))$ with

$$(l_1, r_1) \lhd (l_2, r_2) \lhd \ldots \lhd (l_n, r_n) \lhd (l_1, r_1)\,,$$

written $p_1 \lhd \cdots \lhd p_n \lhd p_1$ – the **length** of this cycle is n. First, *w.l.o.g.* the cycle occurs entirely in B. Assume it has minimal length. If it occurs entirely in A or C, then $(l_i)_{1 \le i \le n}$ (resp. $(r_i)_{1 \le i \le n}$) is a cycle in σ (resp. τ), absurd. So, it passes through B. Next, if *e.g.*

$$p_i^{(B)} \lhd p_{i+1}^{(C)} \lhd \cdots \lhd p_{j-1}^{(C)} \lhd p_j^{(B)}\,,$$

then it is easy to prove that $r_i < r_{i+1} < \cdots < r_{j-1} < r_j$, so that $p_i^{(B)} \lhd p_j^{(B)}$ and the cycle can be shortened, contradicting its minimality – the same argument holds for A.

We restrict to cycles in B. The **depth** of (l, r) is the length of the chain of justifiers to (l_0, r_0) minimal in B – the depth of (l_0, r_0) minimal in B is 0. The cycle has *depth*

$$d = \sum_{1 \leq i \leq n} \mathrm{depth}(l_i, r_i)\,,$$

and we assume *w.l.o.g.* the cycle minimal for the product order on pairs (n, d). In this proof, all arithmetic computations on indices are done modulo n (the length).

Next, we write $p_i \lhd_\sigma p_j$ if $l_i \; (<_\sigma \| <_C) \; l_j$ and $p_i \lhd_\tau p_j$ symmetrically. We notice that $\lhd_\sigma$ and $\lhd_\tau$ alternate – if not we shorten the cycle by transitivity. We assume *w.l.o.g.* that $p_{2k} \lhd_\sigma p_{2k+1}$ and $p_{2k+1} \lhd_\tau p_{2k+2}$ for all k. But then, $\mathrm{pol}(p_{2k}) = \nearrow$ and $\mathrm{pol}(p_{2k+1}) = \swarrow$ so that polarity in the cycle is alternating as well. Indeed, assume *e.g.*

$$p_{2k+1} \lhd p^{\swarrow}_{2k+2} \lhd p_{2k+3}$$

with $p_{2k+1} \lhd_\tau p_{2k+2}$ and $p_{2k+2} \lhd_\sigma p_{2k+3}$. Then, $r_{2k+1} <_{A\|\tau} r^-_{2k+2}$. Negative r^-_{2k+2} cannot be minimal in B, so by Lemma 10.2.2, it has a unique predecessor $\mathrm{j}(r_{2k+2}) \rightarrowtriangle_{A\|\tau} r_{2k+2}$, so $r_{2k+1} <_{A\|\tau} r^-_{2k+2}$ factors as $r_{2k+1} <_{A\|\tau} \mathrm{j}(r_{2k+2}) \rightarrowtriangle_{A\|\tau} r^-_{2k+2}$. Accordingly, $p_{2k+1} \lhd_\tau \mathrm{j}(p_{2k+2}) \lhd_\tau p_{2k+2}$ – but dependencies in the game are respected by both strategies, so $\mathrm{j}(p_{2k+2}) \lhd_\sigma p_{2k+2}$. So $\mathrm{j}(p_{2k+2}) \lhd_\sigma p_{2k+3}$, and we can replace the cycle fragment with

$$p_{2k+1} \lhd \mathrm{j}(p_{2k+2}) \lhd p_{2k+3}$$

which is still in B, has the same length but strictly smaller depth. The symmetric argument applies for σ, so any p_{2k+1} has polarity $\swarrow$ and any p_{2k+2} has polarity $\nearrow$.

We show the cycle cannot have an event minimal in B. Seeking a contradiction, if

$$p^{\swarrow}_{2k+1} \lhd_\tau p^{\nearrow}_{2k+2} \lhd_\sigma p^{\swarrow}_{2k+3}$$

with p_{2k+2} minimal in B, then $l_{2k+2} <_{\sigma\|C} l_{2k+3}$ with l_{2k+2} minimal in B, but then $\partial_{\sigma\|C}(l_{2k+2}) <_{A\|B\|C} \partial_{\sigma\|C}(l_{2k+3})$. Indeed, if $\partial_{\sigma\|C}(l_{2k+2})$ is minimal in B, l_{2k+2} is (by courtesy) minimal in $\sigma \parallel C$. Likewise, since l_{2k+3} occurs in B, $\partial_{\sigma\|C}(l_{2k+3})$ depends (for $\leq_{A\|B\|C}$) on a unique $\partial_{\sigma\|C}(l)$ minimal in B, where l must also be minimal in σ. But since σ is pointed, l_{2k+3} has a unique minimal dependency, hence $l = l_{2k+2}$ and $\partial_{\sigma\|C}(l_{2k+2}) <_{A\|B\|C} \partial_{\sigma\|C}(l_{2k+3})$ as claimed. But then, $r_{2k+2} <_{A\|\tau} r_{2k+3}$, so $p^{\swarrow}_{2k+1} \lhd_\tau p^{\nearrow}_{2k+2} \lhd_\tau p^{\swarrow}_{2k+3}$ and again the cycle can be shortened by transitivity.

We have proved a minimal cycle has a canonical form where the strategies alternate, polarity alternates, all events are in B and non-minimal. Since $p^{\nearrow}_{2k} \lhd_\sigma p^{\swarrow}_{2k+1}$, writing $p = (l, r) = \mathrm{j}(p_{2k+1})$, $l = \mathrm{j}(l_{2k+1})$ as well. From Lemma 10.4.7, we know that $l = \mathrm{j}(l_{2k+1})$ is comparable with l_{2k} in $\sigma \parallel C$ (by visibility of σ). If $\mathrm{j}(l_{2k+1}) = l_{2k}$, then $r_{2k} \lhd_\tau r_{2k+1}$ as well. This gives $p_{2k-1} \lhd_\tau p_{2k+2}$, contradicting minimality of the cycle. So $\mathrm{j}(l_{2k+1}) \neq l_{2k}$. Similarly, $\mathrm{j}(r_{2k+2})$ is comparable with r_{2k+1} in $A \parallel \tau$, but distinct.

Assume that we have $p_{2k} \lhd_\sigma \mathrm{j}(p_{2k+1})$ for some k. Since $\mathrm{j}(p_{2k+1}) \lhd_\tau p_{2k+1} \lhd_\tau p_{2k+2}$ we can replace the cycle fragment $p_{2k} \lhd p_{2k+1} \lhd p_{2k+2}$ with the cycle fragment

$$p_{2k} \lhd \mathrm{j}(p_{2k+1}) \lhd p_{2k+2}$$

which has the same length but smaller depth, absurd. So, $\mathrm{j}(p_{2k+1}) \lhd_\sigma p_{2k}$ for all k (symmetrically, $\mathrm{j}(p_{2k+2}) \lhd_\tau p_{2k+1}$ for all k). In particular, $\mathrm{j}(l_{2k+1}) <_{\sigma\|C} l^-_{2k}$ but by

Lemma 10.2.2, l^-_{2k} has a unique immediate predecessor $\mathsf{j}(l_{2k})$. So, $\mathsf{j}(p_{2k+1}) \lhd_\sigma \mathsf{j}(p_{2k})$ for all k; and likewise $\mathsf{j}(p_{2k+2}) \lhd_\tau \mathsf{j}(p_{2k+1})$ for all k. So we can replace the full cycle with

$$\mathsf{j}(p_n) \lhd \mathsf{j}(p_{n-1}) \lhd \ldots \lhd \mathsf{j}(p_1) \lhd \mathsf{j}(p_n)$$

which has the same length but smaller depth, contradiction. □

The deadlock-free lemma is a powerful observation with far-reaching consequences in linking game semantics and relational models. It also gives a lot of weight to the notion of visibility, as a simple, well-behaved under-approximation of innocence.

As a direct consequence of the deadlock-free lemma, we obtain:

Corollary 10.4.9. *Consider $-$-boards A, B, C, and $\sigma : A \vdash B, \tau : B \vdash C$ visible. Then, $(\!|\tau \odot \sigma|\!) = (\!|\tau|\!) \circ (\!|\sigma|\!)$.*

Proof. $\subseteq$. Follows from Lemma 10.4.6.

$\supseteq$. Consider $(\mathsf{x}_A, \mathsf{x}_B) \in (\!|\sigma|\!)$ and $(\mathsf{x}_B, \mathsf{x}_C) \in (\!|\tau|\!)$, with witnesses $x^\sigma \in \mathscr{C}^+(\sigma)$ and $x^\tau \in \mathscr{C}^+(\tau)$ with $\partial_\sigma(x^\sigma) = x^\sigma_A \parallel x^\sigma_B$, $\partial_\tau(x^\tau) = x^\tau_B \parallel x^\tau_C$, with $x^\sigma_A \in \mathsf{x}_A$, $x^\sigma_B, x^\tau_B \in \mathsf{x}_B$, and $x^\tau_C \in \mathsf{x}_C$. In particular, there is $\theta : x^\sigma_B \cong_B x^\tau_B$. Now, the composite bijection

$$\varphi \;:\; x^\sigma \parallel x^\tau_C \overset{\partial_\sigma \parallel x^\tau_C}{\simeq} x^\sigma_A \parallel x^\sigma_B \parallel x^\tau_C \overset{x^\sigma_A \parallel \theta \parallel x^\tau_C}{\simeq} x^\sigma_A \parallel x^\tau_B \parallel x^\tau_C \overset{x^\sigma_A \parallel \partial^{-1}_\tau}{\simeq} x^\sigma_A \parallel x^\tau\,,$$

is secured by Lemma 10.4.8. Thus by Proposition 7.4.4, there are reindexings $y^\sigma \cong_\sigma x^\sigma$ and $y^\tau \cong_\tau x^\tau$ such that $\partial_\sigma(y^\sigma) = y^\sigma_A \parallel y^\sigma_B$, $\partial_\tau(y^\tau) = y^\tau_B \parallel y^\tau_C$ with $y^\sigma_B = y^\tau_B$ – so $y^\tau \odot y^\sigma \in \mathscr{C}(\tau \odot \sigma)$. But since $y^\sigma \cong_\sigma x^\sigma$ and $y^\tau \cong_\tau x^\tau$, we also know that $y^\sigma_A \in \mathsf{x}_A$ and $y^\tau_C \in \mathsf{x}_C$ still, so $y^\tau \odot y^\sigma$ witnesses $(\mathsf{x}_A, \mathsf{x}_C) \in (\!|\tau \odot \sigma|\!)$ as required. □

10.4.4 A Relative Seely $\sim$-Functor

Next, we show that the relational collapse preserves the rest of the structure – it should form a *relative Seely $\sim$-functor*. Relative Seely ($\sim$-)functors are simply ($\sim$-)functors equipped with isomorphisms expressing the preservation of the additional structure, satisfying certain coherence conditions. Just like relative Seely categories, relative Seely ($\sim$-)functors are not a standard notion. However, they are certainly an unsurprising one, so in this monograph we opted to relegate it to the appendix (Appendix A.2).

To construct our concrete relative Seely $\sim$-functor, we start with:

Proposition 10.4.10. *We have a $\sim$-functor $(\!|-|\!) : \mathbf{NTCG\text{-}Vis} \to \mathbf{Rel}$.*

Proof. Preservation of copycat is by Proposition 10.4.5, and preservation of composition is by Proposition 10.4.9. It remains to show that $(\!|-|\!)$ preserves $\approx$.

Hence, consider $\sigma, \sigma' : A \vdash B$ with $f : \sigma \approx \sigma'$, and take $(\mathsf{x}_A, \mathsf{x}_B) \in (\!|\sigma|\!)$. By definition, there is a witness $x^\sigma \in \mathscr{C}^+(\sigma)$ such that $\partial_\sigma\, x^\sigma \in \mathsf{x}_A \vdash \mathsf{x}_B$. But then,

$$f[x^\sigma]_A \vdash f[x^\sigma]_B \;:\; x^\sigma_A \vdash x^\sigma_B \;\cong_{A \vdash B}\; f(x^\sigma)_A \vdash f(x^\sigma)_B$$

so that $f(x^\sigma) \in \mathsf{x}_A \vdash \mathsf{x}_B$ as required. The other inclusion holds by symmetry. □

Next, we examine preservation of the full categorical structure.

Preservation of the monoidal structure. Following Definition A.2.1, we require

$$\begin{array}{rcccc} s^{\otimes}_{A,B} & : & (\!|A|\!) \times (\!|B|\!) & \cong & (\!|A \otimes B|\!) \\ s^{1} & : & 1 & \cong & (\!|1|\!) \end{array}$$

isomorphisms in **Rel**, natural in A and B – those follow from the bijections of Lemma 7.1.22 in the obvious way, which is indeed compatible with payoff.

Proposition 10.4.11. *We have* $((\!|-|\!), s^{\otimes}, s^{1})$ *a symmetric monoidal* $\sim$*-functor.*

Proof. All verifications are straightforward. For naturality, we must show that

$$\begin{array}{ccc} (\!|A|\!) \times (\!|B|\!) & \xrightarrow{s^{\otimes}_{A,B}} & (\!|A \otimes B|\!) \\ \downarrow{\scriptstyle (\!|\sigma|\!)\times(\!|\tau|\!)} & & \downarrow{\scriptstyle (\!|\sigma\otimes\tau|\!)} \\ (\!|A'|\!) \times (\!|B'|\!) & \xrightarrow[s^{\otimes}_{A',B'}]{} & (\!|A' \otimes B'|\!) \end{array}$$

commutes for $\sigma : A \vdash A'$ and $\tau : B \vdash B'$, which is direct by Proposition 8.1.1. □

Preservation of !. Next we deal with the exponential structure, which is the most challenging. Inspecting the requirements for relative Seely functors we must first show that $(\!|-|\!)$ preserves strict objects – but this is trivial: as a proper Seely category, **Rel** is regarded as a relative Seely category with all objects strict. Then we must provide

$$s^{!}_{S} : \mathscr{M}_f((\!|S|\!)) \cong (\!|!S|\!)$$

for every strict S, subject to coherence conditions. We first construct the bijection:

Proposition 10.4.12. *For any* S *strict, there is a bijection* $s^{!}_{S} : \mathscr{M}_f((\!|S|\!)) \cong (\!|!S|\!)$.

Proof. First, recall from Lemma 8.3.1 that we have a bijection

$$\mathrm{Fam}(\mathscr{C}^{\neq\emptyset}(S)) \simeq \mathscr{C}(!S)$$

following which configurations of null payoff of $!S$ correspond to families of non-empty configurations of null payoff of S. But S is strict, thus its configurations of null payoff are non-empty. Hence the bijection above may be directly refined into another bijection

$$\mathrm{Fam}(\llparenthesis S \rrparenthesis) \simeq \llparenthesis !S \rrparenthesis$$

where $\llparenthesis A \rrparenthesis$ is the set of configurations of null payoff. Inlining this bijection, we may define the inverse of $s^{!}_{S}$ on representatives directly as follows:

$$\begin{array}{rcccc} (s^{!}_{S})^{-1} & : & \llparenthesis !S \rrparenthesis & \to & \mathscr{M}_f((\!|S|\!)) \\ & & [x^{S}_{i} \mid i \in I] & \mapsto & [\mathbf{x}^{S}_{i} \mid i \in I] \end{array}$$

where $\mathbf{x}^{i}_{S}$ denotes the symmetry class of x^{S}_{i}, $[\mathbf{x}^{S}_{i} \mid i \in I]$ (with a bold font) is our notation for families, and $[\mathbf{x}^{S}_{i} \mid i \in I]$ (with non-bold font) is multiset comprehension.

Invariant. Consider $\theta : [x_i^S \mid i \in I] \cong_{!S} [y_j^S \mid j \in J]$ a symmetry, which by definition is given by a bijection $\pi : I \simeq J$ and a family $(\theta_i)_{i\in I}$ where $\theta_i : x_i^S \cong_S y_{\pi(i)}^S$ for all $i \in I$. But then, $\mathsf{x}_i^S = \mathsf{y}_{\pi(i)}^S$, from which, as required, we deduce that

$$[\mathsf{x}_i^S \mid i \in I] = [\mathsf{y}_{\pi(i)}^S \mid i \in I] = [\mathsf{y}_j^S \mid j \in J]\,.$$

Injective. Assume $[\mathsf{x}_i^S \mid i \in I] = [\mathsf{y}_j^S \mid j \in J]$. Thus, there is a bijection $\pi : I \simeq J$ such that for all $i \in I$, $\mathsf{x}_i^S = \mathsf{y}_{\pi(i)}^S$. Hence, for all $i \in I$ there is $\theta_i : x_i^S \cong_S y_{\pi(i)}^S$, altogether forming a symmetry $\theta : [x_i^S \mid i \in I] \cong_S [y_j^S \mid j \in J]$.

Surjective. Consider a multiset $[\mathsf{x}_1, \dots, \mathsf{x}_n] \in \mathscr{M}_f(\llparenthesis S \rrparenthesis)$. For each $1 \leq i \leq n$, consider a representative $x_i \in \mathsf{x}_i$. Then, $[x_i \mid 1 \leq i \leq n]$ provides a pre-image. □

Observe that there is no such bijection $\mathscr{M}_f(\llparenthesis A \rrparenthesis) \simeq \llparenthesis !A \rrparenthesis$ for A non-strict, because

$$[\emptyset] \neq [\emptyset, \emptyset]$$

are distinct elements of $\mathscr{M}_f(\llparenthesis A \rrparenthesis)$, but can only be represented in $\llparenthesis !A \rrparenthesis$ with the same (empty) configuration since there are no events. The relational model may remember how many times a game is *not* visited, which makes no sense in terms of games. This is the reason behind the preservation of the *relative* Seely ∼-categorical structure only.

Among the several diagrams to check, the most subtle is preservation of promotion:

Lemma 10.4.13. *Consider S, T strict and $\sigma : !S \vdash T$.*
Then, the following square commutes in **Rel***:*

$$\begin{array}{ccc} \mathscr{M}_f(\llparenthesis S \rrparenthesis) & \xrightarrow{(\llparenthesis \sigma \rrparenthesis \circ s_S^!)^!} & \mathscr{M}_f(\llparenthesis T \rrparenthesis) \\ {\scriptstyle s_S^!}\downarrow & & \downarrow{\scriptstyle s_T^!} \\ \llparenthesis !S \rrparenthesis & \xrightarrow[\llparenthesis \sigma^! \rrparenthesis]{} & \llparenthesis !T \rrparenthesis \end{array}$$

Proof. We use a multiset notation for stopping positions of $!S$ and $!T$, inlining Proposition 10.4.12. The upper-right path computes to the pairs that can be written as

$$([\mathsf{x}_{i,j} \mid i \in I, j \in J_i],\ \ [\mathsf{y}_i \mid i \in I])$$

for $\mathsf{x}_{i,j} \in \llparenthesis S \rrparenthesis$ and $\mathsf{y}_i \in \llparenthesis T \rrparenthesis$ where we have that for all $i \in I$, $([\mathsf{x}_{i,j} \mid j \in J_i], \mathsf{y}_i) \in \llparenthesis \sigma \rrparenthesis$. Thus for all $i \in I$, there is $x_i^\sigma \in \mathscr{C}^+(\sigma)$ witnessing that pair, *i.e.* we have

$$\partial_\sigma x_i^\sigma \ \ \in \ \ [\mathsf{x}_{i,j} \mid j \in J_i] \vdash \mathsf{y}_i\,.$$

Thus, relying on Proposition 8.3.3 we may form $[x_i^\sigma \mid i \in I] \in \mathscr{C}^+(\sigma^!)$ (by Lemma 8.1.8, $\sigma^!$ only differs from $!\sigma$ via its display map), and a direct verification ensures that

$$\partial_{\sigma^!}[x_i^\sigma \mid i \in I] \ \ \in \ \ ([\mathsf{x}_{i,j} \mid i \in I, j \in J_i], [\mathsf{y}_i \mid i \in I])$$

as required. The other direction follows the same decompositions. □

Relative Seely $\sim$-functor. To wrap up, we introduce the missing components. First,

$$s^{\top} : \emptyset \cong (\!|\top|\!)$$

is simply the empty relation. For the product and the arrow, we prove:

Lemma 10.4.14. *Consider A, S, T $-$-boards with S, T strict. Then, we have bijections*

$$\begin{array}{rcccc} s^{\&}_{S,T} & : & (\!|S|\!) + (\!|T|\!) & \cong & (\!|S \& T|\!) \\ s^{\multimap}_{A,S} & : & (\!|A|\!) \times (\!|S|\!) & \cong & (\!|A \multimap S|\!) \end{array}$$

Proof. For the product, recall first that by Lemma 8.2.9, we have a bijection

$$\mathscr{C}^{\neq\emptyset}(S) + \mathscr{C}^{\neq\emptyset}(T) \simeq \mathscr{C}^{\neq\emptyset}(S \& T)$$

which holds also for symmetries, and thus extends to symmetry classes. But now we note that S, T, and $S \& T$ are strict, therefore configurations of null payoff must be non-empty. It follows that the bijection above extends to stopping positions.

For the arrow, similarly recall that by Lemma 8.2.12, we have a bijection

$$\mathscr{C}^{\neq\emptyset}(A \multimap S) \simeq \mathscr{C}(A) \times \mathscr{C}^{\neq\emptyset}(S)$$

which also holds to symmetry and thus extends to symmetry classes. But again note that since S is strict, configurations with null payoff must be non-empty, and thus the above extends to a bijection on stopping positions. □

The missing five coherence diagrams of relative Seely $\sim$-functors are direct, from an analysis of the symmetry classes reached by the component strategies. Projections, dereliction and monoidality are defined by lifting, hence the corresponding diagrams follow as for copycat in Proposition 10.4.5. For evaluation, this follows from the description of its $+$-covered configurations in Proposition 8.2.27. Altogether:

Corollary 10.4.15. *We have a relative Seely $\sim$-functor* $(\!|-|\!) : \mathbf{NTCG\text{-}Vis} \to \mathbf{Rel}$.

By Theorem A.2.3 we have a cartesian closed $\sim$-functor $(\!|-|\!) : \mathbf{NTCG\text{-}Vis}_! \to \mathbf{Rel}_!$.

Once the deadlock-free lemma is established, this interpretation-preserving collapse from concurrent games to the relational model is strikingly simple, and emphasizes how the concurrent games model may be regarded as a causal and temporal extension of the relational model. This extension is of course far from negligible as it allows one to represent programming languages with non-commutative effects, such as references.

Positional equivalence. In this monograph, we shall exploit this concretely by reasoning on strategies up to the induced equivalence relation:

Definition 10.4.16. *Consider A, B $-$-boards, and $\sigma, \sigma' \in \mathbf{NTCG\text{-}Vis}(A, B)$.*

*We say that σ and σ' are **positionally equivalent**, written $\sigma \equiv \sigma'$, iff $(\!|\sigma|\!) = (\!|\sigma'|\!)$.*

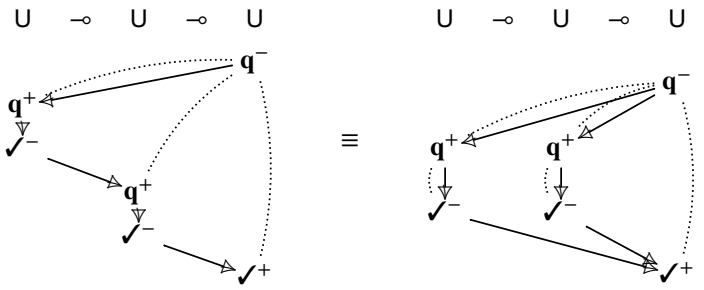


Figure 10.16: Quotienting out evaluation order

$$\frac{\mathsf{x}_A \in (\!|A|\!) \qquad \mathsf{x}_B \in (\!|B|\!)}{\mathsf{x}_A \otimes \mathsf{x}_B \in (\!|A \otimes B|\!)} \qquad \frac{\mathsf{x}_A \in (\!|A|\!) \qquad \mathsf{x}_C \in (\!|C|\!)}{\mathsf{x}_A \multimap \mathsf{x}_C \in (\!|A \multimap C|\!)}$$

$$\frac{\mathsf{x} \in (\!|A|\!)_i \qquad (i \in I)}{(i, \mathsf{x}) \in (\!|\&_{i \in I} A_i|\!)} \qquad \frac{(\mathsf{x}^i_C \in (\!|C|\!))_{i \in I}}{[\mathsf{x}^i_C \mid i \in I] \in (\!|!C|\!)} \qquad \frac{\mathsf{x}_A \in (\!|A|\!) \qquad \mathsf{x}_B \in (\!|B|\!)}{\mathsf{x}_A \vdash \mathsf{x}_B \in (\!|A \vdash B|\!)}$$

Figure 10.17: Syntax for positions for A, B, C $-$-boards with C strict

Concretely, positional equivalence amounts to quotienting out information about the evaluation order, as illustrated in Figure 10.16. It is a very brutal quotient, which would clearly be inconsistent in the presence of any non-commutative effect. But since $(\!|-|\!)$ is a relative Seely $\sim$-functor, positional equivalence *is* a congruence within visible causal strategies: it is preserved by all operations used in computing the interpretation. In other words, there is another relative Seely $\sim$-category where *visible* strategies are considered up to positional equivalence – we shall not move to this $\sim$-category, but the finite definability result of Chapter 12 will hold up to positional equivalence only.

Back to $\rightarrowtriangle$-Strat. The developments presented in this chapter have no interaction with mixed boards and well-bracketing, hence transport transparently from $\mathbf{NTCG}_!$ to $\rightarrowtriangle$-**Strat**. Hence there are $\rightarrowtriangle$-**Vis** and $\rightarrowtriangle$-**Inn**, two cartesian closed sub-$\sim$-categories of $\rightarrowtriangle$-**Strat** with morphisms respectively restricted to *visible* and *parallel innocent* strategies. They both support the interpretation of $\mathsf{PCF}_{/\!/}$, with cartesian closed $\sim$-functors:

$$\begin{array}{rcccl} (\!|-|\!) & : & \rightarrowtriangle\text{-}\mathbf{Vis} & \to & \mathbf{Rel}_! \\ (\!|-|\!) & : & \rightarrowtriangle\text{-}\mathbf{Inn} & \to & \mathbf{Rel}_! \, ; \end{array}$$

preserving the interpretation. Consequently, $\rightarrowtriangle$-**Vis** and $\rightarrowtriangle$-**Inn** both support positional equivalence which is preserved by all operations used in the interpretation of $\mathsf{PCF}_{/\!/}$.

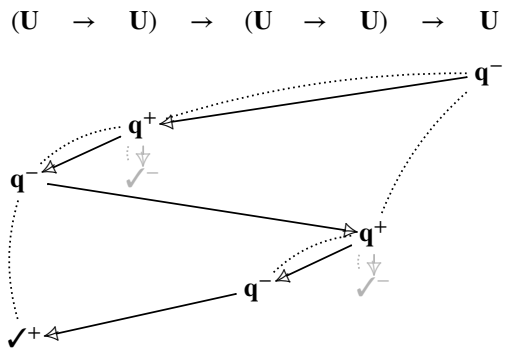


Figure 10.18: A well-bracketed innocent strategy with a non well-bracketed gcc

A syntax for positions. It will be convenient to have a syntactic description of the shape of positions on various −-boards constructions. We give this description in Figure 10.17, directly applying the bijection $s^{\otimes}_{A,B}$ from Proposition 10.4.11, $s^{\multimap}_{A,C}$ and $s^{\&}_{C,D}$ from Lemma 10.4.14, $s^{!}_{S}$ Proposition 10.4.12, and the obvious variant of $s^{\otimes}_{A,B}$ for $\vdash$.

10.5 Globularity

Though we regard parallel innocence (and *visibility*) as the key conditions to understand the causal shape of pure parallel programs, it is not yet precise enough to entail our definability result. For definability purposes, we need yet again more rigid constraints on the causal shape of strategies, taking the form of a final condition called *globularity*.

10.5.1 Motivation and Definition

Well-bracketed gccs. In Section 10.1.2, we introduced *pre-innocence* with the intuition that a gcc may be regarded as a standalone sequential program, and pre-innocence brings constraints as to how these "sequential threads" may be forked and merged. But if these "threads" do correspond to good old sequential computations, they should certainly satisfy a condition akin to sequential well-bracketing (see Definition 3.2.5).

Strategies in $\multimap\!\triangleright$-**Strat** already satisfy a well-bracketing condition (Definition 9.3.2). However, it is based in the *logical* well-bracketing of non-alternating plays (Definition 5.1.2), which is insufficient to ensure that gccs are (sequentially) well-bracketed: this is illustrated in Figure 10.18, displaying a well-bracketed innocent causal strategy with a non well-bracketed gcc highlighted. This invites the condition, for $\boldsymbol{\sigma}$ in $\multimap\!\triangleright$-**Inn**:

wb-threads: any $\rho \in \mathsf{gcc}(\boldsymbol{\sigma})$ is well-bracketed in the sense of Definition 3.2.5.

However, this is not yet enough to fully capture higher-order pure parallel computation. The final condition required has to do with which gccs can be legally merged.

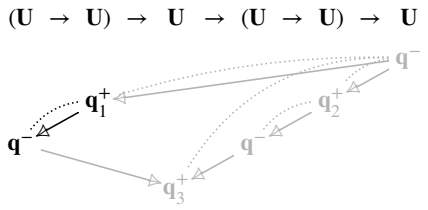


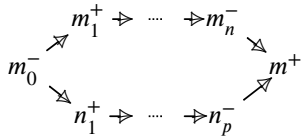
Figure 10.19: An innocent non-globular strategy

Intuitively, all causal merges in $\mathsf{PCF}_{/\!/}$ are generated by the parallel *let* binding

$$\textbf{let} \begin{pmatrix} x_1 & = & N_1 \\ x_2 & = & N_2 \end{pmatrix} \textbf{in } M$$

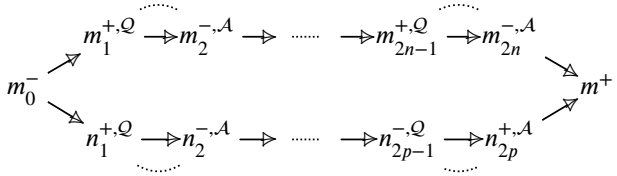
where the two branches are causally merged when the computation of N_1 has *returned*, and the computation of N_2 has *returned* also – so all Questions involved in the computation of N_1 have an answer, and likewise for N_2. But not all parallel innocent strategies satisfy this pattern: the strategy shown in Figure 10.19 is parallel innocent, satisfies condition *wb-threads*, but is not definable within $\mathsf{PCF}_{/\!/}$ as there is a causal merge of computation branches that have not *returned* yet. Banning this invites the condition:

globules: for any diagram with $X = \{m_1, \ldots, m_n\}$ and $Y = \{n_1, \ldots, n_p\}$ disjoint:



in $\boldsymbol{\sigma}$, then every question in X (resp. Y) is answered in X (resp. Y).

We call such a diagram a *globule*. Note that only Player can merge parallel threads, and only if he is responsible for the fork (by parallel innocence), so polarities in the definition of globules are not restrictive. Additionally, one can also observe that globules must always have Question/Answer assignments as in the diagram:



(10.2)

Indeed, if m_1 was an answer, it would be maximal in $\boldsymbol{\sigma}$ (by Lemma 6.1.16, as by *answer-closing*, answers are maximal in A) and the merge would be impossible. So m_1 is a question, and by *globules* it has an answer in $\{m_1, \ldots, m_{2n}\}$. By Lemma 6.1.16 this answer depends immediately on m_1 in $\boldsymbol{\sigma}$ and so must be m_2. Repeating this we get the description above. Hence by *globules*, parallel threads that eventually might merge have to follow a strict call/return discipline. Altogether, we set:

Definition 10.5.1. *Consider A, B mixed boards, and $\sigma \in \multimap\!\triangleright$-**Inn**$(A, B)$.*
*Then σ is **globular** if it satisfies conditions* wb-threads *and* globules.

10.5.2 Composition of Globularity

In this section, we develop the compositional structure of globularity.

Basic categorical structure. As usual when introducing a new condition on strategies, we must show that it is compatible with all our constructions on strategies – and as is often the case, the most crucial point is compatibility with copycat and composition.

Recall that the definition of globularity concerns morphisms in $\multimap\!\triangleright$-**Inn**, defined as (a subcategory of) a Kleisli category, *i.e.* its morphisms are certain strategies on $!A \vdash B$ where A and B are mixed boards. In examining compatibility of globularity with the categorical structure, it is convenient to work with the basic constructions in **NTCG** (copycat, composition) rather than those in the Kleisli category (Kleisli composition). Here we shall do that, keeping in mind the slight abuse that technically the definition of globularity refers to the Question/Answer labelling, which was not required on boards, but arises only in mixed boards, via the link with the arena.

Proposition 10.5.2. *The copycat strategy $\mathbf{cc}_A : A \vdash A$ is globular.*

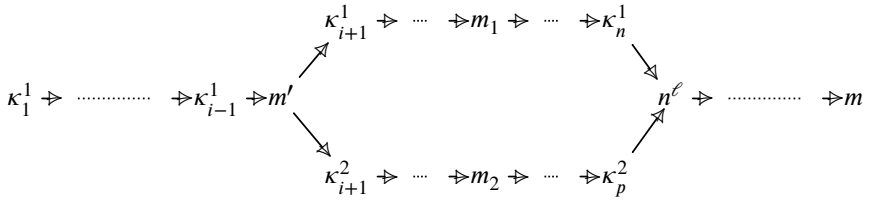
Proof. The condition *wb-threads* boils down to the proof that P-views of the alternating copycat are well-bracketed in the sense of Definition 3.2.5. *Globularity* holds vacuously, as the event structure for copycat has no causal merge. □

As usual, this reasoning applies to all copycat strategies involved in the interpretation: projections, structural morphisms for the monoidal structure, dereliction, etc.

Now, we establish stability of globularity under composition:

Proposition 10.5.3. *Consider $\sigma : A \vdash B$ and $\tau : B \vdash C$ globular.*
Then, $\tau \odot \sigma : A \vdash C$ is globular.

Proof. For *wb-threads*, we first prove that all gccs in $\tau \circledast \sigma$ are well-bracketed – this is a variation of the fact that alternating innocent strategies are well-bracketed iff their meager form is (Proposition 3.2.13). For *globules*, by the same reasoning as in Proposition 10.3.2, a globule in $\tau \odot \sigma$ originates in a diagram picturing causal chains



in $\tau \circledast \sigma$, constructed as and satisfying the same properties as in Proposition 10.3.2 – in particular n has polarity ℓ without loss of generality. But then as in Proposition 10.3.2,

$$\mathsf{gce}(\ulcorner \kappa^1_{\leq n} \urcorner^\sigma, \ulcorner \kappa^2_{\leq p} \urcorner^\sigma) = \mathsf{gce}(\kappa^1_{\leq n}, \kappa^2_{\leq p})_\sigma = m'_\sigma \,,$$

by the forking lemma, *i.e.* $\boldsymbol{\sigma}$ sees the same fork as in $\boldsymbol{\tau} \circledast \boldsymbol{\sigma}$. Since $\boldsymbol{\sigma}$ is *globular*, all questions between $m'_{\boldsymbol{\sigma}}$ and $n_{\boldsymbol{\sigma}}$ in both $\ulcorner \kappa^1_{\leq n} \urcorner^{\boldsymbol{\sigma}}$ and $\ulcorner \kappa^2_{\leq n} \urcorner^{\boldsymbol{\sigma}}$ have an answer. But as observed above, gccs in $\boldsymbol{\tau} \circledast \boldsymbol{\sigma}$ are well-bracketed, which entails that all questions between m' and n in κ^1 and κ^2 also have an answer – hence, *globules* is satisfied in $\boldsymbol{\tau} \odot \boldsymbol{\sigma}$. □

Interpretation of $\mathsf{PCF}_{/\!/}$. As for parallel innocence, preservation of globularity for all other operations is straightforward. We omit the details of that, and of the fact that the basic primitives of $\mathsf{PCF}_{/\!/}$ yields globular strategies. Altogether, we get:

Theorem 10.5.4. *there is* $\multimap\!\triangleright$**-Glob**, *a cartesian closed sub-*$\sim$*-category of* $\multimap\!\triangleright$**-Strat**, *with morphisms restricted to globular strategies.*

Moreover, $\multimap\!\triangleright$**-Glob** *supports the interpretation of* $\mathsf{PCF}_{/\!/}$.

10.6 History and Related Work

Parallel innocence and its accompanying theory (the forking lemma and stability under composition, the deadlock-free lemma and the collapse to **Rel**, globularity) have been developed in close collaboration with Simon Castellan during his PhD thesis[4] – they first appear in published form in [Castellan et al., 2015]; though the presentation given here differs significantly, having matured quite a bit since then.

Parallel innocence was first developed aiming for a finite definability result for (essentially) $\mathsf{PCF}_{/\!/}$, which was achieved in [Castellan et al., 2015]. It became a cornerstone of subsequent developments in concurrent games. Though surprisingly, it is not *parallel innocence* per se that turned out so useful in our later work, but *visibility* and the deadlock-free lemma! Indeed, this is one of the key ingredients in our subsequent results relating concurrent games and relational models, in the probabilistic [Castellan et al., 2018b], the general weighted case [Clairambault and Paquet, 2021], the quantum case [Clairambault and de Visme, 2020] and more recently for generalized species of structure [Clairambault et al., 2023b]. In a sense, the deadlock-free lemma explains why the relational models work at all for pure parallel programs: no deadlocks can arise in composing them, so that causal information has no impact on possible synchronizations and can safely be forgotten. No such miracle occurs for effectful programming languages, for which the causal information is unavoidable.

Related work. In [Melliès and Mimram, 2007], Melliès and Mimram have sketched a notion of innocent strategies in the setting of non-alternating asynchronous games (which are essentially the linear, causally deterministic part of **TCG**), with the purpose of eliminating deadlocks from composition. This appears with more details in Mimram's PhD thesis [Mimram, 2008]. According to their definition, innocence amounts to a correctness criterion inspired from linear logic proof nets. We are not aware of a direct link with the present developments, though this would be interesting to explore.

[4] More precisely, visibility and parallel innocence were invented on the 7th of february 2014, a (probably) rainy friday – as witnessed by Figure 8.5.

Chapter 11

Sequentiality

The conditions of *parallel innocence* and *globularity* developed in the previous chapter enforce *pure computation*, which may still be parallel. In contrast, the conditions developed in this chapter aim to reinstate *sequentiality*, which may still be impure.

Sequentiality is probably the simplest of our conditions on strategies, but it still comes with a few subtleties. Firstly, the concept of sequentiality is of course tied to *time*, which is notably absent from concurrent strategies! So it makes sense that the definition will have to refer to linearizations of strategies, in the style developed in the unfolding of causal to non-alternating strategies in Chapter 9. Secondly, a phenomenon occurs that is similar to what we encountered for parallel innocence: as *non-determinism* arises from the combination of parallelism and state, *sequentiality* must reinstate it – but not quite under the same form as in the previous chapter: it is hard to unify *sequential* determinism and *causal* determinism (see discussion in Section 14.2.2).

Outline. The developments of this chapter will proceed as follows. In Section 11.1, we develop sequentiality, aiming to show that sequential strategies form a sub-relative Seely $\sim$-category **NTCG-Seq**, that refines to a sub-cartesian closed $\sim$-category $\multimap_{\triangleright}$-**Seq** of $\multimap_{\triangleright}$-**Strat**, supporting the interpretation of IA. In Section 11.2, we show that the non-alternating unfolding $\circlearrowleft$-Unf of Chapter 9 refines into an interpretation-preserving

$$\Updownarrow\uparrow\text{-}\mathsf{Unf} \;:\; \multimap_{\triangleright}\text{-}\mathbf{Seq} \to \Updownarrow\uparrow\text{-}\mathbf{Strat}$$

which entails that $\multimap_{\triangleright}$-**Seq** is intensionally fully abstract for IA. In Section 11.3, we show that in combination with *globularity*, $\Updownarrow\uparrow$-Unf yields an interpretation-preserving functor

$$\Updownarrow\uparrow\text{-}\mathsf{Unf} \;:\; \multimap_{\triangleright}\text{-}\mathbf{SeqGlob} \to \Updownarrow\uparrow\text{-}\mathbf{InnWB}$$

proving $\multimap_{\triangleright}$-**SeqGlob** intensionally fully abstract for PCF.

11.1 Sequentiality

Sequentiality has no interaction with mixed boards or well-bracketing, hence we carry it out in the relative Seely $\sim$-category **NTCG**. Likewise, for succinctness, the examples shall be based on the *small* interpretation of ground types (see Section 9.1.1) – although the actual interpretation remains the same as for $\mathtt{IA}_{/\!/}$, *i.e* large.

11.1.1 Definition of Sequentiality

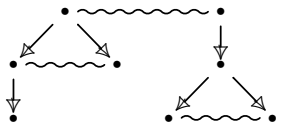
What is a sequential causal strategy? The right answer is more subtle than it looks.

Sequential event structures. As strategies are certain event structures, it seems reasonable to start by looking at what is a sequential event structure. Event structures are concurrent generalizations of trees or forests: there is a natural representation of forests as event structures, which entails a canonical notion of sequentiality:

Definition 11.1.1. *An event structure E is **sequential** if it satisfies the conditions:*

forestial: *for all $e_1, e_2 \leq_E e'$, then $e_1 \leq_E e_2$ or $e_2 \leq_E e_1$.*
#-branching: *for all $e_1, e_2 \in E$, $e_1 \#_E e_2$ iff they are causally incomparable.*

In other words, a sequential event structure has the shape of a forest as in



whose configurations, always totally ordered, correspond to branches of the forest.

However, it is quickly apparent that this is not useful as a definition of sequential causal strategies. Technically, *receptivity* entails that conflicts between negative moves should be exactly as in the game. But negative events may not conflict in the game, as in *e.g.* the initial moves of $!\mathbf{B}$; so no strategy can actually be sequential on $!\mathbf{B}$.

More fundamentally, it turns out that although IA is indeed a sequential language, the shape of causality in the corresponding strategies is certainly *not* sequential. We show two examples: first in Figure 11.1 we repeat the example of Figure 10.3. This is (an affine linear version of) the interpretation of the term in (10.1), a term of IA which should hence be deemed sequential – we see that even though the program itself does not spawn parallel threads, the strategy cannot be sequential as in Definition 11.1.1 as it must show how the program reacts when Opponent does. A similar phenomenon occurs for pure sequential innocent programs: Figure 11.2 shows a configuration of

$$\vdash \lambda f^{\mathbb{U}\to\mathbb{U}}.\, f\ \mathbf{skip};\ f\ \mathbf{skip};\ \mathbf{tt} \ : \ (\mathbb{U} \to \mathbb{U}) \to \mathbb{B}$$

which has a typical shape for pure sequential programs. We see that although the causal shape of pure sequential programs is indeed forestial, the branches do *not* conflict.

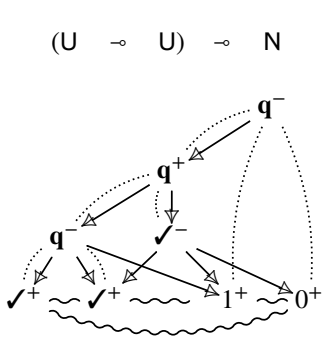


Figure 11.1: A sequential strategy

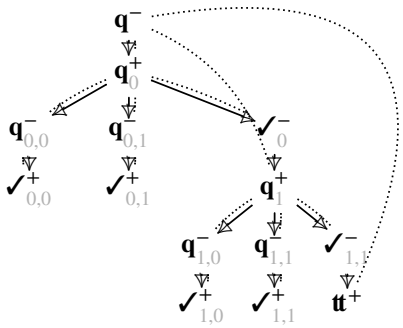


Figure 11.2: A sequential innocent configuration

Alternating linearizations. So how shall we instead define sequentiality?

Let us set the stage for the definition. First, recall that the definition of *linearizations* (Definition 9.2.1) associates to any event structure the set $\mathscr{L}(E)$ of words of $|E|$, which are non-repetitive and whose prefixes form configurations. Now through its display map, σ has polarities, so we can look at those *alternating linearizations*:

Definition 11.1.2. *Consider E an event structure with polarities.*

*The set $\downarrow\uparrow$-$\mathscr{L}(E)$ of **alternating linearizations** comprises those $s \in \mathscr{L}(E)$ such that polarities alternate and the first move (if s is non-empty) is negative.*

We insist that $\downarrow\uparrow$-$\mathscr{L}(\sigma)$ comprises words of events of σ: we have not yet projected σ to the game. Its elements are internal witnesses of sequential executions, invisible to the environment. Despite this, we can perform on $\downarrow\uparrow$-$\mathscr{L}(\sigma)$ certain operations usually done on alternating legal plays on arenas. In particular, we can compute the *P-view*:

Definition 11.1.3. *Consider $\sigma \in \mathbf{NTCG}(A, B)$ pointed, and $s \in \downarrow\uparrow$-$\mathscr{L}(\sigma)$.*

*Its **P-view** is the subsequence $\ulcorner s \urcorner$ defined by:*

$$\begin{array}{rcll} \ulcorner \varepsilon \urcorner & = & \varepsilon & \\ \ulcorner s m^+ \urcorner & = & \ulcorner s \urcorner m^+ & \\ \ulcorner s n^+ s' m^- \urcorner & = & \ulcorner s \urcorner n^+ m^- & \text{if } m \text{ points to } n, \\ \ulcorner s m^- \urcorner & = & m^- & \text{if } m \text{ is minimal,} \end{array}$$

*where m **points to** n if $\partial_\sigma(n) \rightarrow_{A \vdash B} \partial_\sigma(m)$; or if $\partial_\sigma(m)$ is minimal in A and $n = \mathrm{init}(m)$.*

*We say $s \in \downarrow\uparrow$-$\mathscr{L}(\sigma)$ is **P-visible** if for all $t \sqsubseteq s$, $\ulcorner t \urcorner \in \downarrow\uparrow$-$\mathscr{L}(\sigma)$.*

This mimics the traditional definition (Definition 3.2.7), except that sequences $s \in \downarrow\uparrow$-$\mathscr{L}(\sigma)$ do not have pointers; those are derived from the immediate causality in the game as in Lemma 9.2.8. Observe that this is where the hypothesis *pointed* is needed: as causal strategies play on $A \vdash B$, the causal dependency from initial moves in B to initial moves in A may only be recovered from the immediate causality in σ: pointedness ensures that moves minimal in A have a unique initial dependency which must be minimal in B, which ensures that that the justifier is always well-defined.

Sequentiality. Programs of IA will respect sequentiality only within a sequential execution: we wish to impose that in any state reachable in a sequential execution, Player will not throw parallel threads. But as explained in Section 10.1, as for parallel innocence, sequentiality must also reinstate a notion of determinism. Altogether we get:

Definition 11.1.4. *A strategy* $\sigma \in \mathbf{NTCG}(A, B)$ *is* ***sequential*** *if:*

$$\begin{array}{rl} \text{pointed:} & \text{if } s \in \sigma\text{, there is a unique } \mathrm{init}(s) \leq_\sigma s \text{ minimal in } \sigma, \\ \text{sequential determinism:} & \text{if } tn_1^+, tn_2^+ \in \Downarrow\uparrow\text{-}\mathscr{L}(\sigma)\text{, then } n_1 = n_2, \\ \text{sequential visibility:} & \text{if } s \in \Downarrow\uparrow\text{-}\mathscr{L}(\sigma)\text{, writing } \ulcorner s \urcorner = s_1 \dots s_n \text{ and } 1 \leq j \leq n, \\ & \text{if } s_j \text{ is non-minimal then there is } 1 \leq i < j \text{ s.t. } \mathsf{j}(s_j) = s_i, \end{array}$$

where the justifier $\mathsf{j}(m)$ *of* $m \in \sigma$ *refers to Definition 10.2.1.*

For *sequential determinism*, more than asking that the unfolding acts deterministically on alternating plays, this condition ensures that no *internal* non-deterministic choice is alternatingly reachable, even when this choice would yield no observable non-deterministic behaviour (surprisingly for us, this turned out necessary).

Sequential visibility is perhaps puzzling, as P-visibility of alternating strategies (Definition 3.2.8) is usually associated not to sequentiality, but to the absence of higher-order state (see Section 3.3.2). From a given control point, a P-visible strategy may only call a procedure bound within the branch of the syntax tree leading to that control point. In contrast, with higher-order state, a program may call a procedure stored in the memory, originating from a remote program phrase outside the current branch. This phenomenon is independent of sequentiality, but in **NTCG** the causality due to the syntax tree blurs with that passing via state or semaphores. So strategies arising from the interpretation of $\mathsf{IA}_{/\!/}$ are "morally" P-visible but formalizing this is nontrivial[1]. For us it is not worth the trouble as P-visibility is not required for full abstraction for $\mathsf{IA}_{/\!/}$[2]. Consequently, it suffices to reinstate it once we restrict to sequential strategies.

The strategy in Figure 10.6 is not sequential: it fails sequential determinism as two Player moves are available after the initial move. In contrast, the strategy in Figure 11.1 *is* sequential: although it may perform non-deterministic choices and play Player moves in parallel, none of these behaviours are reachable in an alternating execution.

We mention a direct consequence of sequentiality:

Lemma 11.1.5. *Consider* $\sigma \in \mathbf{NTCG}(A, B)$ *a sequential causal strategy.*
Then, it automatically satisfies the following additional condition:

$$\text{reachable sequentiality:} \quad \text{if } tn^+ \in \mathscr{L}(\sigma)\text{, if } t \in \Downarrow\uparrow\text{-}\mathscr{L}(\sigma) \text{ then } tn \in \Downarrow\uparrow\text{-}\mathscr{L}(\sigma).$$

Proof. Consider $tn^+ \in \mathscr{L}(\sigma)$ with $t \in \Downarrow\uparrow\text{-}\mathscr{L}(\sigma)$. If $tn \notin \Downarrow\uparrow\text{-}\mathscr{L}(\sigma)$, this necessarily means that tn^+ fails alternation, *i.e.* $t = sm^+$. But then, we argue that $sn^+ \in \mathscr{L}(\sigma)$ as well. Indeed, this may fail only if $m^+ \rightarrowtriangle_\sigma n^+$. But by *courteous*, this implies $\partial_\sigma(m) \rightarrowtriangle_{A \vdash B} \partial_\sigma(n)$ as well. But as a board, $A \vdash B$ is *alternating*, contradiction. □

[1] This was done by Laird in the first interleaving games model [Laird, 2001b], via threading information.

[2] Proposition 5.1.11 shows that non-visible behaviour characteristic of higher-order state can be mimicked by running several threads in parallel and using signaling via *interference* to jump control between them.

This means that restricting a sequential strategy to its alternating linearizations only cuts at Opponent moves: it never prevents Player from playing.

11.1.2 Categorical Structure

We show that sequential causal strategies form a sub-$\sim$-category of **NTCG**.

Sequentiality of copycat. First, we show that the copycat strategy is sequential:

Lemma 11.1.6. *Consider A a $-$-board. Then, $\mathbf{ac}_A : A \vdash A$ is sequential.*

Proof. First, $\mathbf{ac}_A$ is pointed, as its causal shape is forestial. For the other two conditions, we observe that the even-length alternating linearizations of $\mathbf{ac}_A$ are exactly those words on $s \in |A_1 \vdash A_2|^*$ (with indices used for disambiguation only) such that for all even-length prefix $t \sqsubseteq s$, $t \upharpoonright A_1 = t \upharpoonright A_2$, with the restriction $\upharpoonright$ defined in the obvious way. The two conditions *sequential determinism* and *sequential visibility* follow easily. □

Composition of sequential determinism. As often when introducing a new condition, much of the work required is in checking stability under composition. Here in particular, something is striking: recall from Section 4.1.3 that proving associativity of composition for alternating strategies involves the study of a "state diagram" for interactions (in contrast, nothing of the sort shows up when constructing the associator for the composition of causal strategies) – this diagram intervenes again here.

Let us start by introducing some further terminology. Consider $\boldsymbol{\sigma} : A \vdash B$ and $\boldsymbol{\tau} : B \vdash C$ two causal strategies, and $p \in \boldsymbol{\tau} \circledast \boldsymbol{\sigma}$ an event of the interaction. We have analyzed above Lemma 6.2.14 the different possible polarities of p – at least one of $p_{\boldsymbol{\sigma}} \in \boldsymbol{\sigma}$ and $p_{\boldsymbol{\tau}} \in \boldsymbol{\tau}$ must be defined, possibly both if p occurs in B, in which case p does not appear in the composition. Otherwise p occurs in A or C and we have $p \in \boldsymbol{\tau} \odot \boldsymbol{\sigma}$; let us write $p_\odot = p \in \boldsymbol{\tau} \odot \boldsymbol{\sigma}$ if that is the case, and undefined otherwise.

Applying these projections move-by-move, from $s \in \mathscr{L}(\boldsymbol{\tau} \circledast \boldsymbol{\sigma})$ we get:

Lemma 11.1.7. *Consider $s = s_1 \dots s_n \in \mathscr{L}(\boldsymbol{\tau} \circledast \boldsymbol{\sigma})$. Then, forming*

$$\begin{array}{rcl} s_{\boldsymbol{\sigma}} & = & (s_1)_{\boldsymbol{\sigma}} \dots (s_n)_{\boldsymbol{\sigma}} \\ s_{\boldsymbol{\tau}} & = & (s_1)_{\boldsymbol{\tau}} \dots (s_n)_{\boldsymbol{\tau}} \\ s_\odot & = & (s_1)_\odot \dots (s_n)_\odot , \end{array}$$

removing moves where the projection is undefined, we have that

$$u_{\boldsymbol{\sigma}} \in \mathscr{L}(\boldsymbol{\sigma}) , \qquad u_{\boldsymbol{\tau}} \in \mathscr{L}(\boldsymbol{\tau}) , \qquad u_\odot \in \mathscr{L}(\boldsymbol{\tau} \odot \boldsymbol{\sigma}) .$$

Proof. From the definition, it is immediate that for all $x^{\boldsymbol{\sigma}} \in \mathscr{C}(\boldsymbol{\sigma})$, $x^{\boldsymbol{\tau}} \in \mathscr{C}(\boldsymbol{\tau})$ causally compatible, we have $(x^{\boldsymbol{\tau}} \circledast x^{\boldsymbol{\sigma}})_{\boldsymbol{\sigma}} = x^{\boldsymbol{\sigma}} \in \mathscr{C}(\boldsymbol{\sigma})$ (where $(x^{\boldsymbol{\tau}} \circledast x^{\boldsymbol{\sigma}})_{\boldsymbol{\sigma}}$ is obtained as for linearizations above, applying $(-)_{\boldsymbol{\sigma}}$ event-by-event), $(x^{\boldsymbol{\tau}} \circledast x^{\boldsymbol{\sigma}})_{\boldsymbol{\tau}} = x^{\boldsymbol{\tau}} \in \mathscr{C}(\boldsymbol{\tau})$ and $(x^{\boldsymbol{\tau}} \circledast x^{\boldsymbol{\sigma}})_\odot \in \mathscr{C}(\boldsymbol{\tau} \odot \boldsymbol{\sigma})$ (by definition of $\boldsymbol{\tau} \odot \boldsymbol{\sigma}$). The claims on linearizations follow. □

A linearization $u \in \mathscr{L}(\sigma)$ may be – or not – alternating, *i.e.* $u \in \Downarrow\uparrow\text{-}\mathscr{L}(\sigma)$. If it is, then as for legal plays on arenas (Definition 3.1.4) two cases are possible: if it has even length (so that the last move is by Player), we say that it is **in state** O. If it has odd length (so that the last move is by Opponent), we say that it is **in state** P.

Putting the projections of the lemma above to work, we have the state diagram:

Lemma 11.1.8. *Consider A, B, C $-$-boards, and $\sigma : A \vdash B$, $\tau : B \vdash C$ sequential.*

For any $u \in \mathscr{L}(\tau \circledast \sigma)$ such that $u_\odot \in \Downarrow\uparrow\text{-}\mathscr{L}(\tau \odot \sigma)$ we have $u_\sigma \in \Downarrow\uparrow\text{-}\mathscr{L}(\sigma)$ and $u_\tau \in \Downarrow\uparrow\text{-}\mathscr{L}(\tau)$, and we are in one of the following three cases:

(1) *$u_\sigma, u_\tau, u_\odot$ are respectively in state O, O, O,*
(2) *$u_\sigma, u_\tau, u_\odot$ are respectively in state O, P, P,*
(3) *$u_\sigma, u_\tau, u_\odot$ are respectively in state P, O, P.*

Proof. By induction on u. If u is empty, this is clear. Consider $um \in \mathscr{L}(\tau \circledast \sigma)$. By induction hypothesis u is in one of cases *(1)*, *(2)* and *(3)*. We distinguish cases:

(1) Seeking a contradiction, assume m occurs in B. Then, one of m_σ or m_τ is positive – say *w.l.o.g.* the former. By IH, u_σ is alternating in state O, so ends with a Player move. But so, $u_\sigma m_\sigma^+ \in \mathscr{L}(\sigma)$ with $u_\sigma \in \Downarrow\uparrow\text{-}\mathscr{L}(\sigma)$, so $u_\sigma m_\sigma \in \Downarrow\uparrow\text{-}\mathscr{L}(\sigma)$ since σ satisfies reachable sequentiality, contradiction. So, m occurs in A or C – assume *w.l.o.g.* in A. Since $u_\odot$ is in state O, m is negative – then, it is direct that um satisfies *(3)*.

(2) First assume m occurs in A. Since $u_\odot \in \Downarrow\uparrow\text{-}\mathscr{L}(\tau \circledast \sigma)$ is in state P, then m is positive; then m_σ is positive, contradicting reachable sequentiality of σ with the fact that u_σ is in state O. Similarly, if m occurs in B it has polarity $\nearrow$ and we transition to *(3)*, and if m occurs in C it has polarity $\nearrow$ and we transition to *(1)*.

(3) Symmetric to *(2)*. □

As long as the external Opponent respects the alternation discipline, interactions follow the familiar *state diagram* of interactions in alternating game semantics, shown in Figure 4.2. None of the interacting agents can be the first to break alternation, so the interaction ends up alternating. It follows that $\tau \odot \sigma$ satisfies *sequential determinism*:

Proposition 11.1.9. *Consider $\sigma : A \vdash B$ and $\tau : B \vdash C$ sequential causal strategies.*

Then, $\tau \odot \sigma$ satisfies sequential determinism*.*

Proof. Consider $tn_1^+, tn_2^+ \in \Downarrow\uparrow\text{-}\mathscr{L}(\tau \odot \sigma)$ completed to $u_1n_1, u_2n_2 \in \mathscr{L}(\tau \circledast \sigma)$, and u' the greatest common prefix of u_1n_1 and u_2n_2, with $u'm_1 \sqsubseteq u_1n_1$ and $u'm_2 \sqsubseteq u_2n_2$. Necessarily, the visible restriction of u' is a prefix $t' \sqsubseteq tn_i$, so in particular $t' \in \Downarrow\uparrow\text{-}\mathscr{L}(\tau \odot \sigma)$. We distinguish cases on Lemma 11.1.7 applied to u'.

For *(1)* (state OOO), m_1 and m_2 must both be the next negative event appearing in t, so $m_1 = m_2$, contradiction. For *(2)* (state OPP), m_1 and m_2 both have polarity $\nearrow$ and we have $u'_\tau \in \Downarrow\uparrow\text{-}\mathscr{L}(\tau)$, $u'_\tau(m_1)^+_\tau, u'_\tau(m_2)^+_\tau \in \Downarrow\uparrow\text{-}\mathscr{L}(\tau)$. Hence $(m_1)_\tau = (m_2)_\tau$ since τ satisfies sequential determinism – so $m_1 = m_2$ by local injectivity of the projection Π_τ; contradiction. For *(3)* (state POP), the reasoning is symmetric. □

By the state diagram, in an alternating interaction, only one agent has control at any point. So any alternatingly reachable non-deterministic choice in $\tau \odot \sigma$ can be attributed

to $\boldsymbol{\sigma}$ and $\boldsymbol{\tau}$. Since they are both deterministic, no such choice exists. We single out the linearizations of an interaction $u \in \mathscr{L}(\boldsymbol{\tau} \circledast \boldsymbol{\sigma})$ that satisfy the state diagram of interactions as in Lemma 11.1.8: the *alternating interactions*.

Definition 11.1.10. *Consider $u \in \mathscr{L}(\boldsymbol{\tau} \circledast \boldsymbol{\sigma})$. It is an **alternating interaction**, written $u \in \updownarrow\!\uparrow\text{-}\mathscr{L}(\boldsymbol{\tau} \circledast \boldsymbol{\sigma})$, if $u_{\odot} \in \updownarrow\!\uparrow\text{-}\mathscr{L}(\boldsymbol{\tau} \odot \boldsymbol{\sigma})$, $u_{\boldsymbol{\sigma}} \in \updownarrow\!\uparrow\text{-}\mathscr{L}(\boldsymbol{\sigma})$ and $u_{\boldsymbol{\tau}} \in \updownarrow\!\uparrow\text{-}\mathscr{L}(\boldsymbol{\tau})$.*

The notation $\updownarrow\!\uparrow\text{-}\mathscr{L}(\boldsymbol{\tau} \circledast \boldsymbol{\sigma})$ should hopefully not cause confusion, since $\boldsymbol{\tau} \circledast \boldsymbol{\sigma}$ does not have polarities: the polarity of synchronized events is indetermined.

A fundamental property is that an alternating linearization between sequential strategies is necessarily witnessed by an alternating interaction:

Lemma 11.1.11. *Consider $\boldsymbol{\sigma} : A \vdash B$ and $\boldsymbol{\tau} : B \vdash C$ sequential, and $u \in \mathscr{L}(\boldsymbol{\tau} \circledast \boldsymbol{\sigma})$. If $u_{\odot} \in \updownarrow\!\uparrow\text{-}\mathscr{L}(\boldsymbol{\tau} \odot \boldsymbol{\sigma})$, then $u \in \updownarrow\!\uparrow\text{-}\mathscr{L}(\boldsymbol{\tau} \circledast \boldsymbol{\sigma})$.*

Proof. By induction on u. Consider $um \in \mathscr{L}(\boldsymbol{\tau} \circledast \boldsymbol{\sigma})$ such that $(um)_{\odot} \in \updownarrow\!\uparrow\text{-}\mathscr{L}(\boldsymbol{\tau} \odot \boldsymbol{\sigma})$. By induction hypothesis, $u \in \updownarrow\!\uparrow\text{-}\mathscr{L}(\boldsymbol{\tau} \circledast \boldsymbol{\sigma})$, so it must be in one of the states of Lemma 11.1.7. For *(1)* (state OOO). Then necessarily m is negative, and it is immediate that the invariant is preserved. For *(2)* (state OPP), then we reason by cases depending on when m occurs. If it occurs in A, then it must be positive in $\boldsymbol{\tau} \odot \boldsymbol{\sigma}$ since $(um)_{\odot} \in \updownarrow\!\uparrow\text{-}\mathscr{L}(\boldsymbol{\tau} \odot \boldsymbol{\sigma})$. Then $m_{\boldsymbol{\sigma}}$ is defined and positive. So $u_{\boldsymbol{\sigma}} \in \updownarrow\!\uparrow\text{-}\mathscr{L}(\boldsymbol{\sigma})$ and $u_{\boldsymbol{\sigma}} m^+_{\boldsymbol{\sigma}} \in \mathscr{L}(\boldsymbol{\sigma})$, thus by Lemma 11.1.5 we have $u_{\boldsymbol{\sigma}} m^+_{\boldsymbol{\sigma}} \in \updownarrow\!\uparrow\text{-}\mathscr{L}(\boldsymbol{\sigma})$ as well, contradicting that $u_{\boldsymbol{\sigma}}$ is in state O. Thus m occurs in B or C, so that $m_{\boldsymbol{\tau}}$ is defined. But then $u_{\boldsymbol{\tau}} \in \updownarrow\!\uparrow\text{-}\mathscr{L}(\boldsymbol{\tau})$ and $u_{\boldsymbol{\tau}} m_{\boldsymbol{\tau}} \in \mathscr{L}(\boldsymbol{\tau})$, thus by Lemma 11.1.5 we have $u_{\boldsymbol{\tau}} m_{\boldsymbol{\tau}} \in \updownarrow\!\uparrow\text{-}\mathscr{L}(\boldsymbol{\tau})$, thus $m_{\boldsymbol{\tau}}$ is positive. If m occurs in B then the invariant is preserved and um is in state POP, if m occurs in C then the invariant is preserved and um is in state OOO. Finally, *(3)* (state POP) is symmetric. □

Composition of sequential visibility. Next, we must prove that the composition of sequential strategies satisfies *sequential visibility*. This is a direct variation of the stability under composition of P-visible strategies in alternating game semantics.

As announced, we generalize P-views to alternating interactions:

Definition 11.1.12. *For $u \in \updownarrow\!\uparrow\text{-}\mathscr{L}(\boldsymbol{\tau} \circledast \boldsymbol{\sigma})$, we define the subsequence $\ulcorner u \urcorner$ with*

$$
\begin{array}{rcll}
\ulcorner \varepsilon \urcorner & = & \varepsilon & \\
\ulcorner um \urcorner & = & \ulcorner u \urcorner m & \textit{(if } m \textit{ is positive or synchronized)} \\
\ulcorner un^+ u' m^- \urcorner & = & \ulcorner u \urcorner n^+ m' & \textit{(if } m \textit{ is justified by } n\textit{)} \\
\ulcorner um^- \urcorner & = & m^- & \textit{(}m^-\textit{ minimal)}
\end{array}
$$

Those are the same clauses as in Defininition 11.1.3, applying to interactions rather than alternating linearizations of a causal strategy. The idea is the same: we browse the sequence, going back in time and following pointers of moves by the external Opponent.

The main argument behind the composition of sequential visibility is:

Lemma 11.1.13. *Consider $u \in \updownarrow\!\uparrow\text{-}\mathscr{L}(\boldsymbol{\tau} \circledast \boldsymbol{\sigma})$.*

Writing $\ulcorner u \urcorner = u_1 \dots u_n$ and for $1 \leq j \leq n$, if u_j is non-minimal in $\boldsymbol{\tau} \circledast \boldsymbol{\sigma}$ then there is $1 \leq i < j$ such that $\mathrm{j}(u_j) = u_i$ – where $\mathrm{j}(m)$ refers to Definition 10.2.3.

Proof. We show by mutual induction the following properties, for $u \in \Downarrow\Uparrow\text{-}\mathscr{L}(\boldsymbol{\tau} \circledast \boldsymbol{\sigma})$:

(1) writing $\ulcorner u \urcorner = u_1 \dots u_n$ and for $1 \leq j \leq n$, if u_j is non-minimal in $\boldsymbol{\tau} \circledast \boldsymbol{\sigma}$ then there is $1 \leq i < j$ such that $\mathrm{j}(u_j) = u_i$,
(2) if $u = vm$ with $m_{\boldsymbol{\sigma}}$ defined, $\ulcorner u_{\boldsymbol{\sigma}} \urcorner$ is a subsequence of $\ulcorner u \urcorner_{\boldsymbol{\sigma}}$,
(3) if $u = vm$ with $m_{\boldsymbol{\tau}}$ is defined, $\ulcorner u_{\boldsymbol{\tau}} \urcorner$ is a subsequence of $\ulcorner u \urcorner_{\boldsymbol{\tau}}$,

and we refer to *(1)* as $\ulcorner u \urcorner$ being *well-justified*.

For *(1)*, write $u = vn$. If n is negative, then either it is minimal (then the property is trivial), or $u = v_1 l^+ v_2 n^-$ where $\mathrm{j}(n^-) = l^+$ (say *w.l.o.g.* l is positive for $\boldsymbol{\sigma}$). Then by induction hypothesis, $\ulcorner v_1 l^+ \urcorner$ is well-justified and as $\ulcorner v_1 l^+ v_2 n^- \urcorner = \ulcorner v_1 l^+ \urcorner n^-$ with $\mathrm{j}(n) = l$, $\ulcorner u \urcorner$ is well-justified as well. Otherwise, $\ulcorner u \urcorner = \ulcorner v \urcorner n$ and by induction hypothesis, $\ulcorner v \urcorner$ is well-justified, but we must show that the justifier of n is in $\ulcorner v \urcorner$. Necessarily n is positive for $\boldsymbol{\sigma}$ or $\boldsymbol{\tau}$, say $\boldsymbol{\sigma}$ *w.l.o.g.*. As $(vn)_{\boldsymbol{\sigma}} = v_{\boldsymbol{\sigma}} n^+_{\boldsymbol{\sigma}} \in \Downarrow\Uparrow\text{-}\mathscr{L}(\boldsymbol{\sigma})$, as $\boldsymbol{\sigma}$ is sequentially visible, $n_{\boldsymbol{\sigma}}$ must point within $\ulcorner v_{\boldsymbol{\sigma}} \urcorner$. Now by Lemma 11.1.8, since n is positive for $\boldsymbol{\sigma}$, v is in state POP. If the last move of v occurs in C, since $v_\odot$ has state P it must be negative. But since $v_{\boldsymbol{\tau}}$ has state O it must be Player – contradiction, so the last move of v occurs in A or B and its projection to $\boldsymbol{\sigma}$ is defined. Thus by *(2)* of the induction hypothesis on v, $\ulcorner v_{\boldsymbol{\sigma}} \urcorner$ is a subsequence of $\ulcorner v \urcorner_{\boldsymbol{\sigma}}$. Now by sequential visibility of $\boldsymbol{\sigma}$, the justifier of $n_{\boldsymbol{\sigma}}$ is in $\ulcorner v_{\boldsymbol{\sigma}} \urcorner$, thus in $\ulcorner v \urcorner_{\boldsymbol{\sigma}}$, hence it $\ulcorner v \urcorner$.

For *(2)*, assume $m_{\boldsymbol{\sigma}}$ is defined. If it is positive, then by Lemma 11.1.8 v must be in state POP; write $v = v'n$. If n occurs in C, then it must be negative since $v_\odot$ is in state P, but this contradicts that $v_{\boldsymbol{\tau}}$ is in state O. Thus, n occurs in A or C, but then $n_{\boldsymbol{\sigma}}$ is defined (and negative). Hence, by induction hypothesis, $\ulcorner v_{\boldsymbol{\sigma}} \urcorner$ is a subsequence of $\ulcorner v \urcorner_{\boldsymbol{\sigma}}$. But $\ulcorner (um)_{\boldsymbol{\sigma}} \urcorner = \ulcorner u_{\boldsymbol{\sigma}} \urcorner m_{\boldsymbol{\sigma}}$ and $\ulcorner u \urcorner_{\boldsymbol{\sigma}} = \ulcorner v \urcorner_{\boldsymbol{\sigma}} m_{\boldsymbol{\sigma}}$, hence $\ulcorner u_{\boldsymbol{\sigma}} \urcorner$ is a subsequence of $\ulcorner u \urcorner_{\boldsymbol{\sigma}}$ as required. Next if $m_{\boldsymbol{\sigma}}$ is negative, write $u = v_1 n v_2 m$ with $\mathrm{j}(m) = n$. Two subcases arise. If m occurs in A, then it is negative and $\ulcorner u \urcorner = \ulcorner v_1 \urcorner nm$. Likewise, $\ulcorner u_{\boldsymbol{\sigma}} \urcorner = \ulcorner (v_1)_{\boldsymbol{\sigma}} \urcorner n^+_{\boldsymbol{\sigma}} m^-_{\boldsymbol{\sigma}}$, so the result follows by induction hypothesis. Finally if m occurs in B, then by *(1)*, $\ulcorner u \urcorner$ is well-justified, hence n appears in $\ulcorner v_1 n v_2 \urcorner$, *i.e.* $\ulcorner v_1 n v_2 \urcorner = \ulcorner v_1 \urcorner n v'_2$ and $\ulcorner u \urcorner_{\boldsymbol{\sigma}} = \ulcorner v_1 \urcorner_{\boldsymbol{\sigma}} n_{\boldsymbol{\sigma}} (v'_2)_{\boldsymbol{\sigma}} m_{\boldsymbol{\sigma}}$. Likewise by definition $\ulcorner u_{\boldsymbol{\sigma}} \urcorner = \ulcorner (v_1)_{\boldsymbol{\sigma}} \urcorner n_{\boldsymbol{\sigma}} m_{\boldsymbol{\sigma}}$. Finally by induction hypothesis $\ulcorner (v_1)_{\boldsymbol{\sigma}} \urcorner$ is a subsequence of $\ulcorner v_1 \urcorner_{\boldsymbol{\sigma}}$, hence the result follows.

For *(3)*, it is the same reasoning as above. □

From this, we may conclude:

Proposition 11.1.14. *Consider $\boldsymbol{\sigma} : A \vdash B$ and $\boldsymbol{\tau} : B \vdash C$ sequential causal strategies. Then, $\boldsymbol{\tau} \odot \boldsymbol{\sigma}$ satisfies* sequential visibility.

Proof. Consider $s \in \Downarrow\Uparrow\text{-}\mathscr{L}(\boldsymbol{\tau} \odot \boldsymbol{\sigma})$, and write $\ulcorner s \urcorner = s_1 \dots s_n$, consider $1 \leq j \leq n$ with s_j non-minimal. Necessarily, there is $u \in \mathscr{L}(\boldsymbol{\tau} \circledast \boldsymbol{\sigma})$ such that $u_\odot = s \in \Downarrow\Uparrow\text{-}\mathscr{L}(\boldsymbol{\tau} \odot \boldsymbol{\sigma})$. By Lemma 11.1.11, we have $u \in \Downarrow\Uparrow\text{-}\mathscr{L}(\boldsymbol{\tau} \circledast \boldsymbol{\sigma})$. Write $\ulcorner u \urcorner = u_1 \dots u_m$ – from the definition it is immediate that $\ulcorner u \urcorner_\odot = \ulcorner s \urcorner$, so there is a unique j' such that $u_{j'} = s_j$ (necessarily non-minimal). By Lemma 11.1.13, there is some $1 \leq i' < j'$ such that $\mathrm{j}(u_{j'}) = u_{i'}$. First, $u_{i'}$ may occur in B (if $u_{j'}$ is minimal in A and $u_{i'}$ minimal in B). Then by Lemma 11.1.13, there is $1 \leq k' < i'$ such that $\mathrm{j}(u_{i'}) = u_{k'}$, necessarily minimal in C. But then $\mathrm{j}(u_{j'}) = (u_{k'})$ for $\boldsymbol{\tau} \odot \boldsymbol{\sigma}$ – but $u_{k'} = s_k$ for some $1 \leq k < j$, and the property is proved. If $u_{i'}$ does not occur in B, then it is visible and $\mathrm{j}(u_{j'}) = u_{i'}$ in $\boldsymbol{\tau} \odot \boldsymbol{\sigma}$ as well. As above $u_{i'} = s_i$ for some $1 \leq i < j$, which concludes the proof. □

Putting everything together, we have:

Corollary 11.1.15. *The $\sim$-category* **NTCG** *admits a lluf sub-$\sim$-category* **NTCG-Seq** *with morphisms restricted to causal sequential strategies.*

Proof. Firstly, copycat is sequential by Lemma 11.1.6. Consider $\sigma \in \mathbf{NTCG}(A, B)$ and $\tau \in \mathbf{NTCG}(B, C)$ sequential. Firstly, the proof that $\tau \odot \sigma \in \mathbf{NTCG}(A, C)$ is *pointed* is as in Proposition 10.2.6 (which did not rely on visibility). Composition of *sequential determinism* and *sequential visibility* are Propositions 11.1.9 and 11.1.14. □

Next, we move to the rest of the categorical structure.

11.1.3 A Relative Seely $\sim$-Category

All structural morphisms of **NTCG** are obtained by lifting, and it is immediate (as for Lemma 11.1.6) that they are sequential. The next step is to prove that other operations on strategies (tensor, pairing, currying and promotion) preserve sequentiality. We shall consider them all, focusing on the two non-trivial cases: tensor and promotion.

Tensor. Consider $\sigma : A \vdash B$ and $\tau : C \vdash D$. Similarly to interactions, if $m \in \sigma \otimes \tau$, write $m_\sigma \in \sigma$ as m' if $m = (1, m')$ and undefined otherwise; and $m_\tau \in \tau$ partially defined symmetrically. For $s \in \mathscr{L}(\sigma \otimes \tau)$, write $s_\sigma \in \mathscr{L}(\sigma)$ and $s_\tau \in \mathscr{L}(\tau)$ defined in the obvious way. Using these projections, the stability of sequentiality under tensor exploits a state case analysis, similar to Lemma 11.1.8:

Lemma 11.1.16. *Take A, B, C, D $-$-boards, $\sigma : A \vdash B$, $\tau : C \vdash D$ sequential.*
If $s \in \Downarrow\Uparrow\text{-}\mathscr{L}(\sigma \otimes \tau)$, then $s_\sigma \in \Downarrow\Uparrow\text{-}\mathscr{L}(\sigma)$, $s_\tau \in \Downarrow\Uparrow\text{-}\mathscr{L}(\tau)$. Moreover, one of

(1) *s_σ, s_τ, s are respectively in state O, O, O,*
(2) *s_σ, s_τ, s are respectively in state O, P, P,*
(3) *s_σ, s_τ, s are respectively in state P, O, P,*

must hold.

Proof. Straightforward by induction on s, using reachable sequentiality of σ and τ. □

Thus, in a tensor of sequential strategies $\sigma \otimes \tau$, at most one of σ and τ has control at a given point in time, and only Opponent is able to switch. From this, we may prove:

Proposition 11.1.17. *Take A, B, C, D $-$-boards, $\sigma : A \vdash B$, $\tau : C \vdash D$ sequential.*
Then, $\sigma \otimes \tau$ is sequential.

Proof. Pointed: this is straightforward, as the event structure $\sigma \otimes \tau$ is simply $\sigma \parallel \tau$.

Sequential determinism: Consider $sm^+, sn^+ \in \Downarrow\Uparrow\text{-}\mathscr{L}(\sigma \otimes \tau)$. By Lemma 11.1.16, s_σ, s_τ, s must be in one of the states OOO, OPP, or POP. If it is OOO, then s is in state O, contradicting $sm^+ \in \Downarrow\Uparrow\text{-}\mathscr{L}(\sigma \otimes \tau)$. If it is OPP, then m^+_τ, n^+_τ must be defined, or contradict that s_σ is in state O – but then, $(sm^+)_\tau = s_\tau m^+_\tau \in \mathscr{L}(\tau)$, and $(sn)_\tau = s_\tau n^+_\tau \in \mathscr{L}(\tau)$. As $s_\tau \in \Downarrow\Uparrow\text{-}\mathscr{L}(\tau)$ is in state P, thus $s_\tau m^+_\tau, s_\tau n^+_\tau \in \Downarrow\Uparrow\text{-}\mathscr{L}(\tau)$, hence $m^+_\tau = n^+_\tau$ by *sequential determinism* of τ. Hence, $m^+ = n^+$ as required. Finally, if s is in state POP,

the reasoning is symmetric.

Sequential visibility: Consider $s \in \Updownarrow\text{-}\mathscr{L}(\sigma \otimes \tau)$. By Lemma 11.1.16, we have $s_\sigma \in \Updownarrow\text{-}\mathscr{L}(\sigma)$ and $s_\tau \in \Updownarrow\text{-}\mathscr{L}(\tau)$. Moreover, it is immediate from Lemma 11.1.16 that only Opponent may switch between σ and τ, and consequently $\ulcorner s \urcorner$ is entirely in σ or in τ. Hence, sequential visibility of σ and τ allow us to conclude. □

Promotion. The proof for promotion is analogous. Consider A and B strict $-$-boards, and $\sigma : !A \vdash B$ sequential. Recall that every move $m \in \sigma^!$ has the form $m = (i, m')$ where $i \in \mathbb{N}$ is its *copy index*. For $j \in \mathbb{N}$ we then partially define $m_j = m'$ if $i = j$ and undefined otherwise. Again, if $s \in \mathscr{L}(\sigma \otimes \tau)$, then its projection s_j for $i \in \mathbb{N}$ is defined in the obvious way. Using these, we state the promotion state case analysis:

Lemma 11.1.18. *Consider A, B strict $-$-boards, and $\sigma : !A \vdash B$ sequential.*
For any $s \in \Updownarrow\text{-}\mathscr{L}(\sigma^!)$, for any $i \in \mathbb{N}$, $s_i \in \Updownarrow\text{-}\mathscr{L}(\sigma)$ and we are in one of:

(1) *s has state O, and for all $i \in \mathbb{N}$, s_i has state O,*
(2) *s has state P, and there exists a unique $i \in \mathbb{N}$ such that s_i has state P.*

Proof. Straightforward by induction on s, using reachable sequentiality of σ. □

As for tensor, we conclude:

Proposition 11.1.19. *Consider A, B strict $-$-boards, and $\sigma : !A \vdash B$ sequential.*
Then, $\sigma^! : !A \vdash !B$ is a sequential strategy.

Proof. Similar to the proof of Proposition 11.1.17. □

Remainer of the structure. That pairing preserves sequentiality is straightforward; so is currying as the internal event structure is unchanged (along with the justifier function j(−)). Altogether, we have the ingredients to conclude the relative Seely structure:

Theorem 11.1.20. *There is* **NTCG**-**Seq**, *a relative Seely sub-$\sim$-category of* **NTCG**, *with morphisms restricted to sequential causal strategies.*

Proof. Follows from Corollary 11.1.15, Propositions 11.1.17 and 11.1.19, and omitted direct verifications for pairing, currying and structural isomorphisms. □

11.1.4 $\multimap\!\triangleright$-Seq and Interpretation of IA

As a relative Seely $\sim$-category, **NTCG**-**Seq** induces a cartesian closed $\sim$-category:

Definition 11.1.21. *There is $\multimap\!\triangleright$-**Seq**, a cartesian closed sub-$\sim$-category of $\multimap\!\triangleright$-**Strat**.*
*Its objects are mixed boards, its morphisms are those of $\multimap\!\triangleright$-**Strat** restricted to sequential strategies, and the equivalence relation is as in $\multimap\!\triangleright$-**Strat**.*

For stateless primitives (*i.e.* PCF primitives, reference and semaphore queries, bad variables and semaphores), it is direct by inspection to show that the corresponding strategies are sequential – we omit those verifications. Finally, we also have:

Proposition 11.1.22. *Consider $\sigma \in \multimap\triangleright$-**Seq**($\mathbf{V}, A$) sequential with A well-opened.*
*Then, $\sigma \odot \mathsf{cell} \in \multimap\triangleright$-**Strat**($\top, A$) is also sequential.*

Proof. Although cell is not a strategy (it fails *courtesy*), it does satisfy conditions *sequential determinism* and *sequential visibility* (the latter involves the notion of *justifier* which is defined in Definition 10.2.1 only for *pointed* strategies, but this is needed only for strategies from A to B – it is not necessary for $\mathsf{cell} : \mathbf{V}$). It follows from the same arguments as in Proposition 11.1.9 and 11.1.14 that $\sigma \odot \mathsf{cell}$ satisfies these two conditions – it is also obvious from A well-opened that $\sigma \odot \mathsf{cell}$ is pointed. □

The exact same reasoning establishes that for $\sigma \in \multimap\triangleright$-**Seq**($\mathbf{S}, A$) with A well-opened, $\sigma \odot \mathsf{lock} : A$ is sequential as well. Altogether we have the ingredients for:

Corollary 11.1.23. *The interpretation of $\mathsf{IA}_{/\!/}$ in $\multimap\triangleright$-**Strat** of Chapter 9 restricts to an adequate interpretation of IA in $\multimap\triangleright$-**Seq**.*

Proof. That the interpretation of IA terms yields sequential strategies follows from Theorem 11.1.20, verifications for basic primitives, Proposition 11.1.22 for new references, and the analogous for new semaphores. Adequacy is an immediate consequence of adequacy of the adequacy of the interpretation of $\mathsf{IA}_{/\!/}$ in $\mathbf{NTCG}_!$ (Theorem 9.1.5). □

11.2 The Alternating Unfolding

Now that we have constructed $\multimap\triangleright$-**Seq** and proved that the interpretation of IA indeed yields sequential causal strategies, we examine the unfolding of $\multimap\triangleright$-**Seq** onto $\downuparrows$-**Strat**.

11.2.1 Alternating Unfolding on a Mixed Board

The alternating unfolding of a sequential strategy is a simple refinement of the non-alternating unfolding of Section 9.2 – recall that for causal $\sigma : A$ on mixed board A, its *unfolding* was obtained by first considering all linearizations of σ, then displaying these linearizations to the game, and finally recovering plays with pointers by pointifixion.

We now adapt this to the alternating case.

Alternating linearizations. Recall from Definition 11.1.2 the set $\downuparrows$-$\mathscr{L}(E)$ of the alternating linearizations of an event structure with polarities E, which applies to a causal strategy $\sigma : A$ thanks to the polarities inherited from A. Our first observation is:

Lemma 11.2.1. *Consider $\sigma : A$ a causal strategy on a board A.*
The length-preserving monotone function of Lemma 9.2.4 restricts to

$$\begin{array}{rcccl} \partial_\sigma & : & \downuparrows\text{-}\mathscr{L}(\sigma) & \to & \downuparrows\text{-}\mathscr{L}(A) = \downuparrows\text{-}\mathsf{Plays}(A) \\ & & \varepsilon & \mapsto & \varepsilon \\ & & sm & \mapsto & \partial_\sigma(s)\partial_\sigma(m)\,. \end{array}$$

Proof. Straightforward. □

We recall our convention in such a situation: alternating linearizations in σ are often denoted as $s^\sigma \in \Updownarrow\uparrow\text{-}\mathscr{L}(\sigma)$ with the causal strategy added as superscript, and their display is $s^\sigma_A = \partial_\sigma(s^\sigma) \in \Updownarrow\uparrow\text{-}\mathsf{Plays}(A)$. As in the non-alternating case, this displayed linearization enjoys the main properties expected of what would be a play-based alternating strategy on A – to prove that, the following lemma shall be useful:

Lemma 11.2.2. *Consider A a board, and $\sigma : A$ a sequential causal strategy. For any $s \in \Updownarrow\uparrow\text{-}\mathsf{Plays}(\sigma)$, there is a* unique *$t \in \Updownarrow\uparrow\text{-}\mathscr{L}(\sigma)$ such that $s = \partial_\sigma(t)$.*

Proof. Immediate by induction, using *receptivity* and *sequential determinism*. □

We obtain a strategy in the sense of *simple games* (Definition 3.1.2):

Proposition 11.2.3. *Consider $\sigma : A$ a sequential strategy on board A. Then, the set*

$$\Updownarrow\uparrow\text{-}\mathsf{Plays}(\sigma) = \partial_\sigma(\Updownarrow\uparrow\text{-}\mathscr{L}(\sigma))$$

satisfies the following properties:

non-empty:	$\varepsilon \in \Updownarrow\uparrow\text{-}\mathsf{Plays}(\sigma)$,
prefix-closed:	*for all $t \in \Updownarrow\uparrow\text{-}\mathsf{Plays}(\sigma)$, is $s \sqsubseteq t$, then $s \in \circlearrowleft\text{-}\mathsf{Plays}(\sigma)$,*
receptive:	*if $s \in \Updownarrow\uparrow\text{-}\mathsf{Plays}(\sigma)$ and $sa^- \in \Updownarrow\uparrow\text{-}\mathsf{Plays}(A)$, then $sa \in \Updownarrow\uparrow\text{-}\mathsf{Plays}(\sigma)$,*
deterministic:	*if $sa^+, sb^+ \in \Updownarrow\uparrow\text{-}\mathsf{Plays}(\sigma)$, then $a^+ = b^+$,*

Proof. Non-empty, prefix-closed, receptive. Same proof as for Proposition 9.2.5.
Deterministic. Immediate by Lemma 11.2.2 and sequential determinism. □

With the appropriate phrasing it also follows that $\Updownarrow\uparrow\text{-}\mathsf{Plays}(\sigma)$ is P-visible, but we have not defined the P-view of alternating plays on A and it does not seem worth the detour. Thus, following the non-alternating case, we shall now apply the above to a sequential strategy on a mixed board, and follow it up with pointifixion.

Alternating linearizations up to symmetry. The above proposition does not say anything about symmetry. Recall that we proved in Proposition 7.1.15 that for A a gws (hence, in particular, for A a board), the set of plays $\Updownarrow\uparrow\text{-}\mathsf{Plays}(A)$ equipped with the equivalence relation $\cong_A$ (from Definition 7.1.14) is automatically an AJM game. Accordingly, we expect that $\Updownarrow\uparrow\text{-}\mathsf{Plays}(\sigma)$ is *uniform* in the sense of AJM games – this is natural as a consistency check, but it shall also be required later on.

To prove this, the first step is a symmetry-aware version of Lemma 11.2.2. Here for $s^\sigma, t^\sigma \in \Updownarrow\uparrow\text{-}\mathscr{L}(\sigma)$, we write $s^\sigma \cong_\sigma t^\sigma$ for the equivalence relation defined in exactly the same way as the AJM equivalence on plays on gws (Definition 7.1.14).

Lemma 11.2.4. *Consider A a $-$-board, $\sigma : A$ a sequential causal strategy. For all $s^\sigma, t^\sigma \in \Updownarrow\uparrow\text{-}\mathscr{L}(\sigma)$, if $s^\sigma_A \cong_A t^\sigma_A$ then $s^\sigma \cong_\sigma t^\sigma$.*

Proof. By induction on the length of linearizations – note from $s^\sigma_A \cong_A t^\sigma_A$ that s^σ and t^σ have the same length. If they are both empty then there is nothing to prove.

Consider $s^\sigma m^- \in \Downarrow\uparrow\text{-}\mathscr{L}(\sigma)$ displaying to $s^\sigma_A a^- \in \Downarrow\uparrow\text{-}\mathsf{Plays}(\sigma)$, and $t^\sigma n^- \in \Downarrow\uparrow\text{-}\mathscr{L}(\sigma)$ displaying to $t^\sigma_A b^- \in \Downarrow\uparrow\text{-}\mathsf{Plays}(\sigma)$, s.t. $s^\sigma_A a^- \cong_A t^\sigma_A b^-$. By IH, we have $s^\sigma \cong_\sigma t^\sigma$. But then it immediately follows by $\sim$-*receptivity* of σ that $s^\sigma m^- \cong_\sigma t^\sigma n^-$ as required.

Now, consider $s^\sigma m^+ \in \Downarrow\uparrow\text{-}\mathscr{L}(\sigma)$ displaying to $s^\sigma_A a^+ \in \Downarrow\uparrow\text{-}\mathsf{Plays}(\sigma)$, and $t^\sigma n^+ \in \Downarrow\uparrow\text{-}\mathscr{L}(\sigma)$ displaying to $t^\sigma_A b^+ \in \Downarrow\uparrow\text{-}\mathsf{Plays}(\sigma)$, such that $s^\sigma_A a^+ \cong_A t^\sigma_A b^+$. By induction hypothesis, we have $s^\sigma \cong_\sigma t^\sigma$. Now, by *extension* for $\cong_\sigma$, we must have $t^\sigma p^+ \in \Downarrow\uparrow\text{-}\mathscr{L}(\sigma)$ such that $s^\sigma m^+ \cong_\sigma s^\tau p^+$. But by *sequential determinism*, it follows that $n^+ = p^+$, hence we do have $s^\sigma m^+ \cong_\sigma t^\sigma n^+$ as required. □

The following lemma has two purposes: it will ensure that strategies obtained as displayed linearizations are indeed uniform in the sense of AJM strategies, and that two positively isomorphic sequential strategies will yield equivalent AJM strategies.

Lemma 11.2.5. *Consider A a $-$-board and $\sigma, \tau : A$ sequential causal strategies.*
If $\sigma \approx \tau$ (as in Section 7.2.1), then $\Downarrow\uparrow\text{-}\mathsf{Plays}(\sigma) \approx \Downarrow\uparrow\text{-}\mathsf{Plays}(\tau)$ (as in Definition 3.4.4).

Proof. If $\sigma \approx \tau$, by Definition 7.2.4 there is an iso $\varphi : \sigma \cong \tau$ of ess satisfying

$$\partial_\tau \circ \varphi \sim^+ \partial_\sigma ,$$

which in particular entails that for all $s^\sigma \in \Downarrow\uparrow\text{-}\mathscr{L}(\sigma)$, we have $s^\sigma_A \cong_A \varphi(s^\sigma)_A$. Combined with Lemma 11.2.4, this has an important direct consequence: if $s^\sigma \in \Downarrow\uparrow\text{-}\mathscr{L}(\sigma)$ and $s^\tau \in \Downarrow\uparrow\text{-}\mathscr{L}(\tau)$ are such that $s^\sigma_A \cong_A s^\tau_A$, then it also follows that $\varphi(s^\sigma)_A \cong_A s^\tau_A$, and consequently $\varphi(s^\sigma) \cong_\tau s^\tau$ by Lemma 11.2.4 – we shall use that in the proof below.

$\rightarrow$-*simulation:* consider $s^\sigma_A a^+ \in \Downarrow\uparrow\text{-}\mathsf{Plays}(\sigma)$, $s^\tau_A \in \Downarrow\uparrow\text{-}\mathsf{Plays}(\tau)$ such that $s^\sigma_A \cong_A s^\tau_A$. By our auxiliary statement, we necessarily have $\varphi(s^\sigma) \cong_\tau s^\tau$. Now writing $s^\sigma m^+ \in \Downarrow\uparrow\text{-}\mathscr{L}(\sigma)$ the witness for $s^\sigma_A a^+ \in \Downarrow\uparrow\text{-}\mathsf{Plays}(\sigma)$, we have $\varphi(s^\sigma)\varphi(m^+) \in \Downarrow\uparrow\text{-}\mathscr{L}(\tau)$ with $\varphi(s^\sigma) \cong_\tau s^\tau$, hence by *extension* for τ there is n^+ such that $\varphi(s^\sigma)\varphi(m^+) \cong_\tau s^\tau n^+$, with

$$s^\sigma_A a^+ \cong_A \partial_\sigma(s^\sigma m^+) \cong_A \partial_\tau(s^\tau n^+) = s^\tau_A b^+ \in \Downarrow\uparrow\text{-}\mathsf{Plays}(\tau)$$

as required. The case for $\leftarrow$-simulation is symmetric.

$\rightarrow$-*receptive:* consider $s^\sigma_A a^- \in \Downarrow\uparrow\text{-}\mathsf{Plays}(\sigma)$, $s^\tau_A \in \Downarrow\uparrow\text{-}\mathsf{Plays}(\tau)$, and $s^\sigma_A a^- \cong_A s^\tau_A b^-$. By the observation above, $\varphi(s^\sigma) \cong_\tau s^\tau$. Write $s^\sigma m^- \in \Downarrow\uparrow\text{-}\mathscr{L}(\sigma)$ such that $\partial_\sigma(s^\sigma m^-) = s^\sigma_A a^-$, and write $\varphi(m^-) = n^-$. Then $\varphi(s^\sigma)n^- \in \Downarrow\uparrow\text{-}\mathscr{L}(\tau)$ displayed to $\varphi(s^\sigma)_A c^-$, and

$$\varphi(s^\sigma)_A c^- \cong_A s^\sigma_A a^- \cong_A s^\tau_A b^- ,$$

where $\varphi(s^\sigma) \cong_\tau s^\tau$, hence by $\sim$-*receptivity* of τ there must be a unique extension

$$\varphi(s^\sigma)n^- \cong_\tau s^\tau p^-$$

such that $\partial_\tau(s^\tau p^-) = s^\tau_A b^-$. Finally, $\leftarrow$-receptive is symmetric. □

In particular, we obtain as a direct corollary:

Corollary 11.2.6. *If $\sigma : A$ is a sequential strategy, then $\Downarrow\uparrow\text{-}\mathsf{Plays}(\sigma)$ is an AJM strategy.*

Proof. From Proposition 11.2.3 we lack *uniformity*, obvious from Lemma 11.2.5. □

However, our goal is to unfold to strategies in HO games rather than AJM games.

Pointifixion of Alternating Plays From a sequential causal strategy, we wish to obtain an alternating strategy in the sense of Chapter 3; but those play on *arenas* rather than *boards*. Hence the situation is exactly the same as in Section 9.2: it is unclear how to automatically obtain an arena in the sence of Chapter 3 from a board, hence we unfold sequential causal strategies playing on *mixed boards* in the sense of Definition 9.2.6 – comprising the data of the board, the arena, along with the link between the two.

As in Corollary 9.2.11 for the non-alternating case, we have:

Proposition 11.2.7. *Consider A a mixed board. Then, $(\overset{\frown}{-})$ restricts to bijections*

$$\begin{array}{rcccl} (\overset{\frown}{-}) & : & \Updownarrow\uparrow\text{-}\mathsf{Plays}(!A)/\cong_{!A} & \simeq & \Updownarrow\uparrow\text{-}\mathsf{Pla}\overset{\frown}{\mathsf{y}}\mathsf{s}(\underline{A})\,, \\ (\overset{\frown}{-}) & : & \Updownarrow\uparrow\text{-}\mathsf{Plays}(A)/\cong_{A} & \simeq & \Updownarrow\uparrow\text{-}\mathsf{Pla}\overset{\frown}{\mathsf{y}}\mathsf{s}_{\bullet}(\underline{A})\,, \end{array}$$

preserving length and prefix.

Proof. Simply the restriction of Corollary 9.2.11 to alternating plays. □

Corollary 9.2.11 stated a similar bijection between $\circlearrowleft\text{-}\mathsf{Plays}(!A)$ up to symmetry and *pre-plays* $\circlearrowleft\text{-}\mathsf{PrePla}\overset{\frown}{\mathsf{y}}\mathsf{s}(\underline{A})$. The mismatch with the above is because while this bijection does not account for any well-bracketing constraint, non-alternating plays with pointers were defined as always logically well-bracketed (Definition 5.1.3) whereas alternating plays have no such hardwired condition[3] (Definition 3.1.4).

11.2.2 Alternating Unfoldings of Sequential Strategies

Now, we are in position to define the *alternating unfolding* of a sequential causal strategy. As for non-alternating strategies, we shall actually define *three* unfoldings:

(1) $\Updownarrow\uparrow\text{-}\mathsf{Unf}_{\bullet}$ unfolds $\sigma : A$ to an alternating thread-strategy on $\underline{A}$,
(2) $\Updownarrow\uparrow\text{-}\mathsf{Unf}_{!}$ unfolds $\sigma : !A$ to an alternating strategy on $\underline{A}$,
(3) $\Updownarrow\uparrow\text{-}\mathsf{Unf}$ unfolds $\sigma : !A \vdash B$ to an alternating strategy on $\underline{A} \Rightarrow \underline{B}$,

for all mixed boards A and B.

Unfolding to a thread-strategy. First, we define the pointed alternating unfolding of a sequential causal strategy on a mixed board A. The definition is essentially the same as Definition 9.2.12 for the non-alternating unfolding of causal strategies:

Definition 11.2.8. *Consider A a mixed board, and $\sigma : A$ a sequential causal strategy. The **unfolding** of σ is the set* $\Updownarrow\uparrow\text{-}\mathsf{Unf}_{\bullet}(\sigma) = \{\overset{\frown}{(\cdot)}(s^{\sigma}_{A}) \in \Updownarrow\uparrow\text{-}\mathsf{Pla}\overset{\frown}{\mathsf{y}}\mathsf{s}(\underline{A}) \mid s^{\sigma} \in \mathscr{L}(\sigma)\}$.

[3] In general, we consider it better to impose conditions on strategies rather than on plays whenever possible, so as to not unnecessarily constrain the behaviour of the execution environment. But for non-alternating strategies we were brought to assume all plays to be logically well-bracketing, as the non-alternating strategies for basic $\mathsf{IA}_{/\!/}$ primitives are hard to define rigourously otherwise: they have too many plays!

We show that this is indeed a valid alternating thread-strategy. With respect to Proposition 9.2.13 in the non-alternating case, the difficulty is to prove determinism: this puts together *(1)* sequential determinism, *(2)* Proposition 11.2.7 ensuring that alternating plays with pointers on $\underline{A}$ exactly represent alternating plays on A up to symmetry, and *(3)* Lemma 11.2.4 showing uniqueness of the witness in $\Updownarrow$-$\mathscr{L}(\sigma)$ for a plays in $\Updownarrow$-$\mathsf{Plays}(\sigma)$, up to symmetry. Putting those together, we get the proposition:

Proposition 11.2.9. *Consider $\sigma : A$ a causal strategy on mixed board A.*
Then, $\Updownarrow$-$\mathsf{Unf}_\bullet(\sigma) : \underline{A}$ is a P-visible thread-strategy in the sense of Definition 4.3.2.

Proof. It is a set of threads since A is *strict*. *Non-empty* and *prefix-closed* are direct.

Receptive: consider $t = \overset{\frown}{(-)}(s^\sigma_A)$ for $s^\sigma \in \Updownarrow$-$\mathscr{L}(\sigma)$, with $tb^- \in \Updownarrow$-$\mathsf{Plays}_\bullet(\underline{A})$. By Lemma 9.2.10, there is $s^\sigma_A a^- \in \circlearrowleft$-$\mathsf{Plays}(A)$ such that $\overset{\frown}{(-)}(s^\sigma_A a^-) = tb^-$; but by construction $s^\sigma_A a^-$ is alternating, thus in $\Updownarrow$-$\mathsf{Plays}(A)$. Now, by Proposition 11.2.3, there is $s^\sigma m \in \Updownarrow$-$\mathscr{L}(\sigma)$ displaying to $s^\sigma_A a^-$, and *receptive* follows.

Deterministic: Consider $sa^-b^+_1, sa^-b^+_2 \in \Updownarrow$-$\mathsf{Unf}_\bullet(\sigma)$, so there are $t^\sigma m^\sigma n^\sigma \in \Updownarrow$-$\mathscr{L}(\sigma)$ and $u^\sigma c^\sigma d^\sigma \in \Updownarrow$-$\mathscr{L}(\sigma)$ s.t. $sa^-b^+_1 = \overset{\frown}{(-)}(t^\sigma_A m^\sigma_A n^\sigma_A)$ and $sa^-b^+_2 = \overset{\frown}{(-)}(u^\sigma_A c^\sigma_A d^\sigma_A)$. Thus

$$t^\sigma m^\sigma_A n^\sigma_A \in \Updownarrow\text{-}\mathsf{Plays}(\sigma)\,, \qquad u^\sigma c^\sigma_A d^\sigma_A \in \Updownarrow\text{-}\mathsf{Plays}(\sigma)\,,$$

with $\overset{\frown}{(-)}(t^\sigma_A m^\sigma_A) = \overset{\frown}{(-)}(u^\sigma_A c^\sigma_A) = sa^-$, which entails $t^\sigma_A m^\sigma_A \cong_A u^\sigma_A c^\sigma_A$ by Proposition 11.2.7. But then, by Lemma 11.2.4, we must actually have

$$t^\sigma m^\sigma \cong_\sigma u^\sigma c^\sigma\,,$$

but hence by extension, there is $t^\sigma m^\sigma n^\sigma \cong_\sigma u^\sigma c^\sigma e^\sigma$. But by *sequential determinism* for σ, this means $e^\sigma = d^\sigma$, so that $t^\sigma m^\sigma n^\sigma \cong_\sigma u^\sigma c^\sigma d^\sigma$. Thus their display are also symmetric, so their pointifixion must be equal by Proposition 11.2.7.

P-visible: Straightforward by definition of sequential visibility. □

Unfolding to an alternating strategy. Next, we show how to unfold a sequential causal strategy to a full alternating strategy – again the definition closely mimics the non-alternating case, more specifically Definition 9.2.14

Definition 11.2.10. *Consider A a mixed board, and $\sigma : !A$ a sequential causal strategy.*
*The **unfolding** of σ is the set $\Updownarrow$-$\mathsf{Unf}_!(\sigma) = \{\overset{\frown}{(\cdot)}(s^\sigma_{!A}) \mid s^\sigma \in \Updownarrow\text{-}\mathscr{L}(\sigma)\}$.*

This is a well-formed alternating strategy on arena $\underline{A}$:

Proposition 11.2.11. *Consider A a mixed board and $\sigma : !A$ a sequential strategy.*
Then, $\Updownarrow$-$\mathsf{Unf}_!(\sigma) : \underline{A}$ is a P-visible alternating strategy as in Definition 3.1.5.

Proof. Same proof as for Proposition 11.2.9. □

As in the non-alternating case, if $\sigma : A$ is a sequential causal strategy, we may unfold it to a thread-strategy $\Updownarrow$-$\mathsf{Unf}_\bullet(\sigma) : \underline{A}$ and then obtain the corresponding single-threaded alternating strategy $\sigma^!$, or we may directly unfold its promotion as an alternating strategy with $\Updownarrow$-$\mathsf{Unf}_!(\sigma^!)$ – the two options coincide, as we show now.

Lemma 11.2.12. *Consider sequential causal $\sigma : A$, and $\sigma^! : !A$ its promotion.*
Then, $(\Updownarrow\uparrow\text{-}\mathsf{Unf}_\bullet(\sigma))^! = \Updownarrow\uparrow\text{-}\mathsf{Unf}_!(\sigma^!)$.

Proof. $\subseteq$. We actually show by induction on n that for all $s = s_1 \dots s_n \in (\Updownarrow\uparrow\text{-}\mathsf{Unf}_\bullet(\sigma))^!$, there is $t = t_1 \dots t_n \in \Updownarrow\uparrow\text{-}\mathscr{L}(\sigma^!)$ such that $s = (\overset{\frown}{-})(t_{!A})$ and for all prefix $s_1 \dots s_k$, writing

$$s_{\pi(1)} \dots s_{\pi(k)} = \lceil s_1 \dots s_k \rceil$$

the current thread of $s_1 \dots s_k$ for $\pi : \{1, \dots, p\} \to \{1, \dots, k\}$ some injection, we have $t_{\pi(1),\dots,\pi(p)} \in \Updownarrow\uparrow\text{-}\mathscr{L}(\sigma)$ and $s_{\pi(1)} \dots s_{\pi(k)} = (\overset{\frown}{-})(\partial_\sigma(t_{\pi(1)} \dots t_{\pi(p)}))$; the proof by induction is straightforward, relying on Lemma 9.2.10 and receptivity for negative extensions, on Lemma 11.2.4 and *extension* of isomorphism families for positive extensions.

$\supseteq$. Necessarily, any $sa^+ \in \Updownarrow\uparrow\text{-}\mathsf{Unf}_!(\sigma^!)$ comes from some $tm^+ \in \Updownarrow\uparrow\text{-}\mathscr{L}(\sigma^!)$, where m is in the component of $\sigma^!$ with copy index i. It is then straightforward that the moves of tm^+ with the same initial hereditary justifier as m^+ are exactly those events in copy index i, written $t'm^+ \in \Updownarrow\uparrow\text{-}\mathscr{L}(\sigma)$; and then we must have that

$$\lceil sa^+ \rceil = \lceil (\overset{\frown}{-})(\partial_{\sigma^!}(tm^+)) \rceil = (\overset{\frown}{-})(\partial_\sigma(t'm^+)) \in \Updownarrow\uparrow\text{-}\mathsf{Unf}_\bullet(\sigma)$$

as required, concluding the proof of the inclusion. □

Unfolding from mixed board A to mixed board B. As in the non-alternating case, we finish with the definition of the unfolding of a sequential causal strategy from one mixed board to another – again this must deal with the fact that morphisms from A to B in $\multimap\!\triangleright$-**Seq** are causal sequential strategies on $!A \vdash B$ (with at most one initial move) whereas morphisms in $\Updownarrow\uparrow$-**Strat** are alternating P-visible strategies on $\underline{A} \Rightarrow \underline{B}$, whose plays have arbitrarily many initial moves. Thus to deal with these differences, we set:

Definition 11.2.13. *Consider $\sigma : !A \vdash B$ a sequential strategy. Its **unfolding** is:*

$$\Updownarrow\uparrow\text{-}\mathsf{Unf}(\sigma) = \Updownarrow\uparrow\text{-}\mathsf{Unf}_!(\Lambda(\sigma)^!) : \underline{A} \Rightarrow \underline{B}\,.$$

It is a consequence of Proposition 11.2.11 that this is well-defined:

Proposition 11.2.14. *For any $\sigma \in \multimap\!\triangleright$-**Seq**$(A, B)$, we have $\Updownarrow\uparrow\text{-}\mathsf{Unf}(\sigma) \in \Updownarrow\uparrow$-**Strat**$(\underline{A}, \underline{B})$.*

Proof. In complement to Proposition 11.2.11, it remains to prove that $\Updownarrow\uparrow\text{-}\mathsf{Unf}(\sigma)$ is single-threaded. But by Lemma 11.2.12, we have $\Updownarrow\uparrow\text{-}\mathsf{Unf}(\sigma) = \Updownarrow\uparrow\text{-}\mathsf{Unf}_\bullet(\Lambda(\sigma))^!$ which is single-threaded by Proposition 4.3.3; this concludes the proof. □

11.2.3 Unfolding the Basic Categorical Structure

Next, we aim to show that the unfolding defined just above yields a $\sim$-functor

$$\Updownarrow\uparrow\text{-}\mathsf{Unf}(-) : \multimap\!\triangleright\text{-}\mathbf{Seq} \to \Updownarrow\uparrow\text{-}\mathbf{Strat}$$

which we will show preserves the interpretation of $\mathtt{IA}$.

Identities. First, we show that unfolding preserves identities. Of course, one must keep in mind that $\multimap\!\!\triangleright$-**Seq** is built on top of the Kleisli category $\mathbf{NTCG}_!$, so that the identity to be unfolded is actually dereliction $\mathbf{der}_A : !A \vdash A$.

Proposition 11.2.15. *Consider A a mixed board. Then,*

$$\Updownarrow\uparrow\text{-}\mathsf{Unf}(\mathbf{der}_A) = \Updownarrow\uparrow\text{-}\mathfrak{C}_{\underline{A}} \in \Updownarrow\uparrow\text{-}\mathbf{Strat}(\underline{A}, \underline{A})\,.$$

Proof. By Lemma 11.2.12 and Proposition 4.3.4 it suffices to show the equality

$$\Updownarrow\uparrow\text{-}\mathsf{Unf}_\bullet(\Lambda(\mathbf{der}_A)) = \Updownarrow\uparrow\text{-}\mathfrak{C}^\bullet_{\underline{A}}\,,$$

where $\Updownarrow\uparrow\text{-}\mathfrak{C}^\bullet_{\underline{A}}$ denotes the set of *threads* of $\Updownarrow\uparrow\text{-}\mathfrak{C}_{\underline{A}}$.

For $\subseteq$, consider $s \in \Updownarrow\uparrow\text{-}\mathsf{Unf}_\bullet(\Lambda(\mathbf{der}_A))$, meaning that $s = \hat{t}$ with $t = \partial_{\Lambda(\mathbf{der}_A)}(u)$ for some $u \in \Updownarrow\uparrow\text{-}\mathscr{L}(\mathfrak{C}_A)$. Recalling that $|\mathfrak{C}_A| = |A_1 \vdash A_2|$ with tags used for disambiguation, we observed in the proof of Lemma 11.1.6 that alternating linearizations on $\mathfrak{C}_A$ are exactly those $v \in |A_1 \vdash A_2|^*$ such that for all event-length $w \sqsubseteq v$, we have $w \upharpoonright A_1 = w \upharpoonright A_2$. Now any even-length $s' \sqsubseteq s$ comes from $w \sqsubseteq u$ an even-length prefix of u on which this observation applies, and it is direct that $s' \upharpoonright \underline{A}_1 = s' \upharpoonright \underline{A}_2$.

For $\supseteq$, consider $s \in \Updownarrow\uparrow\text{-}\mathfrak{C}^\bullet_{\underline{A}}$, assuming first that it has even length – hence $s \upharpoonright \underline{A}_1 = s \upharpoonright \underline{A}_2$. By Lemma 9.2.10, there is $t \in \Updownarrow\uparrow\text{-}\mathsf{Plays}(A)$ such that $\hat{t} = s \upharpoonright \underline{A}_1 = s \upharpoonright \underline{A}_2$. Interleaving two copies of t, it is straightforward to get $u \in \Updownarrow\uparrow\text{-}\mathsf{Plays}(A_1 \vdash A_2)$ such that $u \upharpoonright A_1 = u \upharpoonright A_2 = t$, and that for all even-length prefix $v \sqsubseteq u$, we have $v \upharpoonright A_1 = v \upharpoonright A_2$ as well, hence $u \in \Updownarrow\uparrow\text{-}\mathscr{L}(\mathfrak{C}_A)$ – it is then a direct verification that writing $w = \partial_{\Lambda(\mathbf{der}_A)}(u)$, we have $\hat{w} = s$ as required. Finally, if s has odd length, we use the above without the last move, and conclude by *receptivity* of $\Updownarrow\uparrow\text{-}\mathsf{Unf}_\bullet(\mathbf{der}_A)$. □

Composition. Here, the reasoning is pretty much the same as in the non-alternating case, except that plays and witnesses have to be ensured alternating – which follows easily from sequentiality. Thus we give fewer details than in the non-alternating case:

Proposition 11.2.16. *Consider $\sigma \in \multimap\!\!\triangleright\text{-}\mathbf{Seq}(A, B)$ and $\tau \in \multimap\!\!\triangleright\text{-}\mathbf{Seq}(B, C)$. Then:*

$$\Updownarrow\uparrow\text{-}\mathsf{Unf}(\tau \odot_! \sigma) = \Updownarrow\uparrow\text{-}\mathsf{Unf}(\tau) \odot \Updownarrow\uparrow\text{-}\mathsf{Unf}(\sigma)$$

Proof. First of all, we prove the inclusion

$$\Updownarrow\uparrow\text{-}\mathsf{Unf}_\bullet(\Lambda(\tau \odot \sigma^!)) \quad \subseteq \quad \Updownarrow\uparrow\text{-}\mathsf{Unf}(\tau) \odot \Updownarrow\uparrow\text{-}\mathsf{Unf}(\sigma)\,,$$

this follows by the same reasoning as in Lemma 9.3.4, additionally observing by induction that the interaction we obtain is in $\Updownarrow\uparrow\text{-}\mathsf{Unf}(\sigma) \parallel \Updownarrow\uparrow\text{-}\mathsf{Unf}(\tau)$ (in the sense of Definition 4.1.7) exploiting that σ and τ satisfy *reachable sequentiality* (Lemma 11.1.5).

Next, we show the following inclusion:

$$[\![\Updownarrow\uparrow\text{-}\mathsf{Unf}(\tau) \odot \Updownarrow\uparrow\text{-}\mathsf{Unf}(\sigma)]\!] \quad \subseteq \quad \Updownarrow\uparrow\text{-}\mathsf{Unf}_\bullet(\Lambda(\tau \odot \sigma^!))\,,$$

in this case this follows exactly as in 9.3.5, as the play for which we construct a witness in $\tau \odot \sigma^!$ is by hypothesis alternating.

From these two inclusions, the desired equality follows as in Corollary 9.3.6. □

Altogether, we have proved:

Theorem 11.2.17. *Unfolding yields a $\sim$-functor* $\Updownarrow\uparrow$-$\mathsf{Unf} : \multimap_{\triangleright}$-**Seq** $\to \Updownarrow\uparrow$-**Strat**.

Proof. The only thing remaining to check is that $\Updownarrow\uparrow$-Unf preserves equivalence, which is proved exactly as in Theorem 9.3.7. □

11.2.4 Unfolding the Interpretation of IA

As in the non-alternating case, it is an immediate verification that the alternating unfolding preserves the cartesian closed structure: preservation of projections is a variation of Proposition 11.2.15, it is a simple verification that unfolding preserves currying and its inverse, so that evaluation is preserved as well – altogether we get

$$\Updownarrow\uparrow\text{-}\mathsf{Unf}(-) : \multimap_{\triangleright}\text{-}\mathbf{Seq} \to \Updownarrow\uparrow\text{-}\mathbf{Strat}$$

a cartesian closed $\sim$-functor. The unfolding is continuous with respect to the dcpo structure on hom-sets, so that the fixpoint operator is preserved. Finally, it follows by inspection that the unfolding preserves the interpretation of all IA primitives, so that:

Theorem 11.2.18. *We have a cartesian closed $\sim$-functor:*

$$\Updownarrow\uparrow\text{-}\mathsf{Unf}(-) : \multimap_{\triangleright}\text{-}\mathbf{Seq} \to \Updownarrow\uparrow\text{-}\mathbf{Strat}\,,$$

preserving the interpretation in the sense that for $\Gamma \vdash M : A$ *any term of* IA*, we have*

$$\Updownarrow\uparrow\text{-}\mathsf{Unf}(\llbracket M \rrbracket_{\multimap_{\triangleright}\text{-}\mathbf{Seq}}) = \llbracket M \rrbracket_{\Updownarrow\uparrow\text{-}\mathbf{Strat}}\,.$$

Intensional full abstraction. We motivated the unfolding

$$\Updownarrow\uparrow\text{-}\mathsf{Unf}(-) : \multimap_{\triangleright}\text{-}\mathbf{Seq} \to \Updownarrow\uparrow\text{-}\mathbf{Strat}$$

as a way to show that $\multimap_{\triangleright}$-**Seq** is intensionally fully abstract for IA, *i.e.* to inherit in concurrent games the full abstraction result for IA in alternating game semantics (Theorem 3.3.5). However, full abstraction does not hold for $\Updownarrow\uparrow$-**Strat** but only with respect to *well-bracketed* P-visible alternating strategies, *i.e.* for $\Updownarrow\uparrow$-**WB**.

Strategies in $\multimap_{\triangleright}$-**Seq** already embark two notions of well-bracketing: the winning mechanism implemented in **NTCG** (see Section 8.2) already bans call/cc, and we also imported logical well-bracketing from non-alternating games in Definition 9.3.2 – so we may hope that the unfolding actually lifts to a cartesian closed $\sim$-functor

$$\Updownarrow\uparrow\text{-}\mathsf{Unf}(-) : \multimap_{\triangleright}\text{-}\mathbf{Seq} \to \Updownarrow\uparrow\text{-}\mathbf{WB}\,,$$

but that is not the case: we showed in Figure 10.18 a sequential causal strategy whose alternating unfolding is not well-bracketed. The strategy is winning: its $+$-covered configurations have a positive payoff. Its non-alternating unfolding is logically well-bracketed (this matches the counter-example to the fact that logical well-bracketing implies well-bracketing, seen in Figure 5.3). Nevertheless, the play following the long causal chain is not well-bracketed as the final move does not answer the pending question. However this does not prevent the interpretation of IA in $\multimap_{\triangleright}$-**Seq** to be intensionally fully abstract, because this phenomenon cannot occur in plays where the initial move has an answer.

Lemma 11.2.19. *Let $s \in \Updownarrow\uparrow\text{-}\mathsf{Plays}(\underline{A})$ be P- and O-visible. Assume that s has the form*

$$s = \ \ldots \quad s_i \ \overset{\cdots}{\cdots}\ s_j \quad \ldots$$

where no further move points to s_j. Then, no move after s_j can point within $s_i \ldots s_j$.

Proof. By P- or O-visibility, s_{j+1} points strictly before s_i. Then no view can ever see $s_{i+1} \ldots s_j$ – so no move can point there. Besides, s_i can only be seen by the player responsible for it, so no move can point to s_i.

A detailed proof appears in [Clairambault and Harmer, 2010, Lemma 5]. □

From this, we may easily deduce the following:

Lemma 11.2.20. *Consider $s \in \Updownarrow\uparrow\text{-}\mathsf{Plays}(\underline{A})$ O- and P-visible, and such that any Question has an answer. Then, s is well-bracketed in the sense of Definition 3.2.5.*

Proof. Assume s has a well-bracketing failure, *i.e.* as in:

$$s = \ \ldots \quad \mathbf{q}_1^{\mathcal{Q}} \ \overset{\cdots}{\cdots}\ \mathbf{q}_2^{\mathcal{Q}} \quad \ldots \quad a^{\mathcal{A}} \quad \ldots$$

with $\mathbf{q}_2$ unanswered when playing a. By *answer-closing*, no further move can point to a. Thus by Lemma 11.2.19, no further move can point to $\mathbf{q}_2$ which must therefore remain unanswered, contradicting the hypothesis that every question has an answer. □

From all this, we may now conclude:

Theorem 11.2.21. *The interpretation of* IA *in* $\multimap\!\triangleright$-**Seq** *is intensionally fully abstract.*

Proof. Let $\vdash M, N : A$ be terms in IA, and assume that $[\![M]\!] \not\approx [\![N]\!]$, *i.e.* there is a test $\boldsymbol{\alpha} \in \multimap\!\triangleright\text{-}\mathbf{Seq}([\![A]\!], [\![\mathbb{U}]\!])$ such that $\boldsymbol{\alpha} \odot_! [\![M]\!] \neq \boldsymbol{\alpha} \odot_! [\![N]\!]$ – assume *w.l.o.g.* that $\boldsymbol{\alpha} \odot_! [\![M]\!]$ converges while $\boldsymbol{\alpha} \odot_! [\![M]\!]$ diverges. Writing $\alpha' = \Updownarrow\uparrow\text{-}\mathsf{Unf}(\boldsymbol{\alpha})$, it follows that

$$\alpha' \odot [\![M]\!]_{\Updownarrow\uparrow\text{-}\mathbf{Strat}} \Downarrow \qquad\qquad \alpha' \odot [\![N]\!]_{\Updownarrow\uparrow\text{-}\mathbf{Strat}} \Uparrow,$$

since, by Theorem 11.2.18, the alternating unfolding preserves composition and the interpretation of IA. Here, α' is a P-visible alternating strategy, but because of the phenomena highlighted in Figure 10.18 it may not be well-bracketed as in Definition 3.2.6. Consider $s \in \alpha'$ involved in $\alpha' \odot_! [\![M]\!] \Downarrow$ – until the rest of the proof, $[\![M]\!]$ is the interpretation in $\Updownarrow\uparrow$-$\mathbf{Strat}$. The initial question of s has an answer. Moreover, $[\![M]\!]$ is the alternating restriction of $\circlearrowleft\text{-}\mathsf{Unf}([\![M]\!]_{\multimap\!\triangleright\text{-}\mathbf{Strat}})$ which is logically well-bracketed, and likewise α' is logically well-bracketed, thus s is logically well-bracketed and hence all its questions are answered. It is P-visible and O-visible since both $[\![M]\!]$ and α' are P-visible. Hence, by Lemma 11.2.20, it is well-bracketed as in Definition 3.2.6.

Consider α' restricted to prefixes of s. Now α' is well-bracketed as in Definition 3.2.5, and it distinguishes $[\![M]\!]$ and $[\![N]\!]$, hence $M \not\approx N$ by Theorem 4.3.11. □

So far we have seen that:

$\rightarrow\!\triangleright$-**Strat** is intensionally fully abstract for $\mathsf{IA}_{/\!/}$ (Theorem 9.3.10),
$\rightarrow\!\triangleright$-**Seq** is intensionally fully abstract for IA (Theorem 11.2.21).

It remains to prove two matching theorems in the presence of parallel innocence.

11.3 Sequential Innocence

In this section, we study the combination of sequentiality and parallel innocence, and establish links with traditional innocence as introduced in Section 3.2.

Let us start with the definition:

Definition 11.3.1. *A causal strategy $\sigma : A$ is **sequential innocent** if it is both* sequential *(as in Definition 11.1.4) and* parallel innocent *(as in Definition 10.1.11).*

As both sequential and parallel innocent causal strategies form a cartesian closed sub-$\sim$-category of $\rightarrow\!\triangleright$-**Strat**, there is a cartesian closed $\sim$-category $\rightarrow\!\triangleright$-**SeqInn** of sequential innocent strategies. As PCF may be interpreted in both $\rightarrow\!\triangleright$-**Seq** and $\rightarrow\!\triangleright$-**Inn**, it follows that it has an adequate interpretation in $\rightarrow\!\triangleright$-**SeqInn** as well. In this section we shall prove that this interpretation is intensionally fully abstract, by proving that $\Updownarrow$-**Strat**$(-)$ sends sequential innocent strategies to innocent alternating strategies as in Definition 3.2.9, and rely on Theorem 4.3.10 to deduce intensional full abstraction.

11.3.1 Causal Analysis of Sequential Innocence

Basic causal shape. So as to establish the link with traditional innocence, we first aim to understand better what constraints on the causal shape are brought by the combination of parallel innocence, and sequentiality. Roughly speaking, parallel innocence only allows Player to merge threads he spawned, whereas sequentiality prevents Player to spawn parallel threads: it follows that the causal shape of a sequential innocent strategy must be purely forestial, with branchings the entire responsibility of Opponent:

Lemma 11.3.2. *Consider A a board, and $\sigma : A$ a* sequential, parallel innocent *strategy. Then, σ is an O-branching alternating forest.*

Proof. First, we prove that for all $m \in \sigma$, its set of dependencies $[m]_\sigma$ is a total order.

Seeking a contradiction, take $m' \in \sigma$ minimal with $m' \rightarrow\!\triangleright_\sigma m_1$ and $m' \rightarrow\!\triangleright_\sigma m_2$ distinct, all within $[m]_\sigma$. By minimality, $[m']_\sigma$ is a total order, *i.e.* a gcc. By Lemma 6.1.16, m_1 and m_2 have the same polarity, opposite of m'. Consider $\rho_1 \in \mathsf{gcc}(\sigma)$ a gcc for m passing through $m' \rightarrow\!\triangleright_\sigma m_1$, and $\rho_2 \in \mathsf{gcc}(\sigma)$ a gcc for m passing through $m' \rightarrow\!\triangleright_\sigma m_2$. Then ρ_1 and ρ_2 have least distinct events m_1 and m_2; hence by pre-innocence m_1 and m_2 are positive. Now, m' must be the only immediate dependency of m_1, and the only immediate dependency of m_2; indeed if there was $m'' \rightarrow\!\triangleright_\sigma m_i$, then considering $\rho' \rightarrow\!\triangleright m_i \in \mathsf{gcc}(\sigma)$ passing through m'', ρ and ρ' would fork at some event smaller than

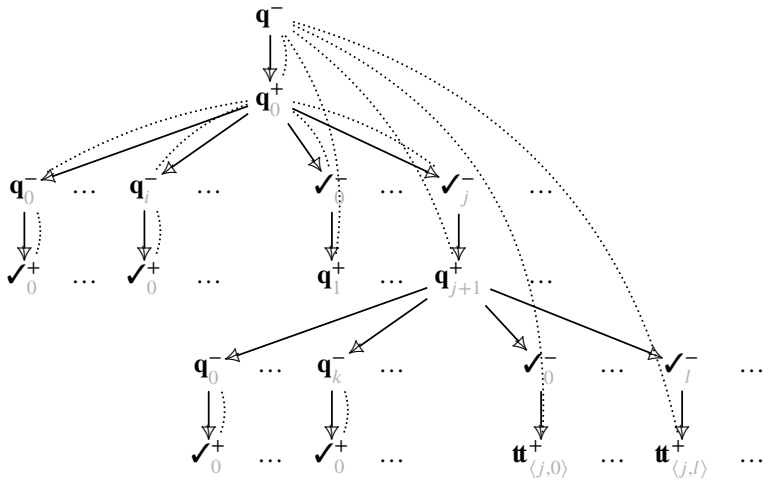


Figure 11.3: A sequential innocent causal strategy

m', contradicting its minimality. Hence, $[m'] \cup \{m_i\} \in \mathscr{C}(\sigma)$ for $i \in \{1,2\}$.

Also writing $[m']$ for the linearization in $\downarrow\uparrow$-$\mathscr{L}(\sigma)$ with events in the same order,

$$[m'],\ [m']m_1,\ [m']m_2\ \in\ \mathscr{L}(\sigma)\,,$$

but by Lemma 6.1.16, $[m']m_1^+$ and $[m']m_2^+$ are alternating. By *sequential determinism* of σ, it follows that $m_1 = m_2$, contradiction. So, for all $m \in \sigma$, $[m]_\sigma$ is a total order.

Thus $(|\sigma|, \leq_\sigma)$ is a forest. Likewise, if $m^- \rightarrowtriangle m_1^+$ and $m^- \rightarrowtriangle m_2^+$ in σ, by *sequential determinism* and the same reasoning as above, $m_1 = m_2$, so σ is O-branching. Finally, as for any causal strategy $\sigma : A$ on A alternating, we have $\rightarrowtriangle_\sigma$ is alternating as well. □

Description. The causal shape of a sequential innocent strategy is a *forest*, but it might not be obvious to the reader what this forest actually represents. Thus we start this discussion by looking at the interpretation as a causal strategy of the simple term

$$\vdash \lambda f^{\mathbb{U}\to\mathbb{U}}.\, f\,\mathbf{skip};\ f\,\mathbf{skip};\ \mathbf{tt} : (\mathbb{U}\to\mathbb{U})\to\mathbb{B}\,.$$

of pure PCF. The interpretation of this term yields the strategy in Figure 11.3.

This event structure is infinite, since Opponent may play any of his available moves an arbitrary number of times, using any natural number as copy index.

Intuitively, this diagram follows the structure of the term: once computation is initiated, Player asks for the output of f. Each time f calls its argument, Player answers $\checkmark^+$. Note though that unlike in traditional game semantics, the strategy shows that these calls are causally independent from each other. Likewise, the game permits Opponent to have f return multiple times. Each time this happens, this triggers another call to f, which may call its argument any number of times, and return any number of times.

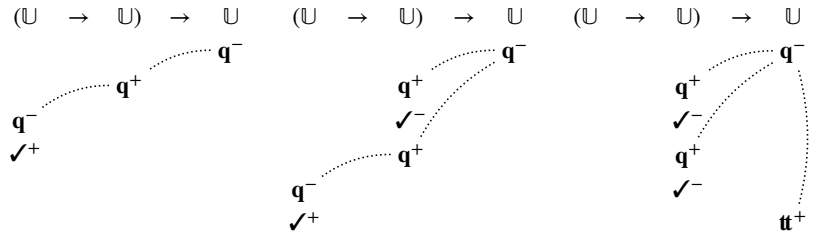
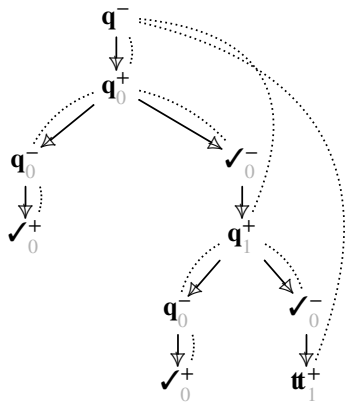


Figure 11.4: Maximal P-views of $\lambda f.\, f\ \mathbf{skip};\ f\ \mathbf{skip};\ \mathbf{tt}$

It seems clear from this diagram that the strategy contains a lot of redundant information: by symmetry, it intuitively suffices to know how the strategy behaves if Opponent performs each available action exactly once, say with copy index 0:



leading to what we shall call the *meager form* in general for parallel innocent strategies, in Section 12.1. Ignoring the copy indices we observe here that we have a tree with three maximal branches, which coincide with the three maximal P-views of the corresponding innocent alternating strategy in the traditional sense, as displayed in Figure 11.4. This plainly shows that P-views actually express the *causal* structures in traditional game semantics already, with however one main difference with respect to concurrent games: sets of P-views cannot express Opponent replications, so that composing them requires using plays, stepping out of the causal reprentation. In contrast, concurrent games remain causal also in the presence of Opponent replications.

Recovering P-views. Next we show that, as expected from the discussion above, the branches of a sequential innocent strategy do correspond to P-views.

Proposition 11.3.3. *Consider A a mixed board, and $\sigma : A$ sequential innocent. For all $\rho \in \mathsf{gcc}(\sigma)$, $(\overset{\frown}{-})(\partial_\sigma(\rho))$ is a P-view.*

Proof. By Lemma 11.3.2, σ is forestial, so that prefixes of ρ are down-closed and inform configurations. Hence, we actually have $\rho \in \mathscr{L}(\sigma)$. Moreover, by Lemma

6.1.16, immediate causality alternates in σ and in particular, $\rho \in \Downarrow\uparrow\text{-}\mathscr{L}(\sigma)$. In particular, $\partial_\sigma(\rho) \in \Downarrow\uparrow\text{-}\mathsf{Plays}(A)$ and hence $(\overset{\frown}{-})(\partial_\sigma(\rho)) \in \Downarrow\uparrow\text{-}\mathsf{Plays}(\underline{A})$.

To prove that it is a P-view, it remains to show that Opponent moves point to the previous move. But if we have $s = s_1 \dots s_i^+ s_{i+1}^- \cdots = (\overset{\frown}{-})(\partial_\sigma(\rho))$, those must come from

$$\rho = \rho_1 \rightarrowtriangle_\sigma \cdots \rightarrowtriangle_\sigma \rho_i^+ \rightarrowtriangle_\sigma \rho_{i+1}^- \rightarrowtriangle_\sigma \cdots ,$$

then $\partial_\sigma(\rho_i^+) \rightarrowtriangle_A \partial_\sigma(\rho_{i+1}^-)$ by *courtesy*, which entails that s_{i+1}^- points to s_i^+. □

Thus indeed, we may regard a sequential innocent strategy as a presentation of an expanded version of the forest of P-views, with an explicit choice of copy indices.

11.3.2 The Unfolding Preserves Innocence

We now aim to show that the unfolding of a sequential innocent strategy is innocent in the sense of traditional Hyland-Ong game semantics, as introduced in Chapter 3.

Alternating linearizations. In order to prove that, our first step will be to study alternating linearizations of sequential innocent strategies, in particular understanding how the mechanism of P-views captures the causal structure.

Recall the *P-view* of an alternating linearization, introduced in Definition 11.1.3.

Lemma 11.3.4. *Consider $\sigma \in \mathbf{NTCG}(A, B)$ sequential innocent.*
Then, for all $tm \in \Downarrow\uparrow\text{-}\mathscr{L}(\sigma)$, we have $\ulcorner tm \urcorner = [m]_\sigma$.

Proof. Here we treat $[m]_\sigma$ as the sequence induced by its total ordering.

The crucial observation is that if $tm^-n^+ \in \Downarrow\uparrow\text{-}\mathscr{L}(\sigma)$, then necessarily $m^- \rightarrowtriangle_\sigma n^+$. To prove that, we prove by induction that for any $t \in \Downarrow\uparrow\text{-}\mathscr{L}(\sigma)$: *(1)* if t has even length, then all maximal events of $|t| \in \mathscr{C}(\sigma)$ are positive; and *(2)* if t has odd length, then $|t| \in \mathscr{C}(\sigma)$ has *exactly one* maximal negative event. Indeed, for $tm^- \in \Downarrow\uparrow\text{-}\mathscr{L}(\sigma)$, then t has even length, so $|t|$ has all its maximal events positive. But then $|tm^-|$ has exactly one maximal negative event, namely m^-. Likewise, for $tm^+ \in \Downarrow\uparrow\text{-}\mathscr{L}(\sigma)$, then $|t|$ has exactly one maximal negative event. Now, the immediate predecessor of m must be negative. But if it is not maximal in $|t|$, this contradicts Lemma 11.3.2, and in particular that σ is O-branching. Therefore, the predecessor of m must be the unique maximal negative event of $|t|$, and $|tm|$ has all maximal events positive as required.

Now, if tm^-n^+, then $|tm^-|$ has exactly one maximal negative event (namely m^-); while the maximal events of $|tm^-n^+|$ are all positive (and comprise n^+). Hence, $m^- \rightarrowtriangle_\sigma n^+$ as required. Likewise, if $t_1m^+t_2n^- \in \Downarrow\uparrow\text{-}\mathscr{L}(\sigma)$ such that $\mathrm{j}(n) = m$ – so $\ulcorner t_1mt_2n \urcorner = \ulcorner t_1 \urcorner mn$ then we must have $\partial_\sigma(m) \rightarrowtriangle \partial_\sigma(n)$ hence $m \rightarrowtriangle_\sigma n$ by Lemma 6.1.16. From these two facts, the lemma is a direct verification by induction on t. □

We regard this as the fundamental reason why the mechanism of P-views works: on alternating linearizations, it recaptures exactly the causal structure explicitely carried by concurrent strategies – a strength of concurrent games is that beyond the alternating

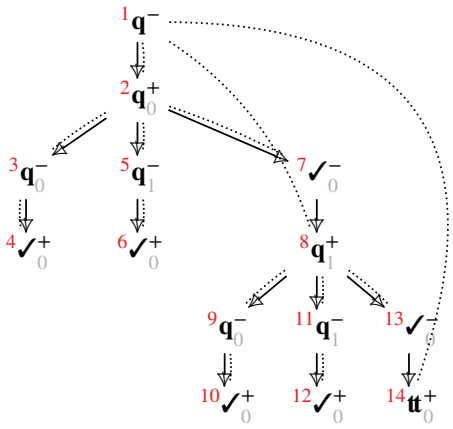


Figure 11.5: Witness for Figure 9.19

deterministic realm, P-views are no longer available while concurrent strategies remain.

As an illustration, Figure 11.5 shows the configuration explored in the play of

$$[\![\lambda f^{\mathbb{U}\to\mathbb{U}}.\, f\ \mathbf{skip};\ f\ \mathbf{skip};\ \mathbf{tt}]\!] : [\![(\mathbb{U}\to\mathbb{U})\to\mathbb{B}]\!]$$

in Figure 9.19 (note that this is another presentation of the diagram on the left hand side of Figure 10.12) – the numbers in red correspond to the order in which the linearization proceeds. As proved above, at each stage the P-view is exactly the branch leading to the corresponding move in the configuration. Opponent could explore the same configuration in a different order, corresponding to a different play visiting the same P-views. On the other hand, only Opponent has any degree of freedom in this exploration: Player has ever at most one possible move, that immediately caused by the last Opponent move[4].

Innocence of the unfolding. Now, we show that as expected, the alternating unfolding of a sequential innocent strategy is innocent in the traditional sense.

Proposition 11.3.5. *Consider A a mixed board, and $\sigma : A$ sequential innocent. Then, $\Updownarrow\uparrow\text{-}\mathsf{Unf}_!(\sigma^!) : \underline{A}$ is an innocent alternating strategy (as in Definition 3.2.9).*

Proof. We already know from Proposition 11.2.11 that $\Updownarrow\uparrow\text{-}\mathsf{Unf}_!(\sigma^!)$ is P-visible.

Consider $sa^-b^+, ta^- \in \Updownarrow\uparrow\text{-}\mathsf{Unf}(\sigma^!)$ s.t. $\ulcorner sa^- \urcorner = \ulcorner ta^- \urcorner$. By definition, there are witnesses $umn \in \Updownarrow\uparrow\text{-}\mathscr{L}(\sigma^!)$ for sa^-b^+ and $vo \in \Updownarrow\uparrow\text{-}\mathscr{L}(\sigma^!)$ for ta^-. By Lemma 11.3.4,

$$\ulcorner um \urcorner = [m]_{\sigma^!}, \qquad \ulcorner vo \urcorner = [o]_{\sigma^!}$$

[4]Different explorations of the same configuration may be related by permuting contiguous OP pairs of moves. Deterministic innocent strategies may be defined as those stable under the permutations of OP pairs permitted by the arena: this is the idea behind Melliès' presentation of innocence [Melliès, 2004a].

where $[m]_{\boldsymbol{\sigma}^!}, [o]_{\boldsymbol{\sigma}^!}$ are seen as the sequences induced by their total ordering. Those are gccs, and as in Proposition 11.3.3 they must be alternating, thus $\ulcorner um \urcorner, \ulcorner vo \urcorner \in \Updownarrow\text{-}\mathscr{L}(\boldsymbol{\sigma}^!)$.

Now, from $\ulcorner sa^- \urcorner = \ulcorner ta^- \urcorner$, by Proposition 11.2.7 we have that $\partial_{\boldsymbol{\sigma}^!} \ulcorner um \urcorner \cong_A \partial_{\boldsymbol{\sigma}^!} \ulcorner vo \urcorner$, which entails that $\ulcorner um \urcorner \cong_{\boldsymbol{\sigma}^!} \ulcorner vo \urcorner$ by Lemma 11.2.4. Now, as $\ulcorner um \urcorner$ extends with n, by *extension* there must be $\ulcorner vo \urcorner p = \ulcorner vop \urcorner \in \Updownarrow\text{-}\mathscr{L}(\boldsymbol{\sigma}^!)$ with $\ulcorner umn \urcorner \cong_{\boldsymbol{\sigma}} \ulcorner vop \urcorner$. Now it follows that $|vo| \cup \{p\} \in \mathscr{C}(\boldsymbol{\sigma}^!)$ by *determinism*, so that $vop \in \Updownarrow\text{-}\mathscr{L}(\boldsymbol{\sigma}^!)$. We may project it to $ta^- b^+ = (\overset{\frown}{-})(\partial_{\boldsymbol{\sigma}^!}(vop))$. Finally, since $\ulcorner umn \urcorner \cong_{\boldsymbol{\sigma}} \ulcorner vop \urcorner$ we must have

$$\partial_{\boldsymbol{\sigma}^!}(\ulcorner umn \urcorner) \cong_{!A} \partial_{\boldsymbol{\sigma}^!}(\ulcorner vop \urcorner)$$

hence $(\overset{\frown}{-})(\partial_{\boldsymbol{\sigma}^!}(\ulcorner umn \urcorner)) = (\overset{\frown}{-})(\partial_{\boldsymbol{\sigma}^!}(\ulcorner vop \urcorner))$, and since $\ulcorner sa^- b^+ \urcorner = \ulcorner (\overset{\frown}{-})(\partial_{\boldsymbol{\sigma}^!}(umn)) \urcorner = (\overset{\frown}{-})(\partial_{\boldsymbol{\sigma}^!}(\ulcorner umn \urcorner))$ and likewise for vop it follows that $\ulcorner sa^- b^+ \urcorner = \ulcorner ta^- b^+ \urcorner$. □

Of course, from the above we also have a characterization of its P-views:

Proposition 11.3.6. *Consider A a mixed board, and $\boldsymbol{\sigma} : A$ sequential innocent.*
The (non-empty) P-views of $\Updownarrow\text{-}\mathsf{Unf}_!(\boldsymbol{\sigma}^!)$ are exactly *those $(\overset{\frown}{-})(\partial_{\boldsymbol{\sigma}}([m]_{\boldsymbol{\sigma}}))$ for $m \in \boldsymbol{\sigma}$.*

Proof. If $m \in \boldsymbol{\sigma}$, then, picking some arbitrary copy index $i \in \mathbb{N}$, we have $(i, m) \in \boldsymbol{\sigma}^!$, and $[(i, m)]_{\boldsymbol{\sigma}^!} \in \Updownarrow\text{-}\mathscr{L}(\boldsymbol{\sigma}^!)$ thus $(\overset{\frown}{-})(\partial_{\boldsymbol{\sigma}^!}([(i, m)]_{\boldsymbol{\sigma}^!})) \in \Updownarrow\text{-}\mathsf{Unf}_!(\boldsymbol{\sigma}^!)$. Moreover, by Proposition 11.3.3, it is a P-view of $\Updownarrow\text{-}\mathsf{Unf}_!(\boldsymbol{\sigma}^!)$. Finally, $(\overset{\frown}{-})(\partial_{\boldsymbol{\sigma}^!}([(i, m)]_{\boldsymbol{\sigma}^!}) = (\overset{\frown}{-})(\partial_{\boldsymbol{\sigma}}([m]_{\boldsymbol{\sigma}}))$.

Reciprocally, consider $s \in \Updownarrow\text{-}\mathsf{Unf}_!(\boldsymbol{\sigma}^!)$ a non-empty P-view, *i.e.* we have $\ulcorner s \urcorner = s$. By definition, there is $tm \in \Updownarrow\text{-}\mathscr{L}(\boldsymbol{\sigma}^!)$ such that $s = (\overset{\frown}{-})(\partial_{\boldsymbol{\sigma}^!}(tm))$, with $\ulcorner tm \urcorner = tm$. By Lemma 11.3.4, $\ulcorner tm \urcorner = [m]_{\boldsymbol{\sigma}^!}$. But then, $m = (i, m')$ for some copy index $i \in \mathbb{N}$ and $(\overset{\frown}{-})(\partial_{\boldsymbol{\sigma}}([m']_{\boldsymbol{\sigma}})) = (\overset{\frown}{-})(\partial_{\boldsymbol{\sigma}^!}([m]_{\boldsymbol{\sigma}^!}) = s$ as required. □

We may now wrap up the unfolding to innocent alternating strategies:

Theorem 11.3.7. *We have a cartesian closed $\sim$-functor:*

$$\Updownarrow\text{-}\mathsf{Unf}(-) : \multimap\text{-}\mathbf{SeqInn} \to \Updownarrow\text{-}\mathbf{Inn}\,,$$

preserving the interpretation in the sense that for $\Gamma \vdash M : A$ any term of PCF*, we have*

$$\Updownarrow\text{-}\mathsf{Unf}(\llbracket M \rrbracket_{\multimap\text{-}\mathbf{SeqInn}}) = \llbracket M \rrbracket_{\Updownarrow\text{-}\mathbf{Inn}}\,.$$

Proof. Follows from Theorem 11.2.18 and Proposition 11.3.5. □

11.3.3 Sequential Globularity

We refine this in the presence of the further constraint of *globularity* (Definition 10.5.1).

Let us start with:

Definition 11.3.8. *A strategy $\boldsymbol{\sigma} \in \multimap\text{-}\mathbf{Strat}(A, B)$ is **sequential globular** if it is both sequential (as in Definition 11.1.4) and globular (as in Definition 10.5.1).*

As both sequential and globular strategies form a cartesian closed sub-$\sim$-category of $\multimap\triangleright$-**Strat**, there is a cartesian closed $\sim$-category $\multimap\triangleright$-**SeqGlob** of sequential globular strategies. As PCF may be interpreted in both $\multimap\triangleright$-**Seq** and $\multimap\triangleright$-**Glob**, it follows that it has an adequate interpretation in $\multimap\triangleright$-**SeqGlob** as well – here we shall prove that this interpretation is actually exactly the same as that in $\downarrow\uparrow$-**InnWB**.

We first show that the unfolding of a sequential globular strategy is well-bracketed:

Proposition 11.3.9. *Consider A and B mixed boards, and $\sigma \in \multimap\triangleright$-**SeqGlob**$(A, B)$. Then, $\downarrow\uparrow$-$\mathsf{Unf}(\sigma) \in \downarrow\uparrow$-**Inn**$(\underline{A}, \underline{B})$ is well-bracketed (in the sense of Definition 3.2.6).*

Proof. Recall that $\downarrow\uparrow$-$\mathsf{Unf}(\sigma)$ is defined as $\downarrow\uparrow$-$\mathsf{Unf}_!(\Lambda(\sigma)^!)$, which is innocent by Proposition 11.3.5. Moreover, by Proposition 11.3.6 its non-empty P-views are exactly the

$$s = (\overset{\frown}{-})(\partial_\sigma([m]_\sigma))$$

for $m \in \sigma$. But by *wb-threads*, $[m]_\sigma$ is a well-bracketed gcc; it entails directly that s is well-bracketed. Thus by Proposition 3.2.13, $\downarrow\uparrow$-$\mathsf{Unf}(\sigma)$ is well-bracketed. □

Together with Theorem 11.2.18, we get:

Theorem 11.3.10. *The alternating unfolding yields*

$$\downarrow\uparrow\text{-}\mathsf{Unf}(-) : \multimap\triangleright\text{-}\mathbf{SeqGlob} \to \downarrow\uparrow\text{-}\mathbf{InnWB}$$

a cartesian closed $\sim$-functor preserving the interpretation of PCF.

In fact, we shall be able to prove later on:

Theorem 11.3.11. *The induced functor $\downarrow\uparrow$-$\mathsf{Unf}(-) : \multimap\triangleright$-**SeqGlob**$/\approx \to \downarrow\uparrow$-**InnWB** is full and faithful: for all A, B mixed boards, the alternating unfolding induces a bijection*

$$\downarrow\uparrow\text{-}\mathsf{Unf}(-) \quad : \quad \multimap\triangleright\text{-}\mathbf{SeqGlob}(A, B)/\approx \quad \simeq \quad \downarrow\uparrow\text{-}\mathbf{InnWB}(\underline{A}, \underline{B})\,.$$

Proof. This requires the notion of *meager form* of a parallel innocent causal strategy, introduced in Section 12.1.2 – hence, we postpone the proof until Section 12.1.3. □

This should not surprise the reader: we have seen in Section 11.3.1 that the causal shape of a sequential innocent causal strategy is a forest, whose branches correspond to P-views enriched with explicit copy indices. Once we factor out the choice of copy indices via positive iso, we are left with exactly the same information as P-views.

11.3.4 Intensional Full Abstraction

At last, we are finally equipped to prove the final result of this chapter:

Theorem 11.3.12. *The interpretation of* PCF *in $\multimap\triangleright$-**SeqGlob** is intens. fully abstract.*

Proof. Let $\vdash M, N : A$ be terms in PCF, and assume $[\![M]\!] \not\simeq [\![N]\!]$, *i.e.* there is $\alpha \in \multimap\!\triangleright$-**SeqGlob**$([\![A]\!], [\![\mathbb{U}]\!])$ s.t. $\alpha \odot_! [\![M]\!] \neq \alpha \odot_! [\![N]\!]$ – assume *w.l.o.g.* $\alpha \odot_! [\![M]\!]$ converges while $\alpha \odot_! [\![M]\!]$ diverges. Then it follows from Theorem 11.3.7 that

$$\downarrow\uparrow\text{-}\mathsf{Unf}(\alpha) \odot [\![M]\!]_{\downarrow\uparrow\text{-}\mathbf{Strat}} \Downarrow \qquad\qquad \downarrow\uparrow\text{-}\mathsf{Unf}(\alpha) \odot [\![N]\!]_{\downarrow\uparrow\text{-}\mathbf{Strat}} \Uparrow,$$

where $\downarrow\uparrow\text{-}\mathsf{Unf}(\alpha)$ is an innocent alternating strategy, but also well-bracketed by Theorem 11.3.10. Thus finally, it follows that $M \not\simeq N$ by Theorem 4.3.10. □

This leaves us with one, final theorem to prove in Part III: that the model $\multimap\!\triangleright$-**Inn** of parallel innocent strategies is intensionally fully abstract for $\mathsf{PCF}_{/\!/}$. This will prove the most challenging of our full abstraction results, investigated in the next chapter.

11.4 History and Related Work

Though it seemed clear for a long time that concurrent strategies supported a fitting notion of sequentiality, the constructions presented in this chapter were not worked out until the long paper [Castellan and Clairambault, 2021], with a few significant suprises in store. The presentation given here is slightly simpler.

Related work. Though the results of this chapter are new, the developments resonate significantly with several earlier works. First and foremost, the structure of sequential innocent strategies is close to the designs of Girard's *Ludics* [Girard, 2001], and part of our developments are reminiscent of Faggian and Hyland's work on the connection between Ludics and Hyland-Ong games [Faggian and Hyland, 2002]. Again, the causal shape of sequential innocent strategies was also noticed by Melliès: diagrams such as in Figure 11.3 are very syntactic in nature, and may be written down via terms of the *non-uniform λ-calculus* introduced in [Melliès, 2004a].

Another notable aspect of this chapter is the analysis of polarities necessary to prove *sequentiality* stable under composition and tensoring (see *e.g.* Lemmas 11.1.8 and 11.1.16). Such lemmas are fundamental structures of the theory of sequentiality. In traditional presentations of alternating strategies, they show up early when proving determinism stable under composition, and associativity (Chapter 4); here instead they show up when proving sequentiality stable under composition. Usually they are kept implicit, relegated to technical appendices (if anything); but they are at the foreground of Melliès' recent *template games* [Melliès, 2019a, Melliès, 2019b].

Chapter 12

Finite Definability for $\mathsf{PCF}_{/\!/}$

We are approaching the end of the journey. Over the course of the previous three chapters, we have established the following intensional full abstraction results

$$\begin{array}{rll} \multimap\!\triangleright\text{-}\mathbf{Strat} & \text{is fully abstract for} & \mathsf{IA}_{/\!/}\,, \\ \multimap\!\triangleright\text{-}\mathbf{Strat} + \textit{sequentiality} & \text{is fully abstract for} & \mathsf{IA}\,, \\ \multimap\!\triangleright\text{-}\mathbf{Strat} + \textit{globularity} + \textit{sequentiality} & \text{is fully abstract for} & \mathsf{PCF}\,, \end{array}$$

and we are left with the one outstanding objective:

$$\multimap\!\triangleright\text{-}\mathbf{Strat} + \textit{globularity} \quad \text{is fully abstract for} \quad \mathsf{PCF}_{/\!/}\,.$$

This is also the most challenging of our full abstraction results: for the others we could leverage earlier work, but here we must prove finite definability from scratch.

Proving finite definability for globular strategies is a technical endeavour, involving a number of steps. First of all, what is "finite" in "finite definability"? Recall that this is already subtle for traditional innocent strategies, where finiteness is defined with respect to a *meager* representation: the forest of P-views (see Section 3.2.4). Accordingly, we shall start in Section 12.1 by providing a compact, *meager* form for parallel innocent strategies. Once this is done, we focus on the causal shape of globular strategies. There our observation, is that parallel behaviour is inherently *first-order*: two threads whose destiny is to be merged cannot delve into the higher-order structure. This allows us to prove a *factorization* result: any globular strategy can be decomposed into (the interpretation of) a pure λ-term, and a globular purely first-order strategy – this is detailed in Section 12.2. Finally, it remains to show finite definability for first-order globular strategies, which we do up to positional equivalence in Section 12.3.

Let us now delve in, starting with the meager form.

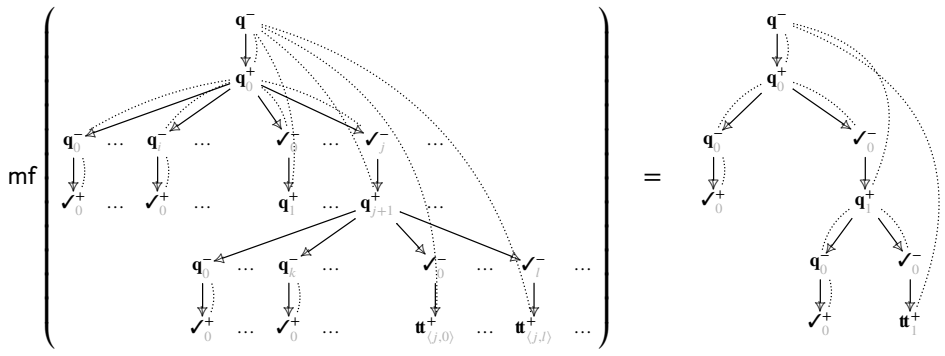


Figure 12.1: Meager form of a sequential innocent strategy on $(\mathbf{U} \to \mathbf{U}) \to \mathbf{B}$

12.1 Meager Form

Analogously to traditional innocent strategies, globular causal strategies support a notion of *meager form*, where all Opponent replications are factored out.

Introducing the meager form We have already seen in Section 11.3.1 the computation of the meager form for a sequential innocent strategy, summed up in Figure 12.1: so as to factor out Opponent's duplications, it suffices to only keep those Opponent moves that have 0 as their copy index (it is actually not straightforward how to make this formal for arbitrary mixed boards, we shall see later on how to do that). In doing so, we get a useful notion of finite sequential innocent strategy: although the tree at the left hand side of Figure 12.1 is infinite, the strategy is finite because factoring out Opponent's duplications yields the finite tree on the right hand side of Figure 12.1.

Then, the full strategy can be fully recovered from it: all the branches of the diagram on the left hand side of Figure 12.1 are copies of some branch of the meager form, and thus with each new Opponent copy of a move, we can make an independent copy of the subsequent subtree, choosing copy indices so as to avoid collisions. We shall see that the only information lost in going from left to right is the specific choice of copy indices, which however does not matter up to positive isomorphism.

This motivates the meager form and its expansion for sequential innocent strategies; but in this section we develop this in general for *parallel* innocent strategies. We show an example of the computation of the meager form of a parallel innocent strategy in Figure 12.2 – again we shall see that no essential information is lost. The diagram on the right hand side informs the full causal strategy: there is one copy of the final answer for each pair of answers to the two arguments, so that Opponent providing a new copy of the value for either argument prompts a new independent copy of the final answer.

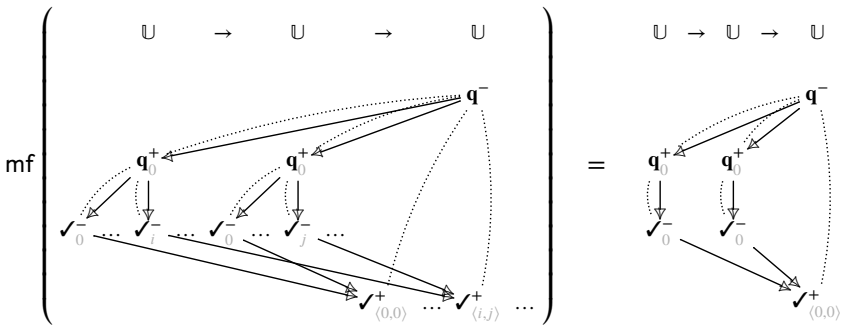


Figure 12.2: Meager form of parallel composition

12.1.1 Updating Mixed Boards

To make this formal, we need a way to specify the *copy index* of a move occurring deep in the board; forcing us to complete the definition of mixed boards (Definition 9.2.6).

Making copy indices explicit. If A is a board (and thus in particular, for a mixed board) its *causality* is forestial, thus any non-minimal event $a \in A$ has a unique **predecessor** for $\rightarrowtriangle_A$, written $\mathsf{pred}(a) \rightarrowtriangle_A a$; by convention we also specify that $\mathsf{pred}(a) = *$ if a is a minimal event. With these conventions in place, we complete Definition 9.2.6 (the intention is that the definition of mixed board was that below from the start, but with some of the components postponed, as they played no role until now).

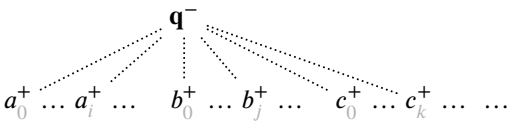
Definition 12.1.1. *A **mixed board** is $(A, \underline{A})$ with A a strict board, $\underline{A}$ an arena, with*

$$\mathsf{lbl}_A : |A| \to |\underline{A}|, \qquad \mathsf{ind}_A : |A| \to \mathbb{N},$$

*with lbl_A a **label** function preserving polarities, satisfying Definition 9.2.6, and ind_A an **indexing** function, with the following additional conditions:*

initialized:	*for all $\underline{a} \in \min(\underline{A})$ there is a unique $a \in \min(A)$ s.t. $\mathsf{lbl}(a) = \underline{a}$, and it additionally satisfies $\mathsf{ind}(a) = 0$.*
local conflict:	*if $a \frown a'$, we have $\mathsf{pred}(a) = \mathsf{pred}(a')$ and $\mathsf{ind}(a) = \mathsf{ind}(a')$,*
invariant conflict:	*if $a_1 \frown a_2$, $\mathsf{lbl}(a_1) = \mathsf{lbl}(b_1)$, $\mathsf{lbl}(a_2) = \mathsf{lbl}(b_2)$, $\mathsf{pred}(b_1) = \mathsf{pred}(b_2)$ and $\mathsf{ind}(b_1) = \mathsf{ind}(b_2)$, then $b_1 \frown b_2$.*
jointly injective:	*for $a_1, a_2 \in A$, if $\mathsf{lbl}(a_1) = \mathsf{lbl}(a_2)$, $\mathsf{ind}(a_1) = \mathsf{ind}(a_2)$, and $\mathsf{pred}(a_1) = \mathsf{pred}(a_2)$, then $a_1 = a_2$.*
wide:	*for any $a \in A$, $\underline{b} \in \underline{A}$ such that $\mathsf{lbl}(a) \rightarrowtriangle_{\underline{A}} \underline{b}$, and $n \in \mathbb{N}$, there is $b \in A$ s.t. $\mathsf{pred}(b) = a$, $\mathsf{lbl}(b) = \underline{b}$ and $\mathsf{ind}(b) = n$.*
+-transparent:	*for $\theta : x \cong_A y$, $\theta \in \mathcal{S}_+(A)$ iff for all $a^- \in x$, $\mathsf{ind}(\theta(a)) = \mathsf{ind}(a)$.*
−-transparent:	*for $\theta : x \cong_A y$, $\theta \in \mathcal{S}_-(A)$ iff for all $a^+ \in x$, $\mathsf{ind}(\theta(a)) = \mathsf{ind}(a)$.*

In particular, the function ind_A explicitly associates to every move $a \in A$ of the board a **copy index**. While this definition looks intimidating, its intention is simply to bind



$$\begin{aligned}
\mathsf{ind}(\mathbf{q}^-) &= 0\\
\mathsf{ind}(a_i^+) &= i\\
\mathsf{ind}(b_i^+) &= i\\
\mathsf{ind}(c_i^+) &= i\\
&\ldots
\end{aligned}$$

Figure 12.3: Copy indices for $\mathsf{ground}(V)$

tightly the board and the arena so that events of the board are determined by their label in the arena, their predecessor in the board and their copy index. By *transparent*, the copy index is the only thing that symmetries can change. By $+$*-transparent*, *positive symmetries* can only change indices of positive moves, while by $-$*-transparent*, *negative symmetries* can only change indices of negative moves.

Accounting for the new component $\mathsf{ind}_A : |A| \to \mathbb{N}$ of mixed boards, we must now define the copy indices for all our mixed board constructions. For basic mixed boards, recall that they were obtained by linking the arenas for ground types with the ground strict boards obtained as $\mathsf{ground}(V)$ for some set V of values. Thus we must equip this with a copy index function: for a set of values $V = \{a, b, c, \ldots\}$, we define the *copy indices* for the ground board $\mathsf{ground}(V)$ (Definition 9.1.1) as specified in Figure 12.3. For all basic mixed boards, it is clear that Definition 12.1.1 is satisfied. We extend the copy index function to all constructions on mixed boards used in the interpretation, via:

$$\begin{aligned}
\mathsf{ind}_{A_1 \& A_2}(i, a) &= \mathsf{ind}_{A_i}(a)\\
\mathsf{ind}_{A \Rightarrow B}(2, b) &= \mathsf{ind}_B(b)\\
\mathsf{lbl}_{A \Rightarrow B}(1, (b, (i, a))) &= i && \text{if } a \in \min(A),\\
\mathsf{lbl}_{A \Rightarrow B}(1, (b, (i, a))) &= \mathsf{ind}_A(a) && \text{otherwise,}
\end{aligned}$$

for $B = \&_{i \in I} B_i$ with B_i well-opened. We omit the routine verification that these definitions indeed preserve the conditions of Definition 12.1.1.

Introducing explicit copy indices only makes explicit a notion that we have already referred to in the exposition; for instance in diagrams such as in Figure 11.5 where the greyed-out subscripts are nothing but the numbers given by the copy index function. In mixed boards, the board is essentially an expanded form of the arena, with all branches replaced with countably many copies: playing a new move consists in picking a justifier (if any), picking an enabled move in the arena, and picking a copy index[1]

Properties. Equipped with this now completed notion of mixed board, we may now prove the property that was postponed after Definition 9.2.6.

[1] In fact, we could have set $\multimap$-**Strat** as having mere arenas as objects, and having strategies play on the generated board – this was the road suggested in [Castellan et al., 2019]. We prefer the present presentation, which allows for a smoother link with the relative Seely category structure of **NTCG** and the relational model.

Lemma 9.2.10. *Consider A a mixed board. We have the following properties:*

(1) *For all $ta \in \mathcal{O}\text{-}\widehat{\mathsf{PrePlays}}_{\bullet}(\underline{A})$, $s \in \mathcal{O}\text{-}\mathsf{Plays}(A)$ such that $t = \mathfrak{F}$, there is $sa' \in \mathcal{O}\text{-}\mathsf{Plays}(A)$ such that $(\overset{\curvearrowleft}{-})(sa') = ta$;*
(2) *For all $ta \in \mathcal{O}\text{-}\widehat{\mathsf{PrePlays}}(\underline{A})$, $s \in \mathcal{O}\text{-}\mathsf{Plays}(!A)$ such that $t = \mathfrak{F}$, there is $sa' \in \mathcal{O}\text{-}\mathsf{Plays}(!A)$ such that $(\overset{\curvearrowleft}{-})(sa') = ta$.*

Proof. (1) Consider $ta \in \mathcal{O}\text{-}\widehat{\mathsf{PrePlays}}_{\bullet}(\underline{A})$ and $s \in \mathcal{O}\text{-}\mathsf{Plays}(A)$ such that $t = \mathfrak{F}$. If t is empty, then a is initial in $\underline{A}$ and the property follows by *initialized*. Otherwise, write

$$s = s_1 \dots s_n \qquad t = \mathfrak{F} = t_1 \dots t_n$$

where a points to t_i. Now, consider p a copy index not yet appearing in s. By *wide*, there is $a' \in A$ such that $\mathsf{lbl}(a') = a$, $\mathsf{pred}(a') = s_i$ and $\mathsf{ind}(a') = p$. Then $sa' \in \mathscr{L}(A)$. Indeed, $|s| \cup \{a'\} \in \mathscr{C}(A)$: it is down-closed as $\mathsf{pred}(a') \in |s|$ and conflict-free by *locally conflicting* since no other move in s has index p. Finally, $(\overset{\curvearrowleft}{-})(sa') = ta$.

(2) same reasoning as for *(1)*. □

12.1.2 Meager Innocent Strategies

Now, we are equipped to define *meager parallel innocent strategies*, the notion analogous to the *meager innocent strategies* in the traditional setting (see Section 3.2.4).

Definition. In analogy to the traditional case, *meager innocent strategies* are supposed to be parallel innocent strategies, but playing on games not allowing any Opponent replications. We formalize this by restricting Opponent to copy index 0:

Definition 12.1.2. *Consider A a mixed board. Then, we set A_+ a strict board with*

$$|A_+| = \{a' \in |A| \mid \forall a^- \leq_A a',\ \mathsf{ind}_A(a) = 0\}\,,$$

*and all other components inherited. We call A_+ the **positive restriction** of A.*

This definition of the positive restriction has the immediate consequence that its only non-trivial symmetries are positive, since Opponent moves must have index 0:

Lemma 12.1.3. *For A a mixed board, if $\theta \in \mathcal{S}(A_+)$ then $\theta \in \mathcal{S}_+(A_+) \subseteq \mathcal{S}_+(A)$.*

This is obvious by *+-transparent*. We now define meager parallel innocent strategies:

Definition 12.1.4. *Consider A a mixed board.*
*A **meager parallel innocent strategy** on A is a parallel innocent strategy on A_+.*

For instance, the diagrams on the right hand side of Figures 12.1 and 12.2 are meager parallel innocent strategies. Note that on the left hand side of these diagrams, the infinite nature of the strategies forces us to resort to a symbolic notation so as to describe those finitely; also the notation conveys no information on the symmetries, although those are in principle part of the data. In contrast, for meager parallel innocent strategies:

Proposition 12.1.5. *Consider A a mixed board, and $\sigma : A_+$ meager parallel innocent. Then its symmetries are trivial,* i.e. $\mathcal{S}(\sigma) = \{\mathrm{id}_x : x \cong_\sigma x \mid x \in \mathscr{C}(\sigma)\}$.

Proof. Consider $\theta : x \cong_\sigma y$ any symmetry. By Lemma 12.1.3, $\partial_\sigma \theta \in \mathcal{S}_+(A_+)$. Hence by *thin*, we have that $x = y$ and $\theta = \mathrm{id}_x$; thus $\mathcal{S}(\sigma)$ is restricted to identities. □

Thus when specifying a meager parallel innocent strategy σ, it is unnecessary to specify the isomorphism family $\mathcal{S}(\sigma)$, which is trivial. Thus it is entirely sufficient to describe σ with a diagram (as on the right hand side of Figures 12.1 and 12.2) describing the event structure concretely, with no ellipse or symbolic representation. Likewise, a *positive iso* between meager parallel innocent strategies boils down to an isomorphism

$$\varphi : \sigma \cong \tau$$

of plain event structures such that $\partial_\tau \circ \varphi \sim \partial_\sigma$, which amounts to renaming the internal events of σ and τ and changing the copy indices of positive moves.

Meager form. The above is interesting, *provided* it does indeed give an alternative but equivalent representation for usual parallel innocent strategies. Accordingly, we now define the *meager form* of parallel innocent strategies – in the obvious way:

Proposition 12.1.6. *Consider A a mixed board, and $\sigma : A$ a parallel innocent strategy. Then, setting the set of events*

$$|\mathsf{mf}(\sigma)| = \{m \in |\sigma| \mid \partial_\sigma [m]_\sigma \in \mathscr{C}(A_+)\}$$

with all components inherited yields a meager parallel innocent strategy $\mathsf{mf}(\sigma)$ on A.

Moreover, $\mathsf{mf}(-)$ preserves $\approx$.

Proof. There are a few conditions to check, which we list below.

Event structure with symmetry. It is easy that $\mathsf{mf}(\sigma)$ is an event structure. For symmetry only the *extension* axiom is non-trivial: if $\theta : x \cong_\sigma y$ with $x, y \in \mathscr{C}(\mathsf{mf}(\sigma))$, then by the same reasoning as in Proposition 12.1.5, $\theta = \mathrm{id}_x$; *extension* follows.

Strategy, winning, visible, parallel innocent. Immediate verifications.

Preservation of $\approx$. Consider $\varphi : \sigma \approx \tau$ a positive isomorphism. Then, the important observation is that φ sends $\mathsf{mf}(\sigma)$ to $\mathsf{mf}(\tau)$. Indeed if $m \in \mathsf{mf}(\sigma)$, then

$$\langle \partial_\sigma, \partial_\tau \circ \varphi \rangle_{[m]_\sigma} : \partial_\sigma [m]_\sigma \cong^+_A \partial_\tau [\varphi(m)]_\tau$$

which by $+$-*transparent*, being positive, must preserve the copy indices of negative events. Hence $\partial_\tau[\varphi(m)]_\tau \in \mathscr{C}(A_+)$, and φ restricts to $\mathsf{mf}(\varphi) : \mathsf{mf}(\sigma) \approx \mathsf{mf}(\tau)$. □

Note that in the construction of $\mathsf{mf}(\sigma)$ we have never actually used the hypothesis that σ is parallel innocent, and the construction works in more generality. However, the operation is not very interesting for non-innocent strategies as it is lossy: if σ is not parallel innocent, one cannot in general reconstruct σ from $\mathsf{mf}(\sigma)$.

Next, we show how to recover a parallel innocent strategy from a meager form.

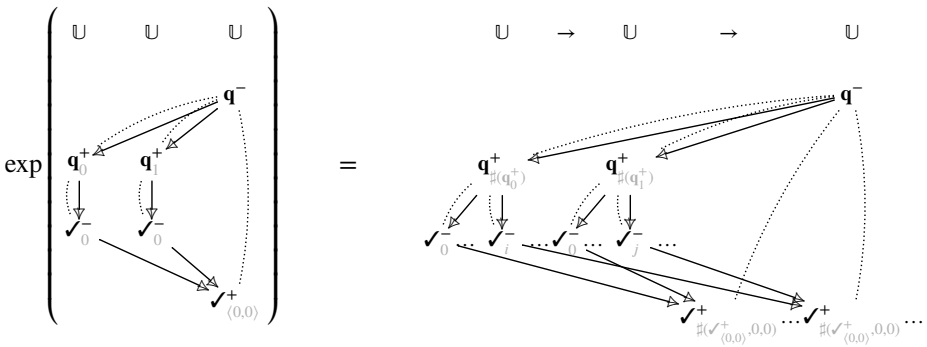


Figure 12.4: Expansion of meager parallel composition

Expansion of a meager parallel innocent strategy. Fix A a mixed board, and $\sigma : A_+$ meager parallel innocent on A. We must infer from σ its behaviour if Opponent performs arbitrary duplications, by performing adequately many copies of its events. We shall achieve that by pairing each event of a meager parallel innocent strategy with the choice of a copy index for each of its duplicable Opponent dependencies, *i.e.* its negative dependencies save for the initial move – we call such choice an *exponential slice*. We must then assign a new copy index for positive moves, hashing together the positive move and the exponential slice. The process is displayed in Figure 12.4.

Consider $m \in \sigma$. How many copies of m should we include in the expanded form? The answer is that there should be one copy for each simultaneous choice of a copy index for each duplicable negative dependency of m. Write

$$[m]^-_\sigma = \{n^- \in |\sigma| \mid n^- \leq_\sigma m\}$$

for the **negative dependencies** of m. Then, we define:

Definition 12.1.7. *An **exponential slice** for σ is a function, for some $m \in \sigma$,*

$$\alpha : [m]^-_\sigma \to \mathbb{N}\,.$$

such that $\alpha(n) = 0$ for n minimal.

*An **expanded event** is $\mathsf{m} = (m, \alpha_m : [m]^-_\sigma \to \mathbb{N})$, we write $\mathsf{m} = (m, \alpha) \in \exp(\sigma)$.*

We write $\mathsf{lbl}((m,\alpha)) = m$. If $\mathsf{lbl}(\mathsf{m})$ is negative we say that m is **negative**, and write $\mathsf{ind}((m,\alpha)) = \alpha(m)$ for its **copy index**. For $\mathsf{m} = (m,\alpha), \mathsf{n} = (n,\beta) \in \exp(\sigma)$, we write $\mathsf{m} \sqsubseteq \mathsf{n}$ iff $m \leq_\sigma n$, and for all $e^- \leq_\sigma m$, we have $\alpha(e^-) = \beta(e^-)$. Note that if m is negative, as $\mathsf{lbl}(\mathsf{m})$ has a unique predecessor in σ, m must have a unique predecessor in $\exp(\sigma)$, written $\mathsf{pred}(\mathsf{m})$. We set the *minimal conflict* $\mathsf{m} \smile_{\exp(\sigma)} \mathsf{n}$ iff m and n are negative, $\mathsf{pred}(\mathsf{m}) = \mathsf{pred}(\mathsf{n})$, $\mathsf{lbl}(\mathsf{m}) \smile_\sigma \mathsf{lbl}(\mathsf{n})$ and $\mathsf{ind}(\mathsf{m}) = \mathsf{ind}(\mathsf{n})$.

Towards defining a strategy, we first extend this to an event structure with symmetry:

Proposition 12.1.8. *Consider* $\exp(\boldsymbol{\sigma})$ *with events* $|\exp(\boldsymbol{\sigma})|$*, components*

$$\begin{array}{rcl} \mathsf{m} \leq_{\exp(\boldsymbol{\sigma})} \mathsf{n} & \Leftrightarrow & \mathsf{m} \sqsubseteq \mathsf{n} \\ \mathsf{m} \#_{\exp(\boldsymbol{\sigma})} \mathsf{n} & \Leftrightarrow & \exists \mathsf{m}' \sqsubseteq \mathsf{m},\ \mathsf{n}' \sqsubseteq \mathsf{n},\ \mathsf{m}' \sim_{\exp(\boldsymbol{\sigma})} \mathsf{n}' \end{array}$$

and $\mathcal{S}(\exp(\boldsymbol{\sigma}))$ *comprising all order-isos preserving* $\mathsf{lbl}(-)$ *between configurations.*

Then, $\exp(\boldsymbol{\sigma})$ *is an event structure with symmetry.*

Proof. The less trivial point is that $\#_{\exp(\boldsymbol{\sigma})}$ is irreflexive so assume $\mathsf{m} \#_{\exp(\boldsymbol{\sigma})} \mathsf{m}$, meaning there are $\mathsf{n}, \mathsf{o} \sqsubseteq \mathsf{m}$ such that $\mathsf{n} \sim_{\exp(\boldsymbol{\sigma})} \mathsf{o}$. So $\mathsf{pred}(\mathsf{n}) = \mathsf{pred}(\mathsf{o}) = \mathsf{p}$, $\mathsf{lbl}(\mathsf{n}) \sim_{\boldsymbol{\sigma}} \mathsf{lbl}(\mathsf{o})$, and $\mathsf{ind}(\mathsf{n}) = \mathsf{ind}(\mathsf{o})$. But this means we have $\mathsf{lbl}(\mathsf{n}) \leq_{\boldsymbol{\sigma}} \mathsf{lbl}(\mathsf{m})$ and $\mathsf{lbl}(\mathsf{o}) \leq_{\boldsymbol{\sigma}} \mathsf{lbl}(\mathsf{m})$ with $\mathsf{lbl}(\mathsf{n}) \sim_{\boldsymbol{\sigma}} \mathsf{lbl}(\mathsf{o})$, thus $\mathsf{lbl}(\mathsf{m}) \#_{\boldsymbol{\sigma}} \mathsf{lbl}(\mathsf{m})$, contradiction since $\boldsymbol{\sigma}$ is an event structure.

Event structure with symmetry. The non-trivial point is the *extension* axiom. Consider $\theta : x \cong_{\exp(\boldsymbol{\sigma})} y$, and $x \vdash_{\exp(\boldsymbol{\sigma})} (m, \alpha)$. If m is positive, then for each $n^- \leq_{\boldsymbol{\sigma}} m$, we have $(n^-, \alpha \upharpoonright [n^-]^-_{\boldsymbol{\sigma}}) \in x$, and we write $\theta((n^-, \alpha \upharpoonright [n^-]^-_{\boldsymbol{\sigma}})) = (n^-, \beta_n)$. Then we set:

$$\gamma(n^-) = \beta_n(n^-)$$

for all $n^- \leq_{\boldsymbol{\sigma}} m$, yielding $(m, \gamma) \in \exp(\boldsymbol{\sigma})$. Then, we observe that $y \vdash_{\exp(\boldsymbol{\sigma})} (m, \gamma)$: indeed if $(p, \gamma') \leq_{\exp(\boldsymbol{\sigma})} (m, \gamma)$ then one can prove that $\theta((p, \alpha')) = (p, \gamma')$ for $\alpha' = \alpha \upharpoonright [p]^-_{\boldsymbol{\sigma}}$. Moreover, $y \cup \{(m, \gamma)\}$ is consistent as immediate conflicts are negative. Likewise $\theta \vdash ((m, \alpha), (m, \gamma))$ by definition of $\mathcal{S}(\exp(\boldsymbol{\sigma}))$. Finally if m is negative, then write $(n, \beta) = \mathsf{pred}(m, \alpha)$ and $(n, \beta') = \theta(n, \beta)$. Set $\alpha' : [m]_{\boldsymbol{\sigma}} \to \mathbb{N}$ as $\alpha'(o) = \beta(o)$ for $o <_{\boldsymbol{\sigma}} m$, and $\alpha'(m) = i$ fresh; then it is straightforward that $\theta \vdash ((m, \alpha), (m, \alpha'))$. □

Now, in order to obtain a strategy, we must specify its display map. In particular, we must set a copy index for all moves. For negative moves, it should clearly match that given by the exponential slice. For positive moves, we choose arbitrarily

$$\sharp : \Sigma_{m^+ \in |\boldsymbol{\sigma}|} \mathbb{N}^{[m^+]^-_{\boldsymbol{\sigma}}} \to \mathbb{N}$$

any injective function, which we shall use to fix a copy index for positive moves – note that the strategy obtained will not depend (up to positive iso) on this choice.

In order to better formulate the definition, we shall use the lemma:

Lemma 12.1.9. *Consider* $\underline{a} \in \underline{A}$ *non-minimal, and write* $\mathsf{pred}(\underline{a}) = \underline{b}$ *its predecessor.*

Then for all $b \in A$ *such that* $\mathsf{lbl}(b) = \underline{b}$*, and* $n \in \mathbb{N}$*, there is a unique* $a \in A$ *such that* $\mathsf{lbl}(a) = \underline{a}$*,* $\mathsf{pred}(a) = b$ *and* $\mathsf{ind}(a) = n$*.*

Proof. Direct consequence of *wide* and *jointly injective* of mixed boards. □

Notice also that each non-minimal $\mathsf{m} \in \exp(\boldsymbol{\sigma})$ has a natural notion of **justifier**: for $\mathsf{m} = (m, \alpha)$, we set $\mathsf{j}(\mathsf{m}) = (\mathsf{j}(m), \alpha \upharpoonright [\mathsf{j}(m)]^-_{\boldsymbol{\sigma}})$ – we make use of this below:

Lemma 12.1.10. *The function defined by induction on $\leq_{\boldsymbol{\sigma}}$ via the clauses*

$$\partial_{\exp(\boldsymbol{\sigma})}(m^-,\alpha) = a^- \quad \textit{such that} \quad \begin{cases} \mathsf{pred}(a^-) &= \partial_{\exp(\boldsymbol{\sigma})}(\mathsf{j}(m^-,\alpha)) \\ \mathsf{lbl}(a^-) &= \mathsf{lbl}(\partial_{\boldsymbol{\sigma}}(m^-)) \\ \mathsf{ind}(a^-) &= \alpha(m^-) \end{cases}$$

$$\partial_{\exp(\boldsymbol{\sigma})}(m^+,\alpha) = a^+ \quad \textit{such that} \quad \begin{cases} \mathsf{pred}(a^+) &= \partial_{\exp(\boldsymbol{\sigma})}(\mathsf{j}(m^+,\alpha)) \\ \mathsf{lbl}(a^+) &= \mathsf{lbl}(\partial_{\boldsymbol{\sigma}}(m^+)) \\ \mathsf{ind}(a^+) &= \sharp(m^+,\alpha) \end{cases}$$

is a map of event structures with symmetry $\partial_{\exp(\boldsymbol{\sigma})} : \exp(\boldsymbol{\sigma}) \to A$.

Proof. It is clear by construction that the image of a configuration is down-closed, and it is easily verified that $\partial_{\exp(\boldsymbol{\sigma})}$ is locally injective. Now considering $x \in \mathscr{C}(\exp(\boldsymbol{\sigma}))$, we must show that $\partial_{\exp(\boldsymbol{\sigma})}(x)$ is conflict-free. Seeking a contradiction, assume there are $\mathsf{m} = (m^-,\alpha)$ and $\mathsf{n} = (n^-,\beta)$ s.t. $\partial_{\exp(\boldsymbol{\sigma})}(\mathsf{m}) \smile_{\exp(\boldsymbol{\sigma})} \partial_{\exp(\boldsymbol{\sigma})}(\mathsf{n})$. By definition of $\partial_{\exp(\boldsymbol{\sigma})}$, $\mathsf{lbl}(\partial_{\boldsymbol{\sigma}}(m^-)) = \mathsf{lbl}(\partial_{\exp(\boldsymbol{\sigma})}(\mathsf{m}))$ and $\mathsf{lbl}(\partial_{\boldsymbol{\sigma}}(n^-)) = \mathsf{lbl}(\partial_{\exp(\boldsymbol{\sigma})}(\mathsf{n}))$. By *local conflict*, $\partial_{\exp(\boldsymbol{\sigma})}(\mathsf{m})$ and $\partial_{\exp(\boldsymbol{\sigma})}(\mathsf{n})$ must have the same predecessor, so $\mathsf{j}(m^-,\alpha)$ and $\mathsf{j}(n^-,\beta)$ have the same display. But they are compatible, thus by local injectivity they must be equal – in turn this entails that $\mathsf{pred}(m^-) = \mathsf{pred}(n^-)$, thus $\mathsf{pred}(\partial_{\boldsymbol{\sigma}}(m^-)) = \mathsf{pred}(\partial_{\boldsymbol{\sigma}}(n^-))$ by *courtesy*. Thus by *invariant conflict*, we have $\partial_{\boldsymbol{\sigma}}(m^-) \smile_A \partial_{\boldsymbol{\sigma}}(n^-)$ which entails $m^- \smile_{\boldsymbol{\sigma}} n^-$. Summing up, $\mathsf{lbl}(\mathsf{m}) \smile_{\boldsymbol{\sigma}} \mathsf{lbl}(\mathsf{n})$, $\mathsf{ind}(\mathsf{m}) = \mathsf{ind}(\mathsf{n})$ since $\mathsf{ind}(\partial_{\exp(\boldsymbol{\sigma})}(\mathsf{m})) = \mathsf{ind}(\partial_{\exp(\boldsymbol{\sigma})}(\mathsf{n})$ by *local conflict*, and $\mathsf{pred}(\mathsf{m}) = \mathsf{pred}(\mathsf{n})$, thus $\mathsf{m} \smile_{\exp(\boldsymbol{\sigma})} \mathsf{n}$.

Finally if $\theta \in \mathscr{S}(\exp(\boldsymbol{\sigma}))$, it is by definition an order-isomorphism preserving labels. From that it is obvious that it must preserve justifiers. It follows from the definition that $\partial_{\exp(\boldsymbol{\sigma})}\,\theta$ is an order-iso preserving labels, hence a symmetry by *transparent*. □

There is still a lot of work to do in order to obtain a causal strategy; first we must construct a strategy in the sense of Definition 6.1.6. But in order to prove properties based on configurations (starting with *receptive*), we must understand better how configurations of $\exp(\boldsymbol{\sigma})$ relate to configurations of $\boldsymbol{\sigma}$. We shall establish such a connection not in general, but in the particular case of so-called *linear* configurations:

Definition 12.1.11. *Consider A a mixed board, $\boldsymbol{\sigma} : A$ a strategy.*

*Then, $x \in \mathscr{C}(\boldsymbol{\sigma})$ is **linear** if for all $n_1^-, n_2^- \in x$, if $\mathsf{pred}(n_1^-) = \mathsf{pred}(n_2^-)$ then $n_1^- = n_2^-$.*

Linear configurations are analogous to P-views: Opponent may point on any given Player move at most once. Their interest comes from the following observation:

Lemma 12.1.12. *If $\boldsymbol{\sigma} : A$ is parallel innocent and $m \in \boldsymbol{\sigma}$, $[m]_{\boldsymbol{\sigma}} \in \mathscr{C}(\boldsymbol{\sigma})$ is linear.*

If $\boldsymbol{\sigma} : A$ meager parallel innocent and $\mathsf{m} \in \exp(\boldsymbol{\sigma})$, then $[\mathsf{m}]_{\exp(\boldsymbol{\sigma})}$ is linear.

Proof. The first is a direct consequence of pre-innocence (and Lemma 6.1.16); the second is straightforward from the definition, boiling down to the pre-innocence of $\boldsymbol{\sigma}$. □

Next, we observe that if $\boldsymbol{\sigma} : A$ is a meager parallel innocent strategy, any linear configuration $x \in \mathscr{C}(\boldsymbol{\sigma})$ may be canonically regarded as a (linear) configuration of $\exp(\boldsymbol{\sigma})$ by adjoining the constant 0 copy index to all Opponent moves:

Lemma 12.1.13. *Consider A a mixed board, and $\boldsymbol{\sigma} : A$ meager parallel innocent. For each linear $x \in \mathscr{C}(\boldsymbol{\sigma})$, with $0_m : [m]^-_{\boldsymbol{\sigma}} \to \mathbb{N}$ the constant 0 function, setting*

$$\exp(x) = \{(m, 0_m) \mid m \in x\}\,,$$

we get $\exp(x) \in \mathscr{C}(\exp(\boldsymbol{\sigma}))$ a linear configuration; and $\kappa_x : m \mapsto (m, 0_m)$ is an order-isomorphism $\kappa_x : x \cong \exp(x)$ compatible with display maps, i.e. $\partial_{\exp(\boldsymbol{\sigma})} \circ \kappa_x \sim^+ \partial_{\boldsymbol{\sigma}}$.

Proof. Straightforward, since meager parallel innocent strategies play on A_+. □

It is also straightforward that κ_x preserves and reflects symmetry. Reciprocally, *all* linear configurations of $\exp(\boldsymbol{\sigma})$ are canonically symmetric to a configuration of $\boldsymbol{\sigma}$:

Lemma 12.1.14. *Consider A a mixed board, meager $\boldsymbol{\sigma} : A$, and $x \in \mathscr{C}(\exp(\boldsymbol{\sigma}))$ linear. There are unique $\mathsf{mf}(x) \in \mathscr{C}(\boldsymbol{\sigma})$ and $\theta_x : x \cong_{\exp(\boldsymbol{\sigma})} \exp(\mathsf{mf}(x))$, given by*

$$\mathsf{mf}(x) = \{m \mid (m, \alpha) \in x\}\,, \qquad \theta_x = \{((m, \alpha), (m, 0_m)) \mid (m, \alpha) \in x\}\,.$$

Proof. Existence. We prove that this data satisfies the required conditions by induction on x. For x empty this is clear. If $x \vdash (m^-, \alpha)$, then by induction hypothesis, $\mathsf{mf}(x) \in \mathscr{C}(\boldsymbol{\sigma})$. It is immediate that $\mathsf{mf}(x) \cup \{m^-\}$ is down-closed. If it was conflicting, then there would be $n^- \in \mathsf{mf}(x)$ such that $m^- \smile_{\boldsymbol{\sigma}} n^-$. But then, by Lemma 6.1.17, we have $\partial_{\boldsymbol{\sigma}}(m^-) \smile_A \partial_{\boldsymbol{\sigma}}(n^-)$, which by *local conflict* entails $\mathsf{pred}(\partial_{\boldsymbol{\sigma}}(m^-)) = \mathsf{pred}(\partial_{\boldsymbol{\sigma}}(n^-))$, hence $\mathsf{pred}(m^-) = \mathsf{pred}(n^-)$ by Lemma 6.1.16. Writing $p^+ \in \boldsymbol{\sigma}$ for this predecessor,

$$(p^+, \alpha') \to_{\exp(\boldsymbol{\sigma})} (m^-, \alpha)\,, \qquad (p^+, \beta') \to_{\exp(\boldsymbol{\sigma})} (n^-, \beta)$$

with all four events in $x \cup \{(m^-, \alpha)\}$ linear. If $\alpha' = \beta'$ we get $(m^-, \alpha) = (n^-, \beta)$ by linearity hence $m^- = n^-$, contradiction. Otherwise $(p^+, \alpha'), (p^+, \beta') \in x$ differ, but this directly entails a failure of linearity as well – contradiction. Now if $x \vdash (m^+, \alpha)$, as above $\mathsf{mf}(x) \cup \{m^+\}$ is down-closed. It must also be conflict-free, since by *causal determinism*, minimal conflicts in $\boldsymbol{\sigma}$ are negative. It is direct to check that $\xi_x : x \cong \mathsf{mf}(x)$ sending (m, α) to m is an order-isomorphism and that $\theta_x = \kappa_{\mathsf{mf}(x)} \circ \xi_x$ is an order-iso preserving labels, hence a symmetry in $\exp(\boldsymbol{\sigma})$ as required.

Uniqueness. Assume we have $y \in \mathscr{C}(\boldsymbol{\sigma})$ and $\theta'_x : x \cong_{\exp(\boldsymbol{\sigma})} \exp(y)$. Then,

$$\theta'_x \circ \theta_x^{-1} : \exp(\mathsf{mf}(x)) \cong_{\exp(\boldsymbol{\sigma})} \exp(y)$$

which entails $\exp(\mathsf{mf}(x)) = \exp(y)$ (so that $\mathsf{mf}(x) = y$) and $\theta'_x \circ \theta_x^{-1}$ is an identity by definition of symmetries in $\exp(\boldsymbol{\sigma})$; this concludes the proof. □

Now, we may finally prove:

Proposition 12.1.15. *Consider A a mixed board, and $\boldsymbol{\sigma} : A$ meager parallel innocent. Then,* $\exp(\boldsymbol{\sigma})$ *with the components above is a parallel innocent strategy.*

Proof. Receptive. Consider $x \in \mathscr{C}(\exp(\boldsymbol{\sigma}))$ and $\partial_{\exp(\boldsymbol{\sigma})}\, x \vdash_A a^-$. Call $c^+ \in \partial_{\exp(\boldsymbol{\sigma})}\, x$ the predecessor of a^- in x – there is a unique $\mathsf{m} = (m, \alpha) \in x$ such that $\partial_{\exp(\boldsymbol{\sigma})}\, \mathsf{m} = c^+$. By Lemma 12.1.12, $[\mathsf{m}]_{\exp(\boldsymbol{\sigma})}$ is linear, and thus by Lemma 12.1.14 there is

$$\theta_x \,:\, [(m,\alpha)]_{\exp(\boldsymbol{\sigma})} \cong_{\exp(\boldsymbol{\sigma})} [(m, 0_m)]_{\exp(\boldsymbol{\sigma})}\,,$$

mapping via $\partial_{\exp(\boldsymbol{\sigma})}$ to a symmetry in A which must extend by some (a^-, b^-). By *wide*, we may assume that $\mathsf{ind}(b^-) = 0$ so that the extension is in A_+. By Lemma 12.1.13 and receptivity of $\boldsymbol{\sigma}$, there is a unique $[m]_{\boldsymbol{\sigma}} \vdash_{\boldsymbol{\sigma}} n^-$ such that $\partial_{\boldsymbol{\sigma}}(n^-) = b^-$. Extending α to $\alpha' \,:\, [n^-]^-_{\boldsymbol{\sigma}} \to \mathbb{N}$ with $\alpha'(n^-) = \mathsf{ind}(a^-)$, by *jointly injective* we get the required extension – it is compatible with x by definition of minimal conflict.

$\sim$*-receptive* is similar, and all further conditions are straightforward. □

This at last concludes the construction of $\exp(\boldsymbol{\sigma})$ as a parallel innocent strategy.

Equivalence of meager and expanded forms. Finally, so as to show that meager parallel innocent strategies provide an equivalent representation of parallel innocent strategies, we must prove that the two constructions $\mathsf{mf}(-)$ and $\exp(-)$ preserve positive isomorphisms, and are inverses up to positive isomorphism.

Proposition 12.1.16. *The constructions* $\mathsf{mf}(-)$ *and* $\exp(-)$ *preserve positive isos.*

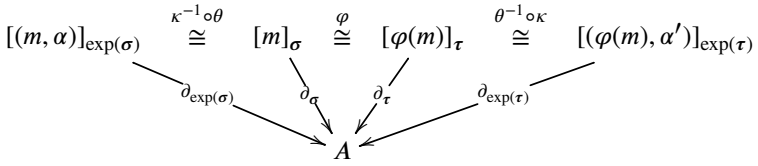
Proof. It is obvious that $\mathsf{mf}(-)$ preserves positive isomorphisms. Consider $\varphi \,:\, \boldsymbol{\sigma} \approx \boldsymbol{\tau}$ a positive iso between meager parallel innocent strategies on A; we must construct

$$\exp(\varphi) \,:\, \exp(\boldsymbol{\sigma}) \approx \exp(\boldsymbol{\tau})$$

a positive isomorphism between the expansions. We simply set:

$$\begin{array}{rcccc} \exp(\varphi) & : & \exp(\boldsymbol{\sigma}) & \approx & \exp(\boldsymbol{\tau}) \\ & & (m,\alpha) & \mapsto & (\varphi(m), \alpha') \end{array}$$

where for $\varphi(n)^- \leq_{\boldsymbol{\tau}} \varphi(m)$, $\alpha'(\varphi(n)) = \alpha(n)$. It is direct that this is an order-iso; it preserves and reflects immediate conflict by *causal determinism* and *invariant conflict*. For positivity, for all $(m, \alpha) \in \exp(\boldsymbol{\sigma})$, $\exp(\varphi)$ on $[(m,\alpha)]_{\exp(\boldsymbol{\sigma})}$ factors as



omitting subscripts for readability – where every triangle commutes up to symmetry via Lemmas 12.1.13 and 12.1.14, and positivity follows by $+$*-transparent*. □

It remains to show that $\mathsf{mf}(-)$ and $\exp(-)$ are inverses up to positive iso. We start with the easy direction, that a meager strategy can be recovered from its expansion:

Proposition 12.1.17. *Consider $\boldsymbol{\sigma}$: A meager parallel innocent. Then, the function*

$$\begin{array}{rccl} \Phi & : & \boldsymbol{\sigma} & \to & \mathsf{mf}(\exp(\boldsymbol{\sigma})) \\ & & m & \mapsto & (m, 0_m)\,, \end{array}$$

yields a positive isomorphism $\Phi : \boldsymbol{\sigma} \approx \mathsf{mf}(\exp(\boldsymbol{\sigma}))$.

Proof. We omit the direct verification that this is an isomorphism of ess. Positivity follows from Lemma 12.1.13 on prime configurations, extended by taking unions. □

Finally we must show that analogously, a full parallel innocent strategy can always be recovered from its meager form. This means that given a parallel innocent strategy $\boldsymbol{\sigma} : A$, we must provide a bijection between events of $\boldsymbol{\sigma}$ and of $\exp(\mathsf{mf}(\boldsymbol{\sigma}))$, hence extract from any $m \in \boldsymbol{\sigma}$ a *label* in $\mathsf{mf}(\boldsymbol{\sigma})$ and an *exponential slice*.

We start by the following generalization of Lemma 12.1.14:

Lemma 12.1.18. *Consider A a mixed board, and $\boldsymbol{\sigma} : A$ a parallel innocent strategy. For $x \in \mathscr{C}(\boldsymbol{\sigma})$ linear, there are unique $\mathsf{mf}(x) \in \mathscr{C}(\mathsf{mf}(\boldsymbol{\sigma}))$ and $\theta_x : x \cong_{\boldsymbol{\sigma}} \mathsf{mf}(x)$.*

Proof. Existence. By induction on x. If x is empty there is nothing to do. If $x \vdash_{\boldsymbol{\sigma}} a^-$, then θ_x extends with some (a^-, b^-). By *wide*, we may assume that $\mathsf{ind}(b^-) = 0$, so that it fits in A_+ (such b^- cannot be in $\mathsf{mf}(x)$ by linearity of x). If $x \vdash_{\boldsymbol{\sigma}} a^+$, then the extension of θ_x automatically does the trick.

Uniqueness. As in Lemma 12.1.14. □

In particular, given parallel innocent $\boldsymbol{\sigma} : A$ and $m \in \boldsymbol{\sigma}$ we know from Lemma 12.1.12 that $[m]_{\boldsymbol{\sigma}}$ is linear, thus Lemma 12.1.18 gives us a unique $\mathsf{mf}([m]_{\boldsymbol{\sigma}}) = [\mathsf{lbl}(m)]_{\boldsymbol{\sigma}} \in \mathscr{C}(\mathsf{mf}(\boldsymbol{\sigma}))$ with $\theta_m : [m]_{\boldsymbol{\sigma}} \cong_{\boldsymbol{\sigma}} [\mathsf{lbl}(m)]_{\boldsymbol{\sigma}}$. As the notation suggests, $\mathsf{lbl}(m)$ is considered as the *label* of m, *i.e.* the move of $\mathsf{mf}(\boldsymbol{\sigma})$ that m is considered a duplicate of.

This duplicate is also characterized by an *exponential slice*, defined as:

$$\begin{array}{rccl} \alpha & : & [\mathsf{lbl}(m)]^-_{\boldsymbol{\sigma}} & \to & \mathbb{N} \\ & & \theta_m(n) & \mapsto & \mathsf{ind}(\partial_{\boldsymbol{\sigma}}(n))\,, \end{array}$$

and writing $\alpha = \mathsf{sl}(m)$ we get $(\mathsf{lbl}(m), \mathsf{sl}(m)) \in \exp(\mathsf{mf}(\boldsymbol{\sigma}))$. Besides:

Proposition 12.1.19. *Consider $\boldsymbol{\sigma} : A$ parallel innocent. Then, the function*

$$\begin{array}{rccl} \Psi & : & \boldsymbol{\sigma} & \to & \exp(\mathsf{mf}(\boldsymbol{\sigma})) \\ & & m & \mapsto & (\mathsf{lbl}(m), \mathsf{sl}(m)) \end{array}$$

yields a positive isomorphism $\Psi : \boldsymbol{\sigma} \approx \exp(\mathsf{mf}(\boldsymbol{\sigma}))$.

Proof. Injective. We prove that if $\mathsf{lbl}(m) = \mathsf{lbl}(m')$ and $\mathsf{sl}(m) = \mathsf{sl}(m')$, then $m = m'$. Indeed if $\mathsf{lbl}(m) = \mathsf{lbl}(m')$, by definition this means there are two symmetries

$$\theta_m : [m]_{\boldsymbol{\sigma}} \cong_{\boldsymbol{\sigma}} [\mathsf{lbl}(m)]_{\boldsymbol{\sigma}}\,, \qquad \theta_{m'} : [m']_{\boldsymbol{\sigma}} \cong_{\boldsymbol{\sigma}} [\mathsf{lbl}(m)]_{\boldsymbol{\sigma}}$$

which we may compose to get $\theta = \theta_{m'}^{-1} \circ \theta_m : [m]_\sigma \cong_\sigma [m']_\sigma$. Now, since $\mathsf{sl}(m) = \mathsf{sl}(m')$, it follows that $\partial_\sigma \theta$ preserves indices of negative events, so that it is positive by $+$-*transparent*. Hence, θ is an identity and $[m]_\sigma = [m']_\sigma$ since σ is *thin*.

Surjective. Consider $(n, \alpha) \in \exp(\mathsf{mf}(\sigma))$, then we construct by induction on m some $m \in \sigma$ s.t. $\mathsf{lbl}(m) = n$ and $\mathsf{sl}(m) = \alpha$, along with a symmetry $\theta : [m]_\sigma \cong_\sigma [n]_\sigma$. If $(n, \alpha) \rightarrowtriangle_{\exp(\mathsf{mf}(\sigma))} (n', \alpha')$ positive, by necessity $\alpha' = \alpha$. Then θ extends to $\theta' : [m']_\sigma \cong_\sigma [n']_\sigma$ so that $\mathsf{lbl}(m') = n'$, and it is direct that $\mathsf{sl}(m') = \alpha$ as required. If (n', α') is negative, then by $\sim$-*receptive* and *wide*, there is a unique $m \rightarrowtriangle_\sigma m'$ such θ extends with (m, m'), and $\mathsf{ind}(\partial_\sigma(m)) = \alpha'(m')$ – so that $\mathsf{lbl}(m') = n'$ and $\mathsf{sl}(m') = \alpha'$.

Remaining verifications are direct, positivity following from $+$-*transparent*. □

Altogether, we have proved:

Theorem 12.1.20. *Consider A a mixed board. The operations* $\mathsf{mf}(-)$ *and* $\exp(-)$ *inform a one-to-one correspondence between parallel innocent strategies on A (up to positive iso), and* meager *parallel innocent strategies on A (up to positive iso).*

Proof. The statement packages Propositions 12.1.16, 12.1.17 and 12.1.19. □

Thus, just as for traditional alternating innocent strategies, parallel innocent strategies have a compact representation via the *meager form*. This is important conceptually: in traditional alternating innocent strategies, the meager form may be regarded as *syntax* as meager sequential innocent strategies are alternative representations of a syntactic notion of normal form for PCF, the *PCF Böhm trees* – we reviewed this correspondence in Section 3.2.4. Here, by analogy, meager parallel innocent strategies may also be regarded as syntax, but *DAG-shaped* rather than tree-shaped.

We included Theorem 12.1.20 because of its conceptual importance, however we shall not rely on it in our definability theorem: we shall only use the notion of *meager form* insofar as it gives an adequate notion of *finiteness* for parallel innocent strategies, that extends the *finiteness* of traditional innocent strategies reviewed in Section 3.2.4.

12.1.3 The Meager Form of Sequential Globularity

Here we take a small detour, and prove Theorem 11.3.11 which we postponed until we had access to the meager form of parallel innocent strategies. We recall the statement:

Theorem 11.3.11. *The induced functor* $\downarrow\uparrow\text{-}\mathsf{Unf}(-) : \rightarrowtriangle\text{-}\mathbf{SeqGlob}/\approx \to \downarrow\uparrow\text{-}\mathbf{InnWB}$ *is full and faithful: for all A, B mixed boards, the alternating unfolding induces a bijection*

$$\downarrow\uparrow\text{-}\mathsf{Unf}(-) \quad : \quad \rightarrowtriangle\text{-}\mathbf{SeqGlob}(A, B)/\approx \quad \simeq \quad \downarrow\uparrow\text{-}\mathbf{InnWB}(\underline{A}, \underline{B})\,.$$

Proof. Faithful. Consider $\sigma, \tau : A$ sequential globular on mixed board A, and assume $\downarrow\uparrow\text{-}\mathsf{Unf}_!(\sigma^!) = \downarrow\uparrow\text{-}\mathsf{Unf}_!(\tau^!)$. In particular, they have the same P-views – recall from Proposition 11.3.6 that P-views of $\downarrow\uparrow\text{-}\mathsf{Unf}_!(\sigma^!)$ are exactly those of the form

$$(\overset{\frown}{-})(\partial_\sigma([m]_\sigma))$$

for all events $m \in \sigma$. But by Lemma 12.1.18, there is a unique

$$\theta_m : [m]_\sigma \cong_\sigma [\mathsf{lbl}(m)]_\sigma$$

with $\mathsf{lbl}(m) \in \mathsf{mf}(\sigma)$. From that, it is clear that m and $\mathsf{lbl}(m)$ generate the same P-view, *i.e.* $(\overset{\frown}{-})(\partial_\sigma([m]_\sigma)) = (\overset{\frown}{-})(\partial_\sigma([\mathsf{lbl}(m)]_\sigma)$. Therefore, the P-views of $\downarrow\uparrow\text{-}\mathsf{Unf}_!(\sigma^!)$ can be obtained simply from the events of the meager form of σ. Moreover, for any P-view s on $\underline{A}$, there is at most one $m \in \mathsf{mf}(\sigma)$ generating s – if they were two, they would be symmetric by Lemma 11.2.4, and thus equal by Proposition 12.1.5. Thus, any P-view of $\downarrow\uparrow\text{-}\mathsf{Unf}_!(\sigma^!) = \downarrow\uparrow\text{-}\mathsf{Unf}_!(\tau^!)$ is witnessed by *exactly one* event of $\mathsf{mf}(\sigma)$, and *exactly one* event of $\mathsf{mf}(\tau)$. This entails a bijection $\varphi : |\mathsf{mf}(\sigma)| \simeq |\mathsf{mf}(\tau)|$, and it is a direct verification that this entails a positive isomorphism $\varphi : \mathsf{mf}(\sigma) \approx \mathsf{mf}(\tau)$. By Theorem 12.1.20, this entails $\sigma \approx \tau$ as well – from this follows that $\downarrow\uparrow\text{-}\mathsf{Unf}(-)$ is faithful.

Full. Now, consider $\sigma : \underline{A}$ an innocent well-bracketed alternating strategy on $\underline{A}$. For every non-empty even-length P-view s of σ, choose a copy index $\sharp s \in \mathbb{N}$ so that this is injective. We then specify a meager causal innocent strategy with events

$$|\sigma| = \{s \in \sigma \mid s \text{ non-empty P-view}\},$$

with causality the prefix ordering, no conflict, trivial symmetries. We set the display map by induction: for odd-length sa^+b^-, we set it to the unique event b' with $\mathsf{lbl}(b') = b$, $\mathsf{pred}(b') = \partial_\sigma(sa^+)$, and $\mathsf{ind}(b') = 0$ – this event exists by *wide*. For even-length $s = s_1a^-s_2b^+$ where b^+ points to a^-, we set the display to the unique event b' with $\mathsf{lbl}(b') = b$, $\mathsf{pred}(b') = \partial_\sigma(s_1a^-)$ and $\mathsf{ind}(b') = \sharp s$. Altogether, it is easily verified that this defines a meager sequential innocent strategy on A and that $(\overset{\frown}{-})(\partial_\sigma([s]_\sigma)) = s$. Hence the P-views of $\downarrow\uparrow\text{-}\mathsf{Unf}_!(\sigma^!)$ are exactly those of σ by Proposition 11.3.6, thus $\downarrow\uparrow\text{-}\mathsf{Unf}_!(\sigma^!) = \sigma$ by Proposition 3.2.13 – from this follows that $\downarrow\uparrow\text{-}\mathsf{Unf}(-)$ is full. □

So that indeed, globular causal strategies are conservative extensions of traditional well-bracketed innocent strategies. Moreover, we established that by showing that the meager form of globular strategies specializes to the meager form of Section 3.2.4.

12.1.4 Finite Tests Suffice

Back on track, we now introduce a notion of *finiteness* for parallel innocent strategies:

Definition 12.1.21. *Consider A a mixed board, and $\sigma : A$ a parallel innocent strategy. Then σ is **finite** iff $|\mathsf{mf}(\sigma)|_+ = \{s \in |\mathsf{mf}\sigma| \mid \mathrm{pol}(s) = +\}$ is finite.*
*In that case, the **size** of σ is the cardinal of $|\mathsf{mf}(\sigma)|_+$.*

As in the case of traditional innocent strategies, our forthcoming definability result will concern only *finite* parallel innocent strategies. So that finite definability does entail intensional full abstraction, we must show that if two strategies can be distinguished by a parallel innocent strategy, then they can be distinguished by a *finite* one.

Any parallel innocent strategy can be approximated by finite ones:

Proposition 12.1.22. *Consider A a mixed board, and $\sigma : A$ parallel innocent. Writing*

$$D = \{\tau : A \mid \tau \text{ finite } \& \ \tau \lhd \sigma\}$$

yields a directed set of finite parallel innocent strategies, such that $\sigma = \vee D$.

Proof. By definition, $\vee D \lhd \sigma$. For the other direction, pick $m \in \sigma$. Considering

$$|\tau| = \{n \in |\sigma| \mid \forall p^+ \leq_\sigma n,\ \mathsf{mf}(p^+) \leq_\sigma \mathsf{mf}(m)\}\,,$$

it is easily verified that inheriting from σ all additional components makes $\tau : A$ a parallel innocent strategy s.t. $\tau \lhd \sigma$. Moreover $m \in \tau$, and τ is finite since all positive events of $\mathsf{mf}(\tau)$ are below $\mathsf{mf}(m)$. Altogether $\tau \in D$ and $m \in \tau$, so $m \in \vee D$. □

The development above concerns strategies on mixed boards, but recall that morphisms in $\multimap\!\triangleright$-**Strat** are strategies *between* mixed boards: a $\sigma \in \multimap\!\triangleright$-**Strat**$(A, B)$ is a well-bracketed causal strategy on $!A \vdash B$. Nevertheless the above directly extends: $\sigma \in \multimap\!\triangleright$-**Inn**$(A, B)$ is **finite** if $\Lambda(\sigma) : A \Rightarrow B$ is. From the above, we get:

Corollary 12.1.23. *Consider A a mixed board, and $\sigma_1, \sigma_2 : A$ parallel innocent.*
If there is $\alpha : !A \vdash \mathbf{U}$ parallel innocent such that

$$\alpha \odot_! \sigma_1 \Downarrow, \qquad \alpha \odot_! \sigma_2 \Uparrow,$$

then there is $\alpha' \lhd \alpha$ finite and parallel innocent, such that

$$\alpha' \odot_! \sigma_1 \Downarrow, \qquad \alpha' \odot_! \sigma_2 \Uparrow\ .$$

Finally if α is globular (resp. well-bracketed), then so is α'.

Proof. By Proposition 12.1.22, $\Lambda(\alpha) = \vee D$ for D some directed set of finite parallel innocent strategies – automatically innocent, as this is inherited from α. The result then directly follows by continuity of composition of causal strategies; preservation of well-bracketing and globularity are immediate. □

12.2 Factorization

Our next goal is to prove a definability result for finite globular strategies, but it shall be more complex than in the traditional case. Indeed, unlike in Section 3.2.4, globular strategies have no "first Player move" to reproduce first syntactically. Hence we organize our definability process differently. Its core is a *factorization result* (Corollary 12.2.18): namely, that every finite globular $\alpha : !(\&A_i) \vdash \mathbb{X}$ may be obtained as

$$\alpha \quad \equiv \quad \mathsf{fo}(\alpha) \odot_! \langle x_i\, \alpha_{k,1}\, \ldots\, \alpha_{k,p_i} \mid i \in I,\ k \in K_i \rangle\,, \tag{12.1}$$

with $\mathsf{fo}(\alpha)$ a strategy on a *first-order type* and $\alpha_{k,j}$ strictly smaller. This reduces finite definability to that for finite *first-order* strategies, dealt with in Section 12.3.1.

We first extract the components mentioned in (12.1): the *first-order substrategy* $\mathsf{fo}(\alpha)$, and the *argument substrategies* $\alpha_{k,j}$. We use as illustration the meager strategy with typical linear configurations in Figure 12.5. The *first-order sub-strategy*, in red, has events those depending on no Opponent question besides the initial move: it is independent of Opponent's exploration of the arguments, and is purely first-order. The Player questions

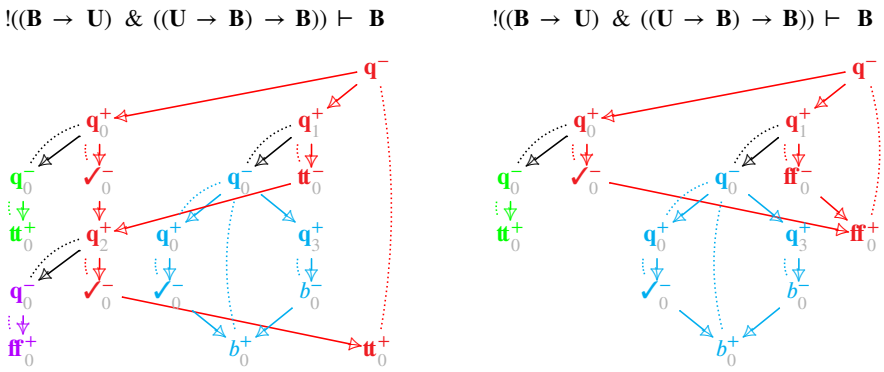


Figure 12.5: A finite meager globular strategy

$$f : \mathbb{B} \to \mathbb{U}, g : (\mathbb{U} \to \mathbb{B}) \to \mathbb{B} \vdash$$
$$\mathbf{let} \left(\begin{array}{l} x = f\ \mathbf{tt} \\ y = g\,(\lambda z^{\mathbb{U}}.\ \mathbf{let} \left(\begin{array}{l} u = z \\ v = g\,\bot \end{array} \right) \mathbf{in}\ v) \end{array} \right) \mathbf{in}\,(\mathbf{if}\ y\ \mathbf{then}\,(f\ \mathbf{ff};\ \mathbf{tt})\ \mathbf{else}\ \mathbf{ff}) : \mathbb{B}$$

Figure 12.6: Factorization and definability for Figure 12.5

in this first-order part play a special role; we call them *primary questions*. Intuitively, they correspond to occurrences of variables not appearing in an argument to a variable call. In Figure 12.5, the primary questions are $\mathbf{q}^+_0$, $\mathbf{q}^+_1$ and $\mathbf{q}^+_2$. Depending on their type, the primary questions admit arguments that Opponent can access by playing questions pointing to them. Parts of the strategy accessed in this way are the *argument sub-strategies* – in Figure 12.5 there are three, respectively prompted by (*i.e.* causally depending on) $\mathbf{q}^-_{0,t}$, $\mathbf{q}^-_{2,k}$ and $\mathbf{q}^-_{1,r}$ and colored accordingly.

The strategy of Figure 12.5 is exactly definable; as illustration we show the term in Figure 12.6, with subterms colored so as to match the four components of the strategy[2].

The proof of factorization is organized as follows: in Section 12.2.1 we extract the first-order part, in Section 12.2.2, we extract the argument substrategies, and in Section 12.2.3, we recompose them to obtain the original strategy back.

12.2.1 The Flow Substrategy

Shallow substrategy. Consider A a type, and $\boldsymbol{\alpha} : [\![A]\!]$ a finite globular strategy.

Necessarily A has form $A_1 \to \cdots \to A_n \to \mathbb{X}$ for $A_i = A_{i,1} \to \cdots \to A_{i,p_i} \to \mathbb{X}_i$.

[2]In the end, our definability process will not quite give the term of Figure 12.6 but a sequential version as we only know how to define first-order strategies in general up to *positional equivalence* – see Section 12.3.1.

Recall $\mathbb{X}, \mathbb{X}_i$ range over ground types, *i.e.* $\mathbb{U}$, $\mathbb{B}$ and $\mathbb{N}$. Up to currying, we write

$$\boldsymbol{\alpha} \,:\, !(\&_{1\leq i\leq n} A_i) \vdash \mathbf{X}\,,$$

omitting semantic brackets. We often shorten the left hand side part to $!(\&A_i)$, and reuse A for the board $!(\&A_i) \vdash \mathbf{X}$. Now, we start with the *shallow substrategy*. First:

Definition 12.2.1. *Consider an event $m \in \boldsymbol{\alpha}$.*
*Then m is **shallow** iff $[m]_{\boldsymbol{\alpha}}$ contains exactly one Opponent question.*

In Figure 12.5, the shallow events are exactly those in red. Note that if m is shallow, then the Opponent question in $[m]_{\boldsymbol{\alpha}}$ is necessarily the initial Opponent question. As one can dive within the A_is only via an Opponent question, this means that m and its causal history must remain in the *first-order* part of the arena.

The shallow substrategy is obtained by restricting $\boldsymbol{\alpha}$ to shallow events:

Proposition 12.2.2. *Set $\mathsf{sh}(\boldsymbol{\alpha})$ with the shallow events of $\boldsymbol{\alpha}$, other components inherited.*
Then, $\mathsf{sh}(\boldsymbol{\alpha}) \,:\, !(\&\mathbf{X}_i) \vdash \mathbf{X}$ is finite globular.

Proof. Write $\mathsf{sh}(A)$ for the board $!(\&\mathbf{X}_i) \vdash \mathbf{X}$. First, for each $m \in \mathsf{sh}(\boldsymbol{\alpha})$, $\partial_{\boldsymbol{\alpha}}(m) \in \mathsf{sh}(A)$: indeed, the least events in A but not in $\mathsf{sh}(A)$ are Opponent questions. For *extension*, as symmetries are order-isos preserving polarities and Q/A labeling, they preserve $\mathsf{sh}(\boldsymbol{\alpha})$. The conditions for a finite globular strategy are direct from $\boldsymbol{\alpha}$. □

This captures the strategy induced by the red part of Figure 12.5.

The flow substrategy Eventually we wish to reconstruct $\boldsymbol{\alpha}$ from its first-order and argument substrategies using the composition mechanisms available on causal strategies. But we cannot hope to do that merely with $\mathsf{sh}(\boldsymbol{\alpha})$. Indeed, two Player questions in $\mathsf{sh}(\boldsymbol{\alpha})$ playing in the same component $\mathbf{X}_i$ must receive the same argument substrategy if $\boldsymbol{\alpha}$ is to be reassembled compositionally – but there is no reason why two Player questions in the same component should always have the same argument substrategy!

Thus rather than working with $\mathsf{sh}(\boldsymbol{\alpha})$, we need to relabel it to send distinct (non-symmetric) Player questions to distinct components. First we define:

Definition 12.2.3. *A **primary question** of $\boldsymbol{\alpha}$ is any $q^{\mathcal{Q},+} \in \mathsf{mf}(\mathsf{sh}(\boldsymbol{\alpha}))$. We write $\mathcal{Q}$ for the set of primary questions, and $\mathcal{Q}_i$ for the primary questions displaying to $\mathbf{X}_i$.*

By construction, $\mathcal{Q} = \uplus_{1\leq i\leq n} \mathcal{Q}_i$ – as observed above, $\mathcal{Q}$ is finite. A $m^{\mathcal{Q},+} \in \mathsf{sh}(\boldsymbol{\alpha})$ might not be a primary question, but it must necessarily be symmetric to $\mathsf{lbl}(m) \in \mathcal{Q}$ a primary question (as its *label*, *i.e.* its representative in the meager form).

Definition 12.2.4. *The **flow substrategy** of $\boldsymbol{\alpha}$, written*

$$\mathsf{flow}(\boldsymbol{\alpha}) \,:\, \bigotimes_{1\leq i\leq n} \bigotimes_{q\in\mathcal{Q}_i} !\mathbf{X}_i \vdash \mathbf{X}$$

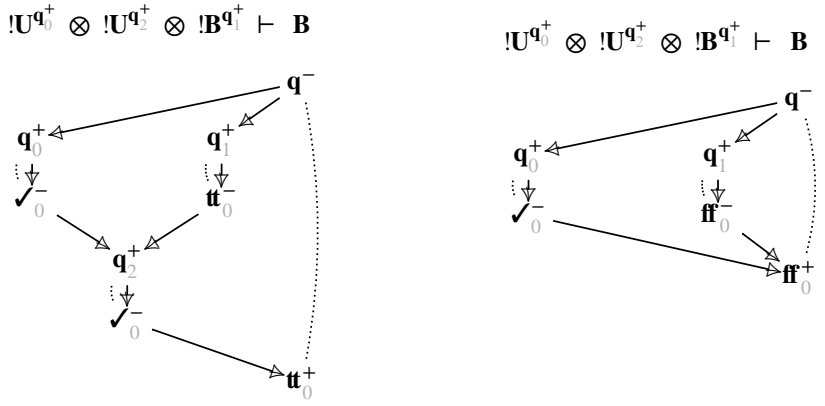


Figure 12.7: Configurations of the flow substrategy for Figure 12.5

is obtained as $\mathsf{sh}(\boldsymbol{\alpha})$ *with the display map* $\partial_{\mathsf{sh}(\boldsymbol{\alpha})}$ *replaced with:*

$$\begin{array}{rclcrcl}
\partial_{\mathsf{flow}(\boldsymbol{\alpha})}(m^{\mathcal{Q},-}) & = & (2,a) & \textit{if} & \partial_{\mathsf{sh}(\boldsymbol{\alpha})}(m^{\mathcal{Q},-}) & = & (2,a) \\
\partial_{\mathsf{flow}(\boldsymbol{\alpha})}(m^{\mathcal{A},+}) & = & (2,a) & \textit{if} & \partial_{\mathsf{sh}(\boldsymbol{\alpha})}(m^{\mathcal{A},+}) & = & (2,a) \\
\partial_{\mathsf{flow}(\boldsymbol{\alpha})}(m^{\mathcal{Q},+}) & = & (1,(i,(\mathsf{lbl}(m),(j,a)))) & \textit{if} & \partial_{\mathsf{sh}(\boldsymbol{\alpha})}(m^{\mathcal{Q},+}) & = & (1,(j,(i,a))) \\
\partial_{\mathsf{flow}(\boldsymbol{\alpha})}(m^{\mathcal{A},-}) & = & (1,(i,(\mathsf{j}(\mathsf{lbl}(m)),(j,a)))) & \textit{if} & \partial_{\mathsf{sh}(\boldsymbol{\alpha})}(m^{\mathcal{A},-}) & = & (1,(j,(i,a)))\,,
\end{array}$$

i.e. *sending each* $q \in \mathcal{Q}$ *and its answers to the copy of* $\mathbf{X}_i$ *specified by indices* i, q.

It is a finite globular strategy (though not a morphism in $\multimap$-**Strat**, but the same definition applies nonetheless). Figure 12.7 shows typical configurations of the flow substrategy for Figure 12.5, tagging each component by a primary question.

12.2.2 The Argument Substrategies

Next, we focus on the *higher-order* structure, aiming to extract the arguments to (the variable calls corresponding to) the primary questions. A slight complication comes from the fact that because answers in the shallow substrategy can be replicated, $\boldsymbol{\alpha}$ actually comprises many copies of each argument substrategy – we solve that by temporarily only considering the part of $\boldsymbol{\alpha}$ where shallow answers have copy index 0:

Definition 12.2.5. *An event* $m \in \boldsymbol{\alpha}$ *is* ***canonical*** *iff for all shallow* $n^{\mathcal{A},-} \leq_{\boldsymbol{\alpha}} m$, $\mathsf{ind}(\partial_{\boldsymbol{\alpha}}(n)) = 0$. *We write* $\|\boldsymbol{\alpha}\|$ *the set of canonical events of* $\boldsymbol{\alpha}$.

A primary question is always canonical; so are all events in $\mathsf{mf}(\boldsymbol{\alpha})$. Now, fix a primary question $q \in \mathcal{Q}_i$. It displays to an initial event in A_i, which is:

$$!A_{i,1} \multimap \ldots \multimap \; !A_{i,p_i} \multimap \mathbf{X}_i\,.$$

Argument substrategies are accessed by Opponent questions pointing to (moves symmetric to) primary questions. Up to symmetry, there are p_i distinct Opponent questions

pointing to q, matching the p_i arguments of A_i. From now on, if $q \in \mathcal{Q}_i$ is a primary question and $q \rightarrowtriangle_{\sigma} m^{Q,-}$ an Opponent question, we shall say that m is **in component** j if it displays to an initial move of $!A_{i,j}$. For $q \in \mathcal{Q}_i$ and $1 \leq j \leq p_i$, we shall extract the *argument sub-strategy* $\alpha_{q,j}$ initiated by Opponent questions pointing to q in component j. As the strategy provides the information for an argument of A_i it must live in $A_{i,j}$; but it can still access the context, so we aim for a finite globular strategy:

$$\boldsymbol{\alpha}_{q,j} \; : \; !(\&A_i) \vdash \; !A_{i,j} \; .$$

Extracting events. To do this, we assign to all events of $\boldsymbol{\alpha}$ *tags*, as follows:

Definition 12.2.6. *Consider $m \in \|\boldsymbol{\alpha}\|$. We write:*

$$\begin{array}{lcll} m & \Subset & \mathsf{sh}(\boldsymbol{\alpha}) & \Leftrightarrow \quad m \text{ is shallow,} \\ m & \Subset & \boldsymbol{\alpha}_{q,j} & \Leftrightarrow \quad \text{there is } q \rightarrowtriangle_{\boldsymbol{\alpha}} n^{Q,-} \text{ in component } j, \text{ such that } n \leq_{\boldsymbol{\alpha}} m, \end{array}$$

where in the second clause, $q \in \mathcal{Q}_i$ is a primary question and $1 \leq j \leq p_i$.

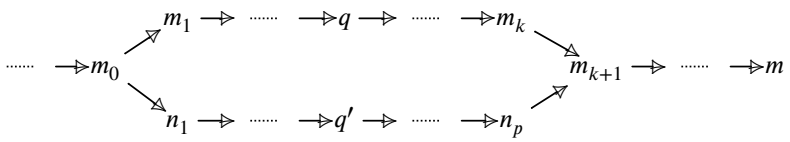
Any $m \in \|\boldsymbol{\alpha}\|$ receives a tag as either in $\mathsf{sh}(\boldsymbol{\alpha})$; or in one of the argument sub-strategies. Crucially, each event receives *exactly one tag* – this is subtle and involves all our structural constraints on strategies. For instance, without *globular*, in Figure 10.19 the move $\mathbf{q}_2^+$ would be tagged for *two* distinct argument sub-strategies.

Lemma 12.2.7. *Every $m \in \|\boldsymbol{\alpha}\|$ receives exactly one tag following Definition 12.2.6.*

Proof. First, each $m \in \|\boldsymbol{\alpha}\|$ receives *at least* one tag. Any $[m]_{\boldsymbol{\alpha}}$ contains at least one Opponent question: the initial move. If it contains exactly one Opponent question, then m is shallow by definition. Assume there are at least two. Take $n^{Q,-} \leq_{\boldsymbol{\alpha}} m$ minimal such that it is a non-initial Opponent question. Then its immediate predecessor must be some primary question $q \in \mathcal{Q}_i$; and so there is $1 \leq j \leq p_i$ s.t. $m \Subset \boldsymbol{\alpha}_{q,j}$.

We prove that m receives *at most* one tag. Clearly if $m \Subset \boldsymbol{\alpha}_{q,j}$ for some $q \in \mathcal{Q}_i$ and $1 \leq j \leq p_i$, there are at least two Opponent questions in $[m]_{\boldsymbol{\alpha}}$ so we cannot have $m \Subset \mathsf{sh}(\boldsymbol{\alpha})$. But m could be in two argument substrategies: assume $m \Subset \boldsymbol{\alpha}_{q,j}$ and $m \Subset \boldsymbol{\alpha}_{q',j'}$ for $q \in \mathcal{Q}_i$, $q' \in \mathcal{Q}_{i'}$, $1 \leq j \leq p_i$ and $1 \leq j' \leq p_{i'}$. We first show that $q = q'$; seeking a contradiction assume they are distinct. But q and q' cannot be comparable: if $q \leq_{\boldsymbol{\alpha}} q'$, $[q']_{\boldsymbol{\alpha}}$ has at least two Opponent questions, contradicting $q' \in \mathsf{sh}(\boldsymbol{\alpha})$.

Take $\rho \rightarrowtriangle m, \rho' \rightarrowtriangle m \in \mathsf{gcc}(\boldsymbol{\alpha})$, respectively passing through q and q'. We draw:



and since q, q' are distinct, the diagram may be chosen with $X = \{m_1, \ldots, m_k\}$ and $Y = \{n_1, \ldots, n_p\}$ disjoint. By *pre-innocence*, m_1 and n_1 are positive. Moreover, since the board is forestial and alternating, it is immediate from Lemma 6.1.16 that m_{k+1} is positive, and m_k, n_p positive. Altogether we have a *globule*, so X and Y are complete by globularity – and moreover the diagram has the shape as in (10.2), and in particular q

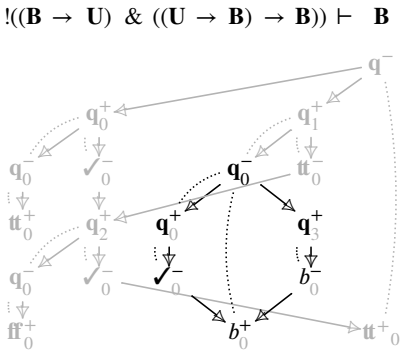


Figure 12.8: Ev. $|\boldsymbol{\alpha}_{\mathbf{q}^+_1,1}|$ of Figure 12.5

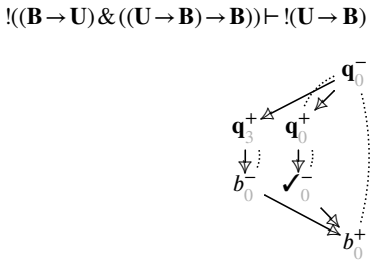


Figure 12.9: Its reassigment in $\boldsymbol{\alpha}_{\mathbf{q}^+_1,1}$

is answered in ρ. Additionally as observed in (10.2), writing $q = m_i, m_{i+1}$ must answer q. But since $m \Subset \boldsymbol{\alpha}_{q,j}$, $q \rightarrowtriangle_{\boldsymbol{\alpha}} n^{Qu,-} \leq_{\boldsymbol{\alpha}} m$ where n is a negative question in component j. Then necessarily, $n = m_{i+1}$, or we get a contradiction with parallel innocence. Thus m_{i+1} is both a question and an answer, contradiction, so $q = q'$.

Finally, if j and j' were distinct, there would be necessarily distinct $q \rightarrowtriangle_{\boldsymbol{\alpha}} n_1^{Q,-}$ in component j and $n_2^{\mathbf{q},-}$ in component j' such that $n_1, n_2 \leq_{\boldsymbol{\alpha}} m$, but this is in direct contradiction with *pre-innocence*. Thus $j = j'$, concluding the proof. □

This shows any $m \in \|\boldsymbol{\alpha}\|$ can always be attributed to exactly *one* of the sub-strategies we wish to extract. Accordingly, the *argument sub-strategies* will have events

$$|\boldsymbol{\alpha}_{q,j}| = \{m \in \|\boldsymbol{\alpha}\| \mid m \Subset \boldsymbol{\alpha}_{q,j}\},$$

completed to ess by inheriting the components from $\boldsymbol{\alpha}$, as will be made explicit later.

Assigning display map. The main challenge in extracting the argument sub-strategies is to define what shall be their new display maps. Indeed, $\partial_{\boldsymbol{\alpha}}$ still displays events in $\boldsymbol{\alpha}_{q,j}$ to moves in $!(\&A_i) \vdash \mathbf{X}$, whereas $\partial_{\boldsymbol{\alpha}_{q,j}}$ must display then to $!(\&A_i) \vdash !A_{i,j}$.

For this, we must give special care to those events that correspond to the new $!A_{i,j}$ component on the right hand side, reindexing them as illustrated in Figures 12.8 and 12.9. For this we split $|\boldsymbol{\alpha}_{q,j}|$ in two subsets: on the one hand, some events depend *statically*, *i.e.* with respect to $\leq_A$ (through $\partial_{\boldsymbol{\alpha}}$) on the primary question q – in Figure 12.8, those are $\mathbf{q}^-_0, \mathbf{q}^+_0, \checkmark^-_0$ and b^+_0. On the other hand, the remaining events must follow from new calls to variables in the context – in Figure 12.8, those are $\mathbf{q}^+_3$ and b^-_0. These two subsets are treated differently when defining the new display map: the former are left unchanged, while the latter are reindexed as in Figure 12.9.

It will be helpful to have notations for the canonical embeddings of the set of moves $|A_i|$ and $|A_{i,j}|$ into $|A|$. More precisely, for each primary question $q \in \mathcal{Q}_i$, we write

$$\iota_q(-) : |A_i| \to |A|$$

the injection adding the sequence of tags addressing A_i within A, originating from the tagged disjoint unions in the board constructions – in particular, ι_q maps the initial move of A_i to $\partial_{\boldsymbol{\alpha}}(q)$. Likewise, $\iota_{q,j} : |A_{i,j}| \to |A|$ addresses the j-th argument of q. Then:

Definition 12.2.8. *We define a display map for $\boldsymbol{\alpha}_{q,j}$ by setting, for $m \in \boldsymbol{\alpha}_{q,j}$:*

$$\begin{array}{lcll} \partial_{\boldsymbol{\alpha}_{q,j}}(m) & = & \iota_{\curlyvee}(a) & \text{if } \partial_{\boldsymbol{\alpha}}(m) = \iota_{q,j}(a), \\ \partial_{\boldsymbol{\alpha}_{q,j}}(m) & = & \partial_{\boldsymbol{\alpha}}(m) & \text{otherwise,} \end{array}$$

where $\iota_{\curlywedge}(a) = (1, a)$ and $\iota_{\curlyvee}(a) = (2, a)$.

Altogether, this lets us extract $\boldsymbol{\alpha}_{q,j}$ as intended:

Proposition 12.2.9. *Consider $q \in \mathcal{Q}_i$ and $1 \leq j \leq p_i$. The **argument substrategy** for q, j is $(|\boldsymbol{\alpha}_{q,j}|, \leq_{q,j}, \#_{q,j}, \mathcal{S}(\boldsymbol{\alpha}_{q,j}), \partial_{\boldsymbol{\alpha}_{q,j}})$, with components $\leq_{q,j}$ and $\#_{q,j}$ the restrictions of $\boldsymbol{\alpha}$, $\mathcal{S}(\boldsymbol{\alpha}_{q,j}) = \{\theta \cap |\boldsymbol{\alpha}_{q,j}|^2 \mid \theta \in \mathcal{S}(\boldsymbol{\alpha})\}$, and $\partial_{\boldsymbol{\alpha}_{q,j}}$ in Definition 12.2.8.*

Then, $\boldsymbol{\alpha}_{q,j} : !(\&A_i) \vdash !A_{i,j}$ is a globular finite strategy.

Proof. A lengthy but routine verification. □

Removing the bang. But we must obtain a morphism in $\multimap$-**Glob**, and thus eliminate the bang on the right hand side. This is done by composition with *dereliction*, *i.e.*

$$\boldsymbol{\alpha}^{\bullet}_{q,j} = \mathsf{der}_{A_{i,j}} \odot \boldsymbol{\alpha}_{q,j} : !(\&A_i) \vdash A_{i,j}\,,$$

it only remains to show that this satisfies the conditions which will later on allow us to proceed from $\boldsymbol{\alpha}^{\bullet}_{q,i}$ with the inductive definability argument:

Proposition 12.2.10. *Consider $q \in \mathcal{Q}_i$ a primary question, and $1 \leq j \leq p_i$.*

Then, $\boldsymbol{\alpha}^{\bullet}_{q,j}$ is finite globular with size strictly lesser than $\boldsymbol{\alpha}$.

Proof. Using Proposition 7.3.1, it is easy that $\boldsymbol{\alpha}^{\bullet}_{q,j}$ is positively isomorphic to the strategy obtained from $\boldsymbol{\alpha}_{q,j}$ by restricting the initial Opponent question to copy index 0. This also informs an injection of $\mathsf{mf}(\boldsymbol{\alpha}^{\bullet}_{q,j})$ into $\mathsf{mf}(\boldsymbol{\alpha})$ not reaching the primary question q, from which follows the announced size constraint. □

Note that $\boldsymbol{\alpha}_{q,j}$ is easily recovered from $\boldsymbol{\alpha}^{\bullet}_{q,j}$, as we have

$$\boldsymbol{\alpha}_{q,j} \quad \approx \quad (\mathsf{der}_{A_{i,j}} \odot \boldsymbol{\alpha}_{q,j})^{!} = (\boldsymbol{\alpha}^{\bullet}_{q,j})^{!} \tag{12.2}$$

via the "bang lemma" (Lemma 10.3.7) and by definition of $\boldsymbol{\alpha}^{!}_{q,j}$.

12.2.3 Recomposition

Having extracted from $\boldsymbol{\alpha}$ its flow substrategy and its argument substrategies, our next aim is to prove that $\boldsymbol{\alpha}$ can be reconstructed from those.

Reconstructing $\boldsymbol{\alpha}$. From the original strategy $\boldsymbol{\alpha} : !(\&A_i) \vdash \mathbf{X}$, we have now extracted the *flow substrategy* and a family of *argument substrategies*:

$$\begin{array}{rcll} \mathsf{flow}(\boldsymbol{\alpha}) & : & \otimes_{1\leq i\leq n} \otimes_{q\in\mathcal{Q}_i} !\mathbf{X}_i \vdash \mathbf{X} & \\ \boldsymbol{\alpha}^{\bullet}_{q,j} & : & !(\&A_i) \vdash A_{i,j} & \text{for each } q \in \mathcal{Q}_i \text{ and } 1 \leq j \leq p_i. \end{array}$$

For each $q \in \mathcal{Q}_i$, we first form a strategy $\boldsymbol{\alpha}_q : !(\&A_i) \vdash \mathbf{X}_i$ gathering all the arguments for the primary question q. This is constructed from the argument substrategies, using the internal language corresponding to the cartesian closed structure of $\multimap$-**Inn**:

$$\boldsymbol{\alpha}_q \quad = \quad x_1 : A_1, \ldots, x_n : A_n \vdash x_i\, \boldsymbol{\alpha}^{\bullet}_{q,1} \ldots \boldsymbol{\alpha}^{\bullet}_{q,p_i} : \mathbf{X}_i\,.$$

Then, we plug each $\boldsymbol{\alpha}_q$ onto the corresponding primary question in the flow substrategy. This is done relying on the relative Seely category structure of **NTCG-Inn**:

$$\mathsf{recomp}(\boldsymbol{\alpha}) \quad = \quad \mathsf{flow}(\boldsymbol{\alpha}) \odot (\otimes_{1\leq i\leq n} \otimes_{q\in\mathcal{Q}_i} \boldsymbol{\alpha}^{!}_{q}) \odot \boldsymbol{\delta}_{\&A_i} \quad : \quad !(\&A_i) \vdash \mathbf{X}\,,$$

where for B a strict board and $n \in \mathbb{N}$, we write $\boldsymbol{\delta}_B : !B \vdash (!B)^{\otimes n}$ for the obvious strategy (leaving n implicit). In the sequel we may only write $\boldsymbol{\delta}$.

Positions of $\mathsf{recomp}(\boldsymbol{\alpha})$. Recall that the *positional equivalence* $\equiv$, studied in Section 10.4, consists in listing the zero-payoff symmetry classes of configurations reached by (+-covered configurations of) a visible strategy. We expect that $\mathsf{recomp}(\boldsymbol{\alpha}) \approx \boldsymbol{\alpha}$, but we shall only prove $\mathsf{recomp}(\boldsymbol{\alpha}) \equiv \boldsymbol{\alpha}$ – this is simpler as positions compose relationally. We shall make use of the syntax for positions on $-$-boards introduced in Figure 10.17.

We now analyse the positions of the recomposition (12.2.3). We start with:

Lemma 12.2.11. *Consider B a strict $-$-board.*
Then, the positions $(\!|\boldsymbol{\delta}_B|\!)$ of $\boldsymbol{\delta}_B : !B \vdash (!B)^{\otimes n}$ are exactly those

$$\left(\sum_{1\leq i\leq n} \mathsf{x}_i \vdash \otimes_{1\leq i\leq n} \mathsf{x}_i \right) \in (\!|!B \vdash (!B)^{\otimes n}|\!)$$

where $\mathsf{x}_i \in (\!|!B|\!)$ for all $1 \leq i \leq n$.

Proof. By a direct variation of Lemma 6.4.4. □

Next we analyse the positions of $\boldsymbol{\alpha}_q$ for $q \in \mathcal{Q}_i$:

Lemma 12.2.12. *For any $q \in \mathcal{Q}_i$, the positions $(\!|\boldsymbol{\alpha}_q|\!)$ are exactly those of the form*

$$\left([(i, \mathsf{y}_{q,1} \multimap \ldots \multimap \mathsf{y}_{q,p_i} \multimap \mathsf{v}_q)] + \sum_{1 \leq j \leq p_i} \mathsf{x}_{q,j}\right) \vdash \mathsf{v}_q \in (\!|!(\&A_i) \vdash \mathbb{X}_i|\!)$$

where for each $1 \leq j \leq p_i$, $(\mathsf{x}_{q,j} \vdash \mathsf{y}_{q,j}) \in (\!|\boldsymbol{\alpha}_{q,j}|\!)$, and for $\mathsf{v}_q \in (\!|\mathbb{X}_i|\!)$.

Proof. From the laws of relative Seely categories, $\boldsymbol{\alpha}_q$ is positively isomorphic to

$$\begin{array}{ccc} !(\&A_i) & \xrightarrow{\delta} & \otimes_{p_i+1} !(\&A_i) \\ & & \downarrow{\scriptstyle (\otimes_{1 \leq j \leq p_i} (\boldsymbol{\alpha}^{\bullet}_{q,j})^{!}) \otimes [\![x_i]\!]} \\ & & (\otimes_{1 \leq j \leq p_i} !\, A_{i,j}) \otimes A_i \xrightarrow{\mathsf{ev}} \mathbb{X}_i \end{array}$$

in **NTCG**. The lemma follows from Corollary 10.4.15, Lemma 12.2.11, (12.2)and characterisations analogous to Lemma 12.2.11 for other copycat-like strategies. □

From all those, we may characterise the positions of $\mathsf{recomp}(\boldsymbol{\alpha})$ as

Corollary 12.2.13. *The positions of* $\mathsf{recomp}(\boldsymbol{\alpha})$ *are exactly those of the form*

$$\sum_{1 \leq i \leq n} \sum_{q \in \mathcal{Q}'_i} \left([(i, \mathsf{y}_{q,1} \multimap \ldots \multimap \mathsf{y}_{q,p_i} \multimap \mathsf{v}_q)] + \sum_{1 \leq j \leq p_i} \mathsf{x}_{q,j}\right) \vdash \mathsf{v} \quad \in \quad (\!|!(\&A_i) \vdash \mathbb{X}|\!)$$

where for all $1 \leq i \leq n$, $\mathcal{Q}'_i$ is a subset of $\mathcal{Q}_i$, and:

$$\left(\otimes_{1 \leq i \leq n} \otimes_{q \in \mathcal{Q}'_i} \mathsf{v}_q \vdash \mathsf{v}\right) \in (\!|\mathsf{flow}(\boldsymbol{\alpha})|\!)\,, \qquad \left((\mathsf{x}_{q,j} \vdash \mathsf{y}_{q,j}) \in (\!|\boldsymbol{\alpha}_{q,j}|\!)\right)_{q \in \mathcal{Q}'_i, 1 \leq j \leq p_i}\,.$$

Proof. Direct from Lemmas 12.2.11, 12.2.12 and Corollary 10.4.15. □

Positions of $\boldsymbol{\alpha}$. Next, we must similarly characterise the positions of $\boldsymbol{\alpha}$.

We call **stopping configurations** of $\boldsymbol{\alpha}$ those $x \in \mathscr{C}^+(\boldsymbol{\alpha})$ such that $\kappa(\partial_{\boldsymbol{\alpha}}(x)) = 0$ (*i.e.* those configurations that contribute a point to the relational collapse), and such that all Opponent answers in x have copy index 0. Write $\mathsf{stop}(\boldsymbol{\alpha})$ for this set; our first observation is that the relational collapse of $\boldsymbol{\alpha}$ is entirely spanned by stopping configurations – the restriction to the copy index of Opponent answers has no consequence.

Lemma 12.2.14. *Consider $\mathsf{x}_\Gamma \vdash \mathsf{x}_\mathbb{X} \in (\!|\boldsymbol{\alpha}|\!)$.*

Then, there is a stopping configuration $x \in \mathsf{stop}(\boldsymbol{\alpha})$ such that $\partial_{\boldsymbol{\alpha}} x \in \mathsf{x}_\Gamma \vdash \mathsf{x}_\mathbb{X}$.

Proof. By definition, there is $y \in \mathscr{C}^+(\boldsymbol{\alpha})$ such that $\partial_{\boldsymbol{\alpha}} y \in \mathsf{x}_\Gamma \vdash \mathsf{x}_\mathbb{X}$. In particular, $\kappa(\partial_{\boldsymbol{\alpha}} y) = 0$. By definition of payoff on ground types, this entails that all questions in y have *exactly one answer*. We construct $z \in \mathscr{C}(A)$ from $\partial_{\boldsymbol{\alpha}}(y)$ using *wide*, by changing the copy index of every negative answer to 0 – by $-$*-transparent*, we obtain a symmetry

$$\theta : \partial_{\boldsymbol{\alpha}}(x) \cong^{-}_{A} z\,,$$

and applying Lemma 7.2.6 we get $y \cong_{\boldsymbol{\alpha}} x$ and $\varphi : \partial_{\boldsymbol{\alpha}} y \cong_A^+ z$. By $+$-*transparent*, φ preserves the copy indices of negative events, thus the indices of negative answers in $\partial_{\boldsymbol{\alpha}} y$ is 0. Finally, as $x \cong_{\boldsymbol{\alpha}} y$ we have $\partial_{\boldsymbol{\alpha}} y \in \mathsf{x}_\Gamma \vdash \mathsf{x}_{\mathbb{X}}$, and y is a stopping configuration. □

Restricting to stopping configurations is important, because events of stopping configurations are *canonical* in the sense of Definition 12.2.5 (*i.e.* the shallow Opponent answers on which they depend have copy index 0) – and argument sub-strategies were extracted from the canonical events of $\boldsymbol{\alpha}$, following Definition 12.2.6. Thanks to this, we shall be able to partition any $x \in \mathsf{stop}(\boldsymbol{\alpha})$ into stopping configurations from the flow substrategy, and the argument sub-strategies. This partitionning is the key argument to link the positions of $\boldsymbol{\alpha}$ with the positions of its sub-strategies.

Let us perform this partitioning. We define components of the partition as

$$\begin{array}{rcl} \mathsf{sh}(x) & = & \{m \in x \mid m \Subset \mathsf{sh}(\boldsymbol{\alpha})\} \\ x_{q,j} & = & \{m \in x \mid m \Subset \boldsymbol{\alpha}_{q,j}\}\,, \quad (\text{for } q \in \mathcal{Q}_i \text{ and } 1 \leq j \leq p_i) \end{array}$$

additionally we write $\mathcal{Q}^x = \mathcal{Q} \cap x$, and $\mathcal{Q}_i^x$ likewise. We have:

Lemma 12.2.15. *Consider* $x \in \mathsf{stop}(\boldsymbol{\alpha})$.
Then $\mathsf{sh}(x) \in \mathsf{stop}(\mathsf{sh}(\boldsymbol{\alpha}))$, $x_{q,j} \in \mathsf{stop}(\boldsymbol{\alpha}_{q,j})$ *and* x *partitions as*

$$x = \mathsf{sh}(x) \uplus \left(\biguplus_{1 \leq i \leq n} \biguplus_{q \in \mathcal{Q}_i^x} \biguplus_{1 \leq j \leq p_i} x_{q,j} \right),$$

moreover this decomposition is compatible with display maps, in the sense that

$$\partial_{\boldsymbol{\alpha}}(x) = \left(\biguplus_{1 \leq i \leq n} \biguplus_{q \in \mathcal{Q}_i^x} \left(\left[\iota_q(z_{q,1} \multimap \ldots \multimap z_{q,p_i} \multimap v_q) \right] \uplus \left[\biguplus_{1 \leq j \leq p_i} \iota_\ell(y_{q,j}) \right] \right) \right) \uplus \iota_r(v)\,,$$

where we have, for all $1 \leq i \leq n$, $q \in \mathcal{Q}_i^x$ *and* $1 \leq j \leq p_i$*:*

$$\partial_{\mathsf{flow}(\boldsymbol{\alpha})}(x) = \iota_\ell \left(\bigotimes_{1 \leq i \leq n} \bigotimes_{q \in \mathcal{Q}_i^x} v_q \right) \uplus \iota_r(v) \qquad \partial_{\boldsymbol{\alpha}_{q,j}}(x_{q,j}) = \iota_\ell(y_{q,j}) \uplus \iota_r(z_{q,j})\,.$$

Proof. From Lemma 12.2.7, lengthy but straightforward verifications. □

Reciprocally, any compatible collection of stopping configurations from the various substrategies may be assembled into a stopping configuration of $\boldsymbol{\alpha}$:

Lemma 12.2.16. *For* $x \in \mathsf{stop}(\mathsf{sh}(\boldsymbol{\alpha}))$ *and* $(x_{q,j} \in \mathsf{stop}(\boldsymbol{\alpha}_{q,j}))_{1 \leq i \leq n, q \in \mathcal{Q}_i^x, 1 \leq j \leq p_i}$,

$$x \uplus \left(\biguplus_{1 \leq i \leq n} \biguplus_{q \in \mathcal{Q}_i^x} \biguplus_{1 \leq j \leq p_i} x_{q,j} \right) \in \mathsf{stop}(\boldsymbol{\alpha})\,.$$

Proof. Lengthy but straightforward verifications. □

From this we may finally conclude the proof of factorization:

Corollary 12.2.17. *The strategies $\boldsymbol{\alpha}$ and $\mathsf{recomp}(\boldsymbol{\alpha})$ are positionally equivalent.*

Proof. Altogether, Lemmas 12.2.15 and 12.2.16 ensure that stopping configurations of $\boldsymbol{\alpha}$ are in one-to-one correspondence with the data of stopping configurations $x \in \mathsf{stop}(\mathsf{sh}(\boldsymbol{\alpha}))$ and families $(x_{q,j})_{1\leq i\leq n, q\in \mathcal{Q}_i^x, 1\leq j\leq p_i}$, in a way compatible with the display maps as stated in Lemma 12.2.15. Taking symmetry classes, we derive the same characterisation of complete positions of $\boldsymbol{\alpha}$ as that of $\mathsf{recomp}(\boldsymbol{\alpha})$ in Corollary 12.2.13. □

Syntactic factorization. Finally, we reformulate this relying on the cartesian closed structure only. The **first-order substrategy** $\mathsf{fo}(\boldsymbol{\alpha}) \in \mathbf{NTCG}_!(\&_{1\leq i\leq n}\&_{q\in\mathcal{Q}_i}\mathbf{X}_i, \mathbf{X})$ is obtained in the obvious way from $\mathsf{flow}(\boldsymbol{\alpha})$ using the relative Seely category structure. Using Corollary 12.2.17, Proposition 12.2.10, and laws of a relative Seely category:

Corollary 12.2.18. *Any globular strategy $\boldsymbol{\alpha} : !(\&A_i) \vdash \mathbf{X}$ factors as*

$$\boldsymbol{\alpha} \equiv \mathsf{fo}(\boldsymbol{\alpha}) \odot_! \langle x_i\, \boldsymbol{\alpha}^\bullet_{q,1}\, \ldots\, \boldsymbol{\alpha}^\bullet_{q,p_i} \mid 1 \leq i \leq n,\ q \in \mathcal{Q}_i \rangle\,,$$

where $\mathsf{fo}(\boldsymbol{\alpha}) : !(\&\mathbf{X}_i) \vdash \mathbf{X}$ and $\boldsymbol{\alpha}^\bullet_{q,j} : !(\&A_i) \vdash A_{i,j}$ are globular.
If $\boldsymbol{\alpha}$ is finite, so are $\mathsf{fo}(\boldsymbol{\alpha})$ and $\boldsymbol{\alpha}^\bullet_{q,j}$, the latter with size strictly lesser than that of $\boldsymbol{\alpha}$.

12.3 Finite Definability and Full Abstraction

12.3.1 First-Order Definability

Assuming we are in the process of an inductive definability procedure, we may assume that we have terms for the argument sub-strategies $\boldsymbol{\alpha}^\bullet_{q,j}$. Thus, it remains to define

$$\mathsf{fo}(\boldsymbol{\alpha}) : !(\&\mathbf{X}_i) \vdash \mathbf{X}$$

the first-order substrategy.

Strict definability. This naturally asks the question: does finite definability hold for first-order globular strategies? It is fairly easy to see that the answer is negative:

Proposition 12.3.1. *The strategy with meager form pictured in Figure 12.10 is finite globular, but not definable in $\mathsf{PCF}_{/\!/}$.*

Sketch. By rewriting means, one may establish that if it was definable in $\mathsf{PCF}_{/\!/}$, then it would be definable just via parallel compositions of variables. However, such terms composed only of parallel compositions and variables are easily seen to yield series-parallel causal shapes; thus the example in Figure 12.10 is undefinable. □

For strategies on the game $!(\&\mathbf{U}) \vdash \mathbf{U}$, it seems that we may get finite definability up to positive isomorphism by restricting globular strategies further to those that have a series-parallel causal shape. But for more complex datatypes, requiring a series-parallel causal shape is not enough to get back definability: an example phenomenon appears in

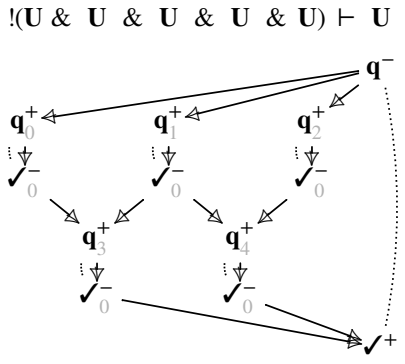


Figure 12.10: An undefinable strategy

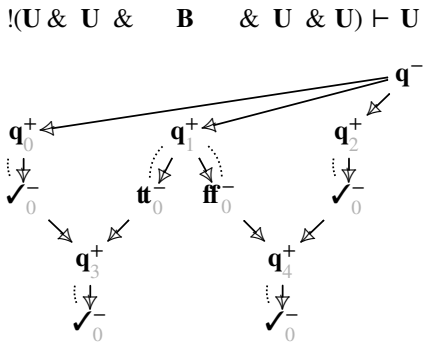


Figure 12.11: Dynamic causal shape

Figure 12.11, where the structure of forking/merging threads is impacted by the values. Though we do not have a proof, it seems clear that this is undefinable: in a term as in

$$\mathbf{let} \begin{pmatrix} x_1 & = & M_1 \\ x_2 & = & M_2 \\ x_3 & = & M_3 \end{pmatrix} \mathbf{in}\ N$$

then the first move in N will depend on the values obtained for x_1, x_2 and x_3: there is no way to indicate that *e.g.* if x_2 is true, then N should not wait for x_1.

Altogether, we have not been able to characterise the globular strategies definable in PCF$_{/\!/}$. A natural option would be to reify the first-order globular strategies, *i.e.* to consider them as syntax and add a new syntactic construct for every first-order globular strategy. This seems interesting, because first-order globular strategies are indeed relevant from a computational point of view; however it seems hard to give then an appealing syntactic theory. Instead, we shall weaken our objective and only perform definability *up to positional equivalence* – which will suffice for our purposes[3].

12.3.2 Positional First-Order Definability

Moving from positive isomorphism to positional equivalence is convenient, because positional equivalence may not respect the evaluation order. This lets us construct a purely sequential version of any finite globular first-order strategy.

From now on, fix a first-order finite globular strategy $\boldsymbol{\alpha} : !(\&\mathbb{X}_i) \vdash \mathbb{X}$.

Necessary primary question. Once prompted, $\boldsymbol{\alpha}$ may launch many computations in parallel, via *minimal primary questions* (*i.e.* that only depend on the Opponent initial

[3]Of course, the factorization result of Corollary 12.2.18 is already up to positional equivalence, but this is mostly for convenience: factorization should hold up to positive isomorphism.

question). The idea is that our sequential definability procedure will choose any of those (the choice does not matter up to positional equivalence), and evaluate it first. However, we must show that all these primary questions are actually *necessary*, in the sense that any stopping configuration necessarily includes this primary question, answered.

Lemma 12.3.2. *Consider* $\boldsymbol{\alpha} \in \multimap\!\triangleright\text{-}\mathbf{Glob}(\&\mathbb{X}_i, \mathbb{X})$ *finite globular.*
Then for each minimal primary question q *and* $x \in \mathsf{stop}(\boldsymbol{\alpha})$*, we have* $q \in x$*.*

Proof. Consider $x \in \mathsf{stop}(\boldsymbol{\alpha})$ and seeking a contradiction, assume $q \notin x$. By *causal determinism*, q is compatible with x. Consider $t \in \mathscr{L}(x)$ any linearization of x, yielding

$$s = \overset{\frown}{\cdot}(\partial_{\Lambda(\boldsymbol{\alpha})}(t)) \in \circlearrowleft\text{-}\mathsf{Unf}(\boldsymbol{\alpha}),$$

it is straightforward that it is (logically) well-bracketed. Now, $\boldsymbol{\alpha}$ is a morphism in $\multimap\!\triangleright\text{-}\mathbf{Strat}$, and must thus be well-bracketed in the sense of Definition 9.3.2. Therefore,

$$sa^+ = \overset{\frown}{\cdot}(\partial_{\Lambda(\boldsymbol{\alpha})}(tq)) \in \circlearrowleft\text{-}\mathsf{Unf}(\boldsymbol{\alpha})$$

should be (logically) well-bracketed. But it is not, since q points to the initial Opponent question which must already be answered in x: a contradiction, thus $q \in x$. □

Thus we may safely begin the sequentialization with any minimal primary question: by this lemma, we know that it will appear in all stopping configurations.

Residual. Choose $q \in \mathcal{Q}_{i_0}$ minimal. As q appears in every stopping configuration, it is safe to first make a call to x_{i_0}, then branch on the possible return values. Since $\boldsymbol{\alpha}$ is finite, there is a finite set V of values leading to an observable result. Now, for each $v \in V$, we define $\boldsymbol{\alpha}_{(qv)}$ the *residual* of $\boldsymbol{\alpha}$ after q yields value v; and then proceed inductively. To define this residual, our first step is to rename $\boldsymbol{\alpha}$ to isolate this first call:

Lemma 12.3.3. *There is* $\boldsymbol{\alpha}_{(q)} \in \multimap\!\triangleright\text{-}\mathbf{Glob}((\&\mathbf{X}_i)\&\mathbf{X}_{i_0}, \mathbf{X})$ *with the same ess as* $\boldsymbol{\alpha}$ *and differing only via its display map, such that:*

(1) *for* $\mathsf{x} \vdash \mathsf{w} \in \mathbf{(}\boldsymbol{\alpha}_{(q)}\mathbf{)}$*, then* $\mathsf{x} = \mathsf{x}' + [(n+1, \mathsf{v})]$ *such that* $\mathsf{x}' + [(i_0, \mathsf{v})] \vdash \mathsf{w} \in \mathbf{(}\boldsymbol{\alpha}\mathbf{)}$*,*
(2) *for* $\mathsf{x} \vdash \mathsf{w} \in \mathbf{(}\boldsymbol{\alpha}\mathbf{)}$*, then* $\mathsf{x} = \mathsf{x}' + [(i_0, \mathsf{v})]$ *such that* $\mathsf{x}' + [(n+1, \mathsf{v})] \vdash \mathsf{w} \in \mathbf{(}\boldsymbol{\alpha}_{(q)}\mathbf{)}$*.*

Proof. The strategy $\boldsymbol{\alpha}_{(q)}$ has the same components as $\boldsymbol{\alpha}$, with display map sending q and its answers to the new component. The characterisation of positions is direct. □

Note that in the second clause, the decomposition $\mathsf{x} = \mathsf{x}' + [(i_0, \mathsf{v})]$ is not unique: there may be several values in component i, but one of them yields a position of $\boldsymbol{\alpha}_{(q)}$.

We set the residual $\boldsymbol{\alpha}_{(qv)}$ as $\boldsymbol{\alpha}_{(qv)} = \boldsymbol{\alpha}_{(q)} \odot_! \langle \mathsf{der}_{\&\mathbf{X}_i}, v \rangle : !(\&\mathbf{X}_i) \vdash \mathbf{X}$ writing $v : !(\&\mathbf{X}_i) \vdash \mathbf{X}_{i_0}$ the constant strategy. In order to characterize its positions, we note:

Lemma 12.3.4. *For any* $v \in V$*, the positions of* $\langle \mathsf{der}_{\&\mathbf{X}_i}, v \rangle^!$ *are exactly of the form*

$$\mathsf{x} \vdash \mathsf{x} + p \cdot [(n+1, \mathsf{v})] \quad \in \quad \mathbf{(}!(\&\mathbf{X}_i) \vdash !((\&\mathbf{X}_i)\&\mathbf{X}_{i_0})\mathbf{)},$$

where $p \cdot [(n+1, \mathsf{v})]$ *denotes the p-fold sum, and for any* $\mathsf{x} \in \mathbf{(}!(\&\mathbf{X}_i)\mathbf{)}$ *and* $p \in \mathbb{N}$*.*

Proof. Straightforward verification. □

Using this lemma, we link the positions of $\boldsymbol{\alpha}$ and $\boldsymbol{\alpha}_{(qv)}$.

Lemma 12.3.5. *Consider positions* $\mathsf{x} \in \mathbf{(}!(\&\mathbf{X}_i)\mathbf{)}$ *and* $\mathsf{w} \in \mathbf{(X)}$.
Then, $\mathsf{x} \vdash \mathsf{w} \in \mathbf{(}\boldsymbol{\alpha}\mathbf{)}$ *iff* $\mathsf{x} = \mathsf{x}' + [(i_0, v)]$ *with* $\mathsf{x}' \vdash \mathsf{w} \in \mathbf{(}\boldsymbol{\alpha}_{(qv)}\mathbf{)}$.

Proof. By definition, we have $\boldsymbol{\alpha}_{(qv)} = \boldsymbol{\alpha}_{(q)} \odot \langle \mathsf{der}_{\&\mathbf{X}_i}, v\rangle^!$, so by Corollary 10.4.15,

$$\mathbf{(}\boldsymbol{\alpha}_{(qv)}\mathbf{)} = \mathbf{(}(\boldsymbol{\alpha}_{(q)})\mathbf{)} \odot \mathbf{(}\langle \mathsf{der}_{\&\mathbf{X}_i}, v\rangle^!\mathbf{)} .$$

The lemma directly follows by Lemmas 12.3.3 and 12.3.4. □

Reconstruction. We now rebuild a term. We have $\boldsymbol{\alpha}_{(qv)}$ a globular finite strategy with size strictly lesser than that of $\boldsymbol{\alpha}$. By induction hypothesis, there is a term

$$x_1 : \mathbb{X}_1, \dots, x_n : \mathbb{X}_n \vdash N_{(qv)} : \mathbb{X}_{i_0} ,$$

for each $v \in V$, such that $[\![N_{(qv)}]\!] \equiv \boldsymbol{\alpha}_{(qv)}$. We set $x_1 : \mathbb{X}_1, \dots, x_n : \mathbb{X}_n \vdash M : \mathbb{X}$ as

$$\begin{array}{l}\mathbf{case}\ x_{i_0}\ \mathbf{of}\\ \quad v_1 \mapsto N_{(qv_1)}\\ \quad v_2 \mapsto N_{(qv_2)}\\ \quad \cdots\\ \quad v_p \mapsto N_{(qv_p)}\end{array} \quad \stackrel{\text{def}}{=} \quad \begin{array}{l}\mathbf{let}\ x = x_{i_0}\ \mathbf{in}\\ \qquad \mathbf{if}\ x =_{\mathbb{X}} v_1\ \mathbf{then}\ N_{(qv_1)}\\ \qquad \mathbf{else\,if}\ x =_{\mathbb{X}} v_2\ \mathbf{then}\ N_{(qv_2)}\\ \qquad \cdots\\ \qquad \mathbf{else\,if}\ x =_{\mathbb{X}} v_p\ \mathbf{then}\ N_{(qv_p)}\\ \qquad\qquad \mathbf{else}\ \bot\end{array}$$

where $V = \{v_1, \dots, v_p\}$, using the syntactic sugar introduced in Section 2.2.3. Write

$$x_1 : \mathbb{X}_1, \dots, x_n : \mathbb{X}_n, x : \mathbb{X}_i \vdash M' : \mathbb{X}$$

for the iterated if statement, *i.e.* M is $\mathbf{let}\ x = x_{i_0}\ \mathbf{in}\ M'$.

Analysis. It remains to analyze the positions of $[\![M]\!]$ and $[\![M']\!]$ to show that $[\![M]\!] \equiv \boldsymbol{\alpha}$ as required. We start by an analysis of the positions of $[\![M']\!]$.

Lemma 12.3.6. *Consider positions* $\mathsf{x} \in \mathbf{(}!(\&\mathbf{X}_i)\mathbf{)}$ *and* $\mathsf{w} \in \mathbf{(X)}$.
Then, $\mathsf{x} \vdash \mathsf{w} \in \mathbf{(}[\![N_{(qv)}]\!]\mathbf{)}$ *iff there is* $p \in \mathbb{N}$ *such that* $\mathsf{x}+p\cdot[(n+1, \mathsf{v})] \vdash \mathsf{w} \in \mathbf{(}[\![M']\!]\mathbf{)}$.

Proof. It is a direct verification, amounting to the correctness of our definition for equality test and the usual laws for conditionals, that for any $v \in V$ we have $[\![M']\!] \odot_! \langle \mathsf{der}, v\rangle = [\![N_{(qv)}]\!]$. The claim then follows by Corollary 10.4.15 and Lemma 12.3.4. □

It remains to take the interpretation of the **let** construction into account. Recall that

$$[\![M]\!] \quad = \quad \mathsf{let}_{\mathbf{X}_{i_0},\mathbf{X}} \odot_! \langle \pi_{i_0}, \Lambda^!([\![M']\!])\rangle \quad : \quad !(\&\mathbf{X}_i) \vdash \mathbf{X}$$

where $\mathsf{let}_{\mathbf{X}_{i_0},\mathbf{X}} : !(\mathbf{X}_{i_0}\&(!\,\mathbf{X}_i \multimap \mathbf{X})) \vdash \mathbf{X}$. The positions of $\mathsf{let}_{\mathbf{X}_{i_0},\mathbf{X}}$ are as follows:

Lemma 12.3.7. *The positions of* $\mathsf{let}_{\mathbf{X}_{i_0},\mathbf{X}}$ *are exactly those of the form*

$$[(1,\mathsf{v})] + [(2,((p\cdot[\mathsf{v}])\multimap \mathsf{w}))] \vdash \mathsf{w} \quad \in \quad \mathbf{(}(!(\mathbf{X}_{i_0}\&(!\mathbf{X}_i\multimap \mathbf{X}))\vdash \mathbf{X})\mathbf{)}$$

for $\mathsf{v}\in\mathbf{(X}_{i_0}\mathbf{)}$, $\mathsf{w}\in\mathbf{(X)}$, *and* $p\in\mathbb{N}$.

Proof. A direct analysis of positions reached by stopping configurations of $\mathsf{let}_{\mathbf{X}_{i_0},\mathbf{X}}$. □

We can now wrap up, showing that $[\![M]\!]$ has the same non-empty positions as $\boldsymbol{\alpha}$.

Lemma 12.3.8. *Consider* $\mathsf{x}\in\mathbf{(}\&\mathbf{X}_i\mathbf{)}$ *and* $\mathsf{w}\in\mathbf{(X)}$.
Then, $\mathsf{x}\vdash\mathsf{w}\in\mathbf{(}[\![M]\!]\mathbf{)}$ *iff* $\mathsf{x}=\mathsf{x}'+[(i_0,\mathsf{v})]$ *with* $\mathsf{x}'\vdash\mathsf{w}\in\mathbf{(}[\![N_{(qv)}]\!]\mathbf{)}$.

Proof. By Lemmas 12.3.6 and 12.3.7. □

Note that this exactly matches the characterisation of positions of $\boldsymbol{\alpha}$ in Lemma 12.3.5. As $\boldsymbol{\alpha}_{(qv)}\equiv[\![N_{(qv)}]\!]$ by induction hypothesis, it follows that $[\![M]\!]\equiv\boldsymbol{\alpha}$. Summing up, we have proved finite first-order definability up to positional equivalence:

Theorem 12.3.9. *For* $\boldsymbol{\alpha}\in{\multimap\!\triangleright}\text{-}\mathbf{Glob}((\&\mathbf{X}_i),\mathbf{X})$ *finite, there is*

$$x_1:\mathbb{X}_1,\dots,x_n:\mathbb{X}_n\vdash M:\mathbb{X}$$

a term of PCF *(not using parallel evaluation) such that* $[\![M]\!]\equiv\boldsymbol{\alpha}$.

12.3.3 Finite Globular Definability

We may now conclude the proof of finite definability.

Theorem 12.3.10. *Let* $\Gamma\vdash A$ *be a* PCF *typing judgment,* $\boldsymbol{\alpha}\in{\multimap\!\triangleright}\text{-}\mathbf{Glob}([\![\Gamma]\!],[\![A]\!])$ *finite.*
Then, there is $\Gamma\vdash M:A$ *such that* $[\![M]\!]\equiv\boldsymbol{\alpha}$.

Proof. Up to currying, $\boldsymbol{\alpha}:!(\&_{1\le i\le n}A_i)\vdash\mathbf{X}$, writing $A_i=A_{i,1}\to\cdots\to A_{i,p_i}\to\mathbb{X}_i$ for $1\le i\le n$. We reason by induction on the size of $\boldsymbol{\alpha}$. By Corollary 12.2.18,

$$\boldsymbol{\alpha}\equiv\mathsf{fo}(\boldsymbol{\alpha})\odot_!\langle x_i\,\boldsymbol{\alpha}^\bullet_{q,1}\ \dots\ \boldsymbol{\alpha}^\bullet_{q,p_i}\mid 1\le i\le n,\ q\in\mathcal{Q}_i\rangle\,,$$

with each $\mathcal{Q}_i$ finite, and for $1\le i\le n$, $q\in\mathcal{Q}_i$ and $1\le j\le p_i$, $\boldsymbol{\alpha}^\bullet_{q,j}:!(\&A_i)\vdash A_{i,j}$ a globular finite strategy of size strictly smaller than $\boldsymbol{\alpha}$. By induction hypothesis, there is

$$x_1:A_1,\dots,x_n:A_n\vdash N_{q,j}:A_{i,j}$$

s.t. $[\![N_{q,j}]\!]\equiv\boldsymbol{\alpha}_{q,j}$. Let us write $\mathcal{Q}_i=\{q_{i,1},\dots,q_{i,k_i}\}$. By Theorem 12.3.9 there is

$$x_{q_{1,1}}:\mathbb{X}_1,\dots,x_{q_{1,k_1}}:\mathbb{X}_1,\dots,x_{q_{n,1}}:\mathbb{X}_n,\dots,x_{q_{n,k_n}}:\mathbb{X}_n\quad\vdash\quad M_{\mathsf{fo}}:\mathbb{X}$$

such that $[\![M_{\mathsf{fo}}]\!]\equiv\mathsf{fo}(\boldsymbol{\alpha})$. Then, we define the term $x_1:A_1,\dots,x_n:A_n\vdash M:\mathbb{X}$ as

$$x_1:A_1,\dots,x_n:A_n\vdash M_{\mathsf{fo}}[x_i\,N_{q_{i,l},1}\ \dots\ N_{q_{i,l},p_i}/x_{q_{i,l}}]:\mathbb{X}\,.$$

Then we may finally compute

$$\begin{aligned}
[\![M]\!] &= [\![M_{\mathsf{fo}}]\!] \odot_! \langle [\![x_i\ N_{q_{i,l},1}\ \ldots\ N_{q_{i,l},p_i}]\!] \mid 1 \leq i \leq n, q_{i,l} \in \mathcal{Q}_i \rangle \\
&\equiv [\![M_{\mathsf{fo}}]\!] \odot_! \langle x_i\ [\![N_{q,1}]\!]\ \ldots\ [\![N_{q,n}]\!] \mid 1 \leq i \leq n, q \in \mathcal{Q}_i \rangle \\
&\equiv \mathsf{fo}(\boldsymbol{\alpha}) \odot_! \langle x_i\ \boldsymbol{\alpha}^{\bullet}_{q,1}\ \ldots\ \boldsymbol{\alpha}^{\bullet}_{q,p_i} \mid 1 \leq i \leq n,\ q \in \mathcal{Q}_i \rangle
\end{aligned}$$

using the substitution lemma for cartesian closed categories, properties of the internal language, the properties of M_{fo} and $N_{q,j}$ and that $\equiv$ is a congruence. □

12.3.4 Intensional Full Abstraction

We may now prove our final full abstraction result:

Theorem 12.3.11. *The model* $\multimap\!\triangleright$-**Glob** *is intensionally fully abstract for* $\mathsf{PCF}_{/\!/}$.

Proof. Consider $\vdash M, N : A$ such that $M \simeq N$, and assume that $[\![M]\!] \not\approx [\![N]\!]$, *i.e.* there is a test $\boldsymbol{\alpha} \in \multimap\!\triangleright$-**Glob**$([\![A]\!], [\![\mathbb{U}]\!])$ such that, *w.l.o.g.*, $\boldsymbol{\alpha} \odot_! [\![M]\!] \Downarrow$ and $\boldsymbol{\alpha} \odot_! [\![N]\!] \Uparrow$.

By Corollary 12.1.23, we assume $\boldsymbol{\alpha}$ is additionally finite. By Theorem 12.3.9, there is a term $x : A \vdash T : \mathbb{U}$ such that $[\![T]\!] \equiv \boldsymbol{\alpha}$. Defining the context $C[-] = (\lambda x^A.\, T)\,[-]$, it follows from the laws of cartesian closed categories that

$$[\![C[M]]\!] = [\![(\lambda x^A.\, T)\ M]\!] \approx [\![T[M/x]]\!] \approx [\![T]\!] \odot_! [\![M]\!] \equiv \boldsymbol{\alpha} \odot_! [\![M]\!] \Downarrow,$$

and $[\![C[N]]\!] \equiv \mathsf{comp}(\boldsymbol{\alpha}) \odot_! [\![N]\!]$. By Theorem 9.1.5, $C[M] \Downarrow$. By hypothesis $M \simeq N$, so $C[N] \Downarrow$. By Theorem 9.1.5 again, $[\![C[N]]\!] \Downarrow$, hence $\boldsymbol{\alpha} \odot_! [\![N]\!] \Downarrow$, contradiction. □

Summing up, in Part III we have constructed a cartesian closed $\sim$-category $\multimap\!\triangleright$-**Strat**. We have defined two conditions on morphisms of $\multimap\!\triangleright$-**Strat**, *sequentiality* and *globularity*, forming three sub-cartesian closed $\sim$-categories of $\multimap\!\triangleright$-**Strat**: $\multimap\!\triangleright$-**Seq**, $\multimap\!\triangleright$-**Glob** and $\multimap\!\triangleright$-**SeqGlob**. We established four intensional full abstraction results

$\multimap\!\triangleright$-**Strat**	is fully abstract for	$\mathsf{IA}_{/\!/}$	(Th. 9.3.10)
$\multimap\!\triangleright$-**Strat** + *sequentiality*	is fully abstract for	IA	(Th. 11.2.21)
$\multimap\!\triangleright$-**Strat** + *globularity* + *sequentiality*	is fully abstract for	PCF	(Th. 11.3.12)
$\multimap\!\triangleright$-**Strat** + *globularity*	is fully abstract for	$\mathsf{PCF}_{/\!/}$	(Th. 12.3.11)

therefore achieving our objective to *disentangle parallelism and state*. A significant part of this consisted in establishing interpretation-preserving cartesian closed $\sim$-functors

$\circlearrowleft$-Unf	:	$\multimap\!\triangleright$-**Strat**	$\to$	$\circlearrowleft$-**Strat**	(Theorem 9.3.8)
$\downarrow\uparrow$-Unf	:	$\multimap\!\triangleright$-**Seq**	$\to$	$\downarrow\uparrow$-**Strat**	(Theorem 11.2.18)
$\downarrow\uparrow$-Unf	:	$\multimap\!\triangleright$-**SeqGlob**	$\to$	$\downarrow\uparrow$-**Inn**	(Theorem 11.3.7)
$(-)$	:	$\multimap\!\triangleright$-**Glob**	$\to$	$\mathbf{Rel}_!$	(Corollary 10.4.15)

to previously established models of different nature, which altogether, or so we hope, substantiates our claim that *thin concurrent games* can play a valuable role at the interface of many denotational semantics of programming languages.

Part IV

Other Developments and Openings

Introduction to Part IV

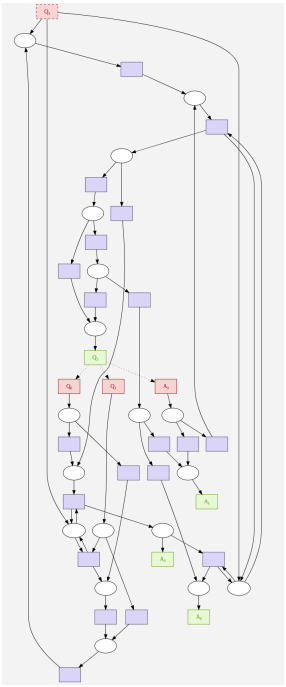


Figure 12.12: Concurrent strategies want cycles too!

In this final part, we conclude the monograph by providing some larger perspective on the landscape of concurrent games. First, in Chapter 13, we shall give a synthesis of a lot of the work done in concurrent games that could not be presented in this document. Then, in Chapter 14 we give some conclusions: we ask to what extent the work done so far substantiates the view evoked in the introduction, of concurrent games as a bridge between interactive semantics. We also then present a few open problems raised in the present development, and conclude with some perspectives.

Chapter 13

Further Work

In this monograph so far, we have given a detailed, technical account of what we consider to be the core results and developments in the theory of (thin) concurrent games. But this is not the full story; and though this monograph is already intimidatingly lengthy, we feel like a survey of the further work accomplished is called for. We show in Figure 13.1 all the works accounted for in this monograph, either in the main technical content or in this survey, grouped by themes and along with the (technical, rather than in terms of mere inspiration) dependencies between them.

This is not intended to be an exhaustive survey of concurrent games, but only of the lines of work that I have been leading. There are other interesting developments in concurrent games in which I did not participate; or for which my participation was minor; or that feel too loosely connected from the main thrust of the present document.

13.1 Effects and Concurrency

This section summarises work that was mostly developed in close collaboration with Simon Castellan, during his PhD thesis (2014-2017).

In 2015, our paper introducing **TCG**, our concept of parallel innocence, and our full abstraction result for parallel evaluation of PCF, was published [Castellan et al., 2015]. At that time, it felt ironic and a little embarassing that we had something called *concurrent games* and yet had not developed semantics for actual concurrent effects.

It *was* very clear for us at the time how to interpret $\mathsf{IA}_{/\!/}$ in **TCG**, but for a while, what was not clear was which theorem to prove! In the denotational semantics literature, the gold standard is full abstraction. We were confident that we could inherit intensional full abstraction by linearizing our model to [Ghica and Murawski, 2008] (though the details of this were not actually written up before this monograph – and they did require some care), but it felt unsatisfactory to only reprove an old result, and at that, one that

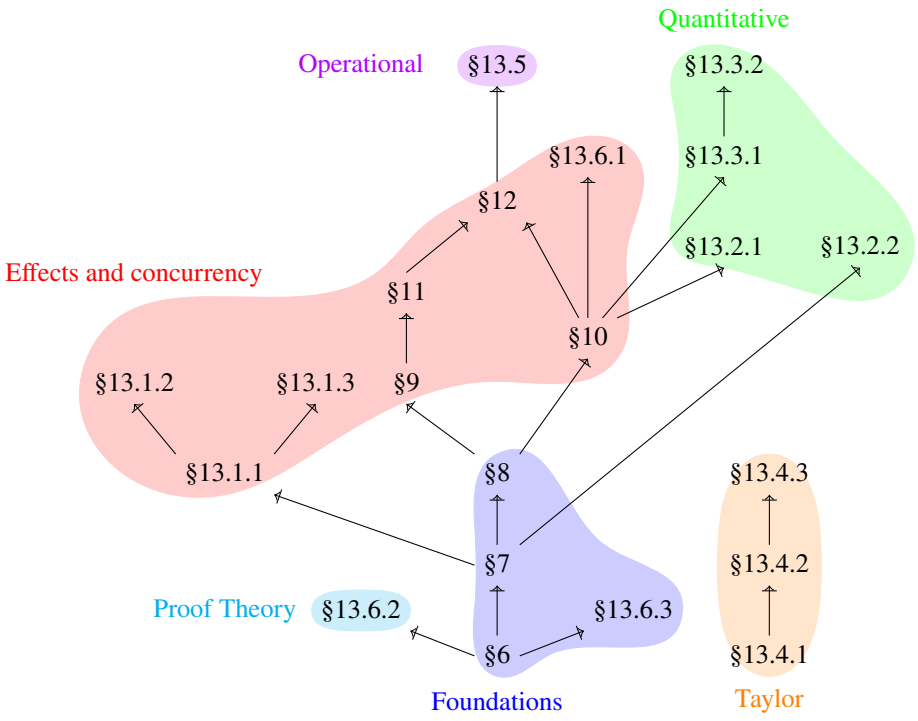


Figure 13.1: The event structure of contributions in concurrent games

does not actually need the causal information. So what is the right nail, for a hammer whose specialty is to uncover structure that by nature is *not* observable?

13.1.1 Non-canonical causal presentation

The mainstream approach, in formalizing the execution of concurrent systems, is by *interleaving* the executions of its sequential components – this is the approach followed by Ghica and Murawski's fully abstract model for $\mathsf{IA}_{/\!/}$ [Ghica and Murawski, 2008].

But considering a concurrent system formed of n sequential processes

$$P_1 \parallel \cdots \parallel P_n\,,$$

even if each P_i performs only *one* atomic action, this induces $n!$ different executions: this is the *state explosion problem*, one of the main challenges to the algorithmic verification of concurrent systems. In contrast, the *truly concurrent* approach followed by concurrent games avoids interleavings and proposes a more compact representation of executions, that may in principle be helpful for algorithmic verification.

This makes concurrent games seem promising as a basis for an observational equivalence checker for $\mathsf{IA}_{/\!/}$, especially in the light of Ghica and Murawski's full abstraction result [Ghica and Murawski, 2008] and the topic of *algorithmic game semantics* [Ghica and McCusker, 2003]. More precisely, the hope was to formalize the intuition that non-alternating strategies in the sense of $\circlearrowleft$-**Strat** already had an implicit causal structure, *i.e.* that a non-alternating strategy interpreting a term M was simply

$$[\![M]\!] = \circlearrowleft\text{-}\mathsf{Unf}(\sigma_M)$$

for σ_M a **canonical causal explanation** for $[\![M]\!]$, a compact representation of the interleavings that may be used instead of the interleavings for algorithmic purposes.

In [Castellan and Clairambault, 2016] we showed that this was *not* the case, already in the affine case. We first built a concurrent games model of an affine version of $\mathsf{IA}_{/\!/}$, a corresponding non-alternating model, and an interpretation-preserving functor from the causal to the interleaving model. Then we showed that a non-alternating strategy could have *distinct* and *incomparable minimal* causal explanations – additionally, we proved this already happens within the interpretation of affine $\mathsf{IA}_{/\!/}$: there are two affine programs with incomparable *causal* behaviour that have the same non-alternating unfolding. This phenomenon is illustrated in Figure 13.2.

This does not show that concurrent games *cannot* help for checking the equivalence of higher-order concurrent programs, it only points out a fundamental obstacle to the route we had in mind. It is also a hint that the value of concurrent games is in its presentation of the *intensional behaviour*, not as an efficient way to present the interleavings. Thus, we then refocused our research on the intensional behaviour.

13.1.2 Non-angelic concurrent games

Another aspect of concurrent games that was clear from the start is that they record explicitly the point of non-deterministic branching. For instance, the program $x : \mathbb{U} \vdash$

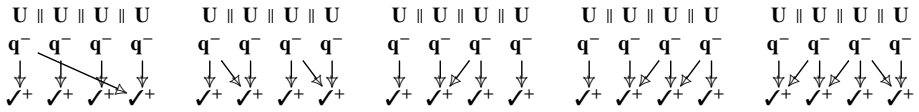


Figure 13.2: As the goal is only to reconstruct interleavings, the branching structure does not matter, so rather than an event structure we can work only with a set of *augmentations*, *i.e.* the partial orders induced by configurations. In this example, both the second and the third augmentation can be removed without changing the interleavings – but no other; yielding two incompatible minimal causal explanations.

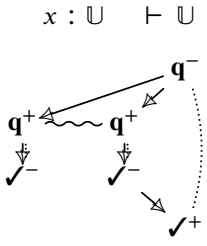


Figure 13.3: Recording non-angelic behaviour

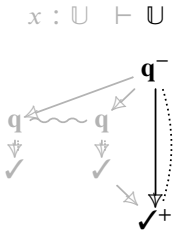


Figure 13.4: After hiding

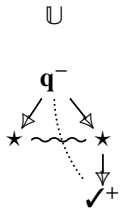


Figure 13.5: Composition with essential events

if choice then x; $\bot$ **else** x; ✓ : $\mathbb{U}$ yields the strategy of Figure 13.3, tracking explicitly the two incompatible calls to x and the fact that one yields to a divergence, while the other yields termination. In most earlier games models (with the notable exception of [Harmer and McCusker, 1999] for Erratic IA), the diverging branch would be ignored: this is a direct consequence of the traditional methodology of representing a program as a *set of plays*, *i.e.* a set of converging behaviour. We only record the converging behaviour, forgetting the non-deterministic branching information, leading to an *angelic* view of termination, *i.e.* may-convergence.

This discussion might lead the reader to expect that concurrent games are sound with respect to must-convergence, but that is not the case: *hiding* removes branches that do not prompt any visible behaviour, even if they witness possible non-deterministic branches. For instance, composing the strategy of Figure 13.3 with **skip** yields the interaction of Figure 13.4 which, after hiding, yields the constant strategy $[\![\mathbf{skip}]\!]$ – because of hiding, we forget that something may have gone wrong in the computation.

Nevertheless, in [Castellan et al., 2018a], we proved that concurrent games are easily amended so as to recall more than angelic information on program behaviour. It is simply a matter of changing *hiding*, so that we retain events that are part of a minimal conflict: those are kept through hiding as events of neutral polarity, dubbed *essential*

events. With this modification, the composition above yields the strategy of Figure 13.5 where the events labelled $\star$ are those essential events. With this interpretation, we keep more than angelic information: in fact, we show that (the transition system generated by) the strategy is *weakly bisimilary* to (the transition system generated by) the operational semantics, for an affine version of $\mathsf{IA}_{/\!/}$. We are not aware of any other work capturing by denotational means such a precise information on program behaviour.

This felt a rather good illustration of the expressivity of concurrent games, and in particular of its feature that it retains the non-deterministic branching information. But, weak bisimulation does not really take causality into account.

13.1.3 Resource-tracking concurrent games

This is a collaboration with Aurore Alcolei during her PhD thesis, and Olivier Laurent.

To explore the expressiveness of causality in concurrent games, we needed to record information about programs that really depends on the causal independence between events, and is incompatible with an interleaving interpretation. As a natural objective, we thought of the following problem. Consider $\mathsf{IA}_{/\!/}$ extended with a primitive $\mathbf{wait}(n)$ that takes a natural number n, and waits for n seconds. The program

$$\vdash \mathbf{newref}\ r\ \mathbf{in}\ \mathbf{let} \begin{pmatrix} x & = & \mathbf{wait}(1);\ r := 1;\ \mathbf{wait}(2) \\ y & = & \mathbf{wait}(2);\ r := 2;\ \mathbf{wait}(1) \end{pmatrix} \mathbf{in}\ !r : \mathbb{N}$$

always evaluates in 6 seconds according to an interleaving operational semantics, as all the **wait** instructions must be handled before termination. But running this program on a multicore architecture, one will observe faster termination times: the program should evaluate to 2 in about 3 seconds, but – if the thread for x gets stuck – we can in principle observe a termination to 1 in at least 4 seconds. This can be captured operationally via a semantics where threads can wait *in parallel*, but it seems hard to capture denotationally as one must be able to *see* when parts of the computation are causally independent.

Investigating this question, we had two main inspirations: firstly [Ghica, 2005], where Ghica builds a variant of $\circlearrowleft$-**Strat** to capture semantically the (interleaving) execution time for an $\mathsf{IA}_{/\!/}$-like language; and secondly [Laird et al., 2013], where the authors use the relational model weighted over the tropical or arctic semirings to capture resource usage for a PCF-like language extended with non-deterministic choice.

In [Alcolei et al., 2019] we introduce a version of concurrent games parametrized by an algebraic structure we call a *resource bimonoid*, that tracks resource consumption *sequentially* and *in parallel* – our main example is $\mathbb{R}_+$ for time, equipped with $+$ (for sequential resource consumption) and max (for parallel resource consumption). In this model, strategies track resource consumption: each positive move comes equipped with an expression describing the resource usage necessary to play this move, as a function of the resources used to reach its causal dependencies. Building on this, we give (among other results) an interpretation of an affine $\mathsf{IA}_{/\!/}$-like language extended with **wait**, correctly predicting execution time in a maximally parallel evaluation model. This is illustrated in Figure 13.6, that shows the interaction yielding the interpretation of the program above, where each event is annotated by its minimal time of appearance.

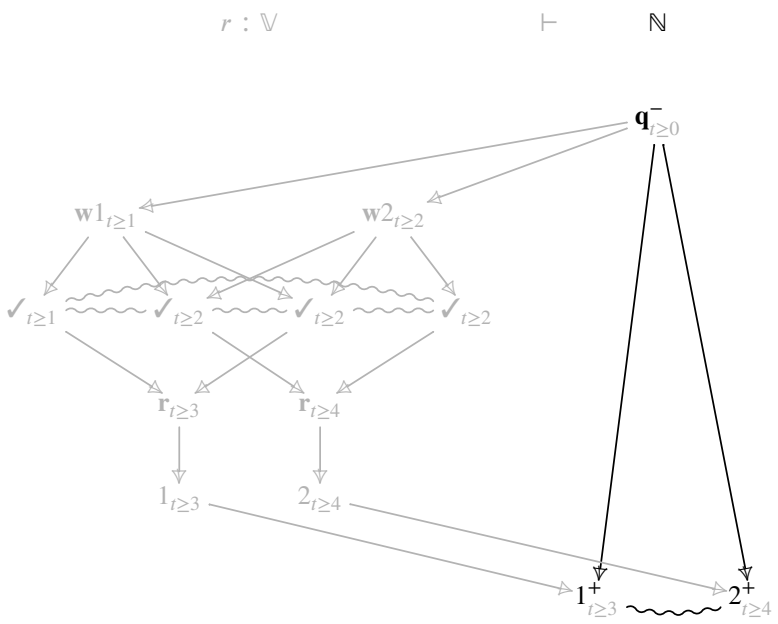


Figure 13.6: Predicting minimal parallel evaluation time

This is neat as it is an application that really exploits the causal information available in concurrent games. It is also the first occurrence of a trend that we followed next: the enriching of concurrent games with quantitative information.

13.2 Quantitative Concurrent Games

This section summarizes work developed in collaboration with Hugo Paquet and Glynn Winskel, as part of Hugo Paquet's PhD thesis[1]; and – regarding the work on semantics of quantum programming – as part of Marc de Visme's PhD thesis (2017-2020).

The original motivation for this line of work was two-fold. The first was an intriguing gap in the literature, originally investigated by Hugo during a research internship at IRIF with Christine Tasson. There were two fully abstract models for fairly close probabilistic languages: probabilistic game semantics [Danos and Harmer, 2000] for probabilistic IA, and the weighted relational model [Laird et al., 2013] for probabilistic PCF. While these models were fairly close in spirit, it was very unclear how to actually relate them formally. Could we have a time-forgetting map from games to weighted relations? From reading Melliès' papers on asynchronous games [Melliès, 2005], it was clear to me that the source of such a time-forgetting map should not be Danos and

[1]Hugo did his PhD in Cambridge with Glynn Winskel, defended in 2019, but I spent most of the year 2016 in Cambridge, and a lot of that work was carried out during that time.

Harmer's model, but rather a concurrent games version, for reasons similar to those that make the relational collapse of Section 10.4 tick.

The second motivation was to give a notion of *probabilistic innocence*, *i.e.* to capture the interactive behaviour of pure parallel programs. It had been known for long that the naive theory of non-deterministic innocence does not work in sequential game semantics [Harmer, 1999]; and it seemed that such a theory could be obtained in probabilistic concurrent games thanks to their explicit handling of the branching information[2].

In both cases, this seemed to be problems where concurrent games were naturally helpful, so we embarked in our journey in probabilistic concurrent games.

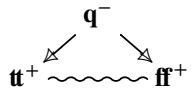
13.2.1 Probabilistic PCF

When we started working on this, there was already a notion of probabilistic concurrent strategy, due to Glynn [Winskel, 2013a], which we first review.

The initial idea is simple: a probabilistic concurrent strategy is a strategy σ with

$$v_\sigma : \mathscr{C}(\sigma) \to [0,1]$$

a *probability valuation* that expresses *the probabilistic weight of the Player choices made so far*. So for instance, a strategy for a fair random boolean would have



with $v(\emptyset) = 1$, $v(\{\mathbf{q}^-\}) = 1$, $v(\{\mathbf{q}^-, \mathbf{tt}^+\}) = v(\{\mathbf{q}^-, \mathbf{ff}^+\}) = \frac{1}{2}$ – playing Opponent moves has no impact on the probability valuation as only the weight of the Player choices is tracked (this is similar to the approach in [Danos and Harmer, 2000]).

Glynn's probabilistic strategies must satisfy additional conditions. The easy ones are

$$\begin{array}{rl} \textit{normalized:} & v(\emptyset) = 1, \\ \textit{receptive:} & \text{if } x^\sigma \subseteq^- y^\sigma, \text{ then } v(x^\sigma) = v(y^\sigma), \end{array}$$

but there is more. If from a configuration $x^\sigma \in \mathscr{C}(\sigma)$, σ can do two events s_1^+ and s_2^+ which are additionally conflicting, then we expect to have the inequality

$$\frac{v(x^\sigma \cup \{s_1^+\})}{v(x^\sigma)} + \frac{v(x^\sigma \cup \{s_2^+\})}{v(x^\sigma)} \leq 1 \ :$$

the first term of the sum is the probability of playing s_1^+ in configuration x^σ, the second is the probability of playing s_2^+ in configuration x^σ; the condition amounts to asking that those form a sub-probability distribution – note the same condition can be written

$$v(x^\sigma) \geq v(x^\sigma \cup \{s_1^+\}) + v(x^\sigma \cup \{s_2^+\}) \, .$$

[2]Other works had already arrived at notions of non-deterministic innocence via adopting strategies with explicit branching structure [Castellan et al., 2014, Tsukada and Ong, 2015].

In sequential probabilistic game semantics [Danos and Harmer, 2000], a similar condition suffices. But the soundness of this relies on the conflict between s_1^+ and s_2^+, so that $x^\sigma \cup \{s_1^+\}$ and $x^\sigma \cup \{s_2^+\}$ delineate two disjoint parts of the probability space. In concurrent games it is possible that $x^\sigma \cup \{s_1^+, s_2^+\}$ is also a configuration, which is accounted for twice in the sum above. Correcting for this yields the inequality

$$v(x^\sigma) \geq v(x^\sigma \cup \{s_1^+\}) + v(x^\sigma \cup \{s_2^+\}) - v(x^\sigma \cup \{s_1^+, s_2^+\}),$$

but a similar phenomenon occurs for three concurrent events, in which case

$$\begin{array}{rcl} v(x^\sigma) & \geq & v(x^\sigma \cup \{s_1^+\}) + v(x^\sigma \cup \{s_2^+\}) + v(x^\sigma \cup \{s_3^+\}) \\ & & -v(x^\sigma \cup \{s_1^+, s_2^+\}) - v(x^\sigma \cup \{s_1^+, s_3^+\}) - v(x^\sigma \cup \{s_2^+, s_3^+\}) \\ & & +v(x^\sigma \cup \{s_1^+, s_2^+, s_3^+\}), \end{array}$$

and the general formula can be written following the *inclusion-exclusion principle* – Glynn called this the *drop condition* as it requires that a quantity, the "drop" obtained as the left hand side term minus the right hand side term, should be positive.

Note that by *receptive*, it suffices to assign probability valuations to +-covered configurations. This makes the compositional structure of probabilistic strategies very easy:

$$v_{\tau \odot \sigma}(x^\tau \odot x^\sigma) \quad = \quad v_\sigma(x^\sigma) \cdot v_\tau(x^\tau) \tag{13.1}$$

for all $x^\tau \odot x^\sigma \in \mathscr{C}^+(\tau \odot \sigma)$; and $v_{\mathbb{c}_A}(\mathbb{c}_x) = 1$ for all $x \in \mathscr{C}(A)$. Probabilistic strategies directly inherit from **CG** the structure of a bicategory (the only difficulty being composition of the drop condition). But it is "only" a bicategory, and we needed much more in order to interpret PCF, so our first task was to merge this with **TCG**.

In doing that, we had the good surprise to find that this needed almost no work at all! In dealing with symmetry, the only condition that must be added to strategies is

$$\textit{invariance:} \quad \text{for all } x^\sigma \cong_\sigma y^\sigma, \text{ we have } v_\sigma(x^\sigma) = v_\sigma(y^\sigma)$$

ensuring that the probability valuation is invariant under symmetry – the drop condition, for instance, remained unchanged[3], forming a relative Seely category that we shall refer to as **ProbNTCG** in the context of this monograph. Then, the same constructions used to model PCF and reported in Chapter 8 work just as well to give a model of probabilistic PCF – this was the starting point of the paper [Castellan et al., 2018b].

Probabilistic innocence. Then in that situation, we looked for a notion of *probabilistic innocence* – note that we are talking about *sequential* probabilistic innocence, as it is already unclear what is parallel *non-deterministic* innocence, see Section 14.2.1. Hence we could build on the sequential non-deterministic innocence of [Castellan et al., 2014]:

Definition 13.2.1. *A strategy σ on board A is* non-deterministic sequential innocent *if:*

$$\begin{array}{rl} \text{negative:} & \textit{for all } s \in \min(\sigma),\ \mathrm{pol}_\sigma(s) = -, \\ \text{forestial:} & \textit{if } s_1, s_2 \leq_\sigma s, \textit{ then } s_1 \leq_\sigma s_2 \textit{ or } s_2 \leq_\sigma s_2, \\ \text{locally conflicting:} & \textit{if } s_1 \smile_\sigma s_2, \textit{ then } [s_1)_\sigma = [s_2)_\sigma, \\ \text{locally sequential:} & \textit{if } s^- \rightarrowtriangle_\sigma s_1^+, s_2^+, \textit{ then } s_1^+ \smile_\sigma s_2^+. \end{array}$$

[3] Note in passing the nightmare avoided by using *thin* concurrent games instead of the *saturated* concurrent games of [Castellan et al., 2014]! See Section 7.1.2 for a reminder on the thin / saturated distinction.

This takes as a definition the forestial shape that was derived in this monograph from combined sequentiality and parallel innocence (see Lemma 11.3.2); with the difference that the resulting forest might not be O-branching: Player is now also able to branch. But that Player branching cannot be parallel branching, it must come with a conflict. Thus any configuration is an O-branching forest, where each branch is an independent exploration of a branch of the term, following the intuitions of Section 11.3.1.

Now for a *probabilistic strategy*, how should the probabilistic weight behave with respect to this forestial structure? What we proposed in [Castellan et al., 2018b] is that on top of non-deterministic sequential innocence, we should have

Definition 13.2.2. *Consider $\sigma : A$ probabilistic, non-deterministic seq. innocent. It is **probabilistic innocent** iff for all $x, y \in \mathscr{C}(\sigma)$ such that $x \cup y \in \mathscr{C}(\sigma)$,*

$$v_\sigma(x \cup y) = \frac{v_\sigma(x) \cdot v_\sigma(y)}{v_\sigma(x \cap y)}\,.$$

This asserts that two *causally independent* branches of a configuration must also be *probabilistically independent*: the extensions $x \cap y \subseteq x$ and $x \cap y \subseteq y$ must explore distinct branches of the syntax tree, and hence trigger independent probabilistic choices.

Probabilistic innocent strategies form a relative Seely category that supports the interpretation of probabilistic PCF, and enjoy a finite definability result yielding intensional full abstraction for probabilistic PCF. But probabilistic PCF has a simpler, actually fully abstract model: probabilistic coherence spaces [Ehrhard et al., 2018] and hence their backbone, the weighted relational model [Laird et al., 2013].

Our paper had a second result, an interpretation-preserving collapse to the weighted relational model, but we shall come back to that in Section 13.3. In any case, there was clearly a parallel between concurrent games and relational models: just as relational models are made quantitative by attaching to each point of the "web" a coefficient, and concurrent games are made quantitative by attaching to each configuration a coefficient. But the relational supports more elaborate weights than just scalars: in particular, points of the web can be decorated by finite dimensional Hilbert spaces and completely positive maps, to obtain a computationally adequate model of the quantum λ-calculus [Pagani et al., 2014]. This suggested to do the same in concurrent games.

13.2.2 Quantum strategies

This is joint work with Marc de Visme for his PhD (2017-2020), and Glynn Winskel.

In Marc's PhD thesis, we set as main objective to build a computationally adequate concurrent games model for the quantum λ-calculus; this is a call-by-value language, linear but with an explicit exponential modality, structured via the types

$$A, B ::= \mathsf{qubit} \mid \mathbf{1} \mid A \otimes B \mid A \oplus B \mid \mathsf{list(A)} \mid A \multimap B \mid !(A \multimap B)\,,$$

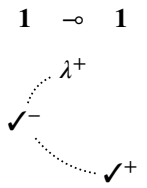
with *e.g.* $\mathbb{B} = \mathbf{1} \oplus \mathbf{1}$ defined as syntactic sugar; linearity is of course crucial because qubits *cannot* be duplicated. Terms include the expected term formers for these types,

along with new primitives, used to manipulate quantum data:

$$\textbf{new} : \mathbb{B} \multimap \mathsf{qubit}\,, \qquad \textbf{meas} : \mathsf{qubit} \multimap \mathbb{B}\,, \qquad U : \mathsf{qubit}^{\otimes n} \multimap \mathsf{qubit}^{\otimes n}\,,$$

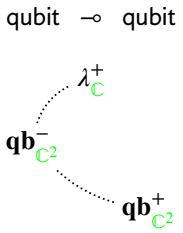
where **new** initializes a new qubit based on a boolean, **meas** measures a qubit (with a probabilistic result), and U imports any n-ary unitary inside the syntax.

Building a concurrent games model for this language posed several challenges. First of course, the language is call-by-value – while there were no earlier concurrent games models of call-by-value languages, it was not too difficult to infer what the corresponding structure should be, looking at traditional games models for call-by-value languages [Abramsky and McCusker, 1997, Honda and Yoshida, 1999] and from the game semantics of tensorial logic in *e.g.* [Melliès and Tabareau, 2007]. In particular, in contrast with **NTCG**, types (ignoring the quantum aspect) should be interpreted as *positive* games, where Player always starts. For instance, the type $\mathbf{1} \multimap \mathbf{1}$ yields the board



where computation starts by Player providing a value – in that written λ^+, as it carries the information that the program yields an abstraction. This enables Opponent to call the function by providing an argument which can only be the unique value of unit type, which in turn enables Player to terminate with again, the unique value of unit type.

The main challenge was dealing with quantum information. For this, our idea follows the paradigm to separate *quantum data* and *classical control*, putting together ideas from probabilistic concurrent games above, and quantum relational semantics [Pagani et al., 2014]. For *classical control*, qubits are interpreted just like the unit type, so that the plain event structure for $\mathsf{qubit} \multimap \mathsf{qubit}$ is exactly like $\mathbf{1} \multimap \mathbf{1}$ above. The difference comes with *quantum data*, for which each event $a \in A$ of the game comes labelled with a (finite dimensional) Hilbert space $\mathcal{Q}_A(a)$. Events that do not carry any quantum state are annotated with the 1-dimensional Hilbert space, the unit for the tensor of Hilbert spaces, *i.e.* the complex numbers $\mathbb{C}$. Events that do have a quantum content carry a non-trivial Hilbert space; for instance $\mathsf{qubit} \multimap \mathsf{qubit}$ becomes the game



with Hilbert space annotations in green. Intuitively, playing a move annotated with a Hilbert space H, has a side effect of opening H in the quantum register. Accordingly,

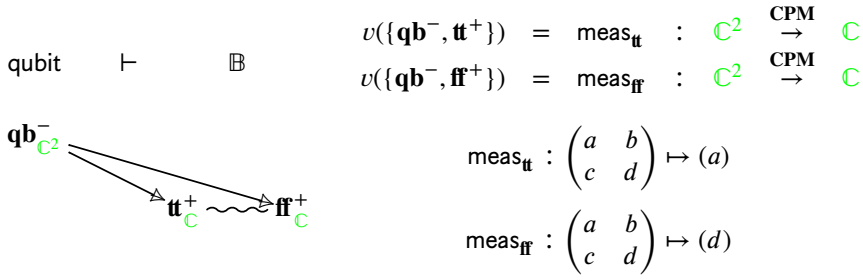


Figure 13.7: Quantum strategy for measurement

any configuration $x \in \mathscr{C}(A)$ yields a finite dimensional Hilbert space

$$\mathcal{Q}_A(x) = \bigotimes_{a \in x} \mathcal{Q}_A(a)\,,$$

the tensor of the Hilbert spaces opened in x. Now the idea is that a strategy $\sigma : A$ is accompanied by a valuation as for probabilistic strategies, except it is *typed*: to each $x^\sigma \in \mathscr{C}(\sigma)$ we associate a *density operator* on the Hilbert space of moves played so far:

$$v_\sigma(x^\sigma) \in \mathbf{DOp}(\mathcal{Q}_A(\partial_\sigma\, x^\sigma))\,;$$

recall that density operators are the standard representation of *mixed quantum states*, *i.e.* supporting pure states but also convex (probabilistic) sums and measurements. Now, the definition of quantum strategies is analogous to that of probabilistic strategies, except that the valuations are *typed*: the *drop condition* can be defined, though it is with respect to the so-called *Löwner order* between density operators and involves some tracing out to bring all density operators to the same Hilbert space.

At the core of our approach, we use that for a strategy between games $\sigma : A \vdash B$, a quantum valuation on $A \vdash B$, *i.e.* a density operator on a tensor Hilbert space

$$v_\sigma(x^\sigma) \in \mathbf{DOp}(\mathcal{Q}_A(x^\sigma_A) \otimes \mathcal{Q}_B(x^\sigma_B))$$

may be equivalently presented as a *completely positive map*, *i.e.* a morphism

$$v^{\rightarrow}_\sigma(x^\sigma) \in \mathbf{CPM}(\mathcal{Q}_A(x^\sigma_A), \mathcal{Q}_B(x^\sigma_B))\,,$$

in the compact closed category **CPM** that we shall not define here, but which should be thought of as taking mixed states on $\mathcal{Q}_A(x^\sigma_A)$ (so, certain operators on $\mathcal{Q}_A(x^\sigma_A)$ to mixed states on $\mathcal{Q}_B(x^\sigma_B)$. This lets us define the quantum valuation of the composition $\tau \odot \sigma$ simply by composition of the quantum valuations of σ and τ respectively, via

$$v^{\rightarrow}_{\tau\odot\sigma}(x^\tau \odot x^\sigma) = v^{\rightarrow}_\tau(x^\tau) \circ v^{\rightarrow}_\sigma(x^\sigma)$$

reminiscent of (13.1) for probabilistic strategies.

As an example, we show in Figure 13.7 the quantum strategies for measurement. Its control flow seems to select a boolean non-deterministically, but each branch is

weighted by the **CPM** map taking a density matrix on $\mathbb{C}^2$ and maps it to (a 1 by 1 matrix with) the probability of obtaining the corresponding value via a measurement.

With Marc and Glynn, we published a first paper [Clairambault et al., 2019] presenting the model; its main results are computational adequacy, but also an important consequence of the quantum *drop condition*: our quantum valuations can be presented as *superoperators*, those **CPM** maps that are trace non-increasing, they correspond to the physically realizable transformations on mixed quantum states. More precisely, consider $\boldsymbol{\sigma} : A$, $x^{\boldsymbol{\sigma}} \in \mathscr{C}(\boldsymbol{\sigma})$ and $\partial_{\boldsymbol{\sigma}}\, x^{\boldsymbol{\sigma}} = x_A$, and split x_A into its positive and negative events $x_A = x_A^- \uplus x_A^+$. Then we may present equivalently $v_{\boldsymbol{\sigma}}(x^{\boldsymbol{\sigma}})$ as

$$v_{\boldsymbol{\sigma}}^{-+}(x^{\boldsymbol{\sigma}}) \in \mathbf{CPM}(\mathscr{Q}_A(x_A^-), \mathscr{Q}_A(x_A^+))\,,$$

which we show is always a superoperator. This was an improvement with respect to the earlier denotational model of the quantum λ-calculus [Pagani et al., 2014], for which there was no such boundedness property – their model construction required valuations to include an infinitary completion of **CPM**, just as the weighted relational model requires an infinitary completion of scalars to work around convergence issues.

In a second paper with Marc [Clairambault and de Visme, 2020], we were able to exploit this boundedness property to adapt in our model the proof method of Ehrhard, Pagani and Tasson for their full abstraction result of probabilistic coherence spaces for Probabilistic PCF [Ehrhard et al., 2018]. This shows that our model is fully abstract for the quantum λ-calculus, modulo a quotient of strategies that amounts to a quantitative relational collapse to the model of [Pagani et al., 2014], which entails that their model was *already* fully abstract. We cover the ideas around this quantitative collapse next.

13.3 Quantitative Relational Collapses

13.3.1 Counting witnesses and quantitative collapse

When working on [Castellan et al., 2018b], one of our goals was to understand the link between probabilistic alternating game semantics [Danos and Harmer, 2000] and probabilistic coherence spaces [Danos and Ehrhard, 2011] via the weighted relational model [Laird et al., 2013]; for that we meant to define an interpretation-preserving functor

$$(\!-\!) : \mathbf{ProbNTCG\text{-}Vis} \to \overline{\mathbb{R}}_+\text{-}\mathbf{Rel}$$

refining quantitatively the one of Corollary 10.4.15 – recall that here, $\overline{\mathbb{R}}_+$-**Rel** is the *weighted relational model* [Laird et al., 2013], here weighted by the semiring $\overline{\mathbb{R}}_+ = \mathbb{R}_+ \uplus \{+\infty\}$ of the completed positive reals. Its objects are sets, and morphisms from A to B are matrices $(\alpha_{a,b})_{a\in A, b\in B}$ where $\alpha_{a,b} \in \overline{\mathbb{R}}_+$ for all $a \in A$ and $b \in B$.

Composition of $\alpha \in \overline{\mathbb{R}}_+\text{-}\mathbf{Rel}(A, B)$ and $\beta \in \overline{\mathbb{R}}_+\text{-}\mathbf{Rel}(B, C)$ is matrix multiplication:

$$(\beta \circ \alpha)_{a,c} = \sum_{b\in B} \alpha_{a,b} \cdot \beta_{b,c}$$

which might not converge if $\boldsymbol{B}$ is infinite – hence the completion to $\overline{\mathbb{R}}_+$.

At the time we were investigating this, it felt like it should be a minor variation of the usual relational collapse of Section 10.4. Recall that for $\boldsymbol{\sigma} : A$, Definition 10.4.3 has

$$\mathbb{(}\boldsymbol{\sigma}\mathbb{)} = \{\mathsf{x}_A \in \mathbb{(}A\mathbb{)} \mid \exists x^{\boldsymbol{\sigma}} \in \mathscr{C}^+(\boldsymbol{\sigma}),\, \partial_{\boldsymbol{\sigma}}\, x^{\boldsymbol{\sigma}} \in \mathsf{x}_A\}\,,$$

recording all $\mathsf{x}_A \in \mathbb{(}A\mathbb{)}$ which is reachable in $\boldsymbol{\sigma}$, which is witnessed by $x^{\boldsymbol{\sigma}} \in \mathscr{C}^+(\boldsymbol{\sigma})$ such that $\partial_{\boldsymbol{\sigma}}\, x^{\boldsymbol{\sigma}} \in \mathsf{x}_A$. Intuitively, instead of only recording the *mere existence* of such a witness, we must *sum the weight* of all possible witnesses. Of course that is too naive: as a strategy needs to be receptive to all Opponent reindexings, and as there are usually countably many of those, each "witness" in the sense above usually has countably many distinct but symmetric peers, yielding almost always infinitely many witnesses.

In [Castellan et al., 2018b], we proposed to solve this issue by summing not over all concrete witnesses as above, but over all *symmetry classes* of witnesses:

$$\mathbb{(}\boldsymbol{\sigma}\mathbb{)}_{\mathsf{x}_A} = \sum_{\mathsf{x}^{\boldsymbol{\sigma}} \in \mathsf{wit}^{\cong}_{\boldsymbol{\sigma}}(\mathsf{x}_A)} v_{\boldsymbol{\sigma}}(\mathsf{x}^{\boldsymbol{\sigma}})$$

for $\mathsf{wit}^{\cong}_{\boldsymbol{\sigma}}(\mathsf{x}_A) = \{\mathsf{x}^{\boldsymbol{\sigma}} \in \mathscr{C}^+_{\cong}(\boldsymbol{\sigma}) \mid \partial_{\boldsymbol{\sigma}}\, \mathsf{x}^{\boldsymbol{\sigma}}\}$ using that by *invariance*, the valuation $v_{\boldsymbol{\sigma}}$ lifts to symmetry classes. This yields a finite coefficient that operates as it should on simple examples. For composition, functoriality amount to proving the equality

$$\mathbb{(}\boldsymbol{\tau} \odot \boldsymbol{\sigma}\mathbb{)}_{\mathsf{x}_A \vdash \mathsf{x}_C} \quad = \quad \sum_{\mathsf{x}_B \in \mathbb{(}B\mathbb{)}} \mathbb{(}\boldsymbol{\sigma}\mathbb{)}_{\mathsf{x}_A \vdash \mathsf{x}_B} \cdot \mathbb{(}\boldsymbol{\tau}\mathbb{)}_{\mathsf{x}_B \vdash \mathsf{x}_C}$$

which seems like it should be a natural consequence of a bijection

$$\mathsf{wit}^{\cong}_{\boldsymbol{\tau}\odot\boldsymbol{\sigma}}(\mathsf{x}_A \vdash \mathsf{x}_C) \quad \simeq \quad \sum_{\mathsf{x}_B \in \mathbb{(}B\mathbb{)}} \mathsf{wit}^{\cong}_{\boldsymbol{\sigma}}(\mathsf{x}_A \vdash \mathsf{x}_B) \times \mathsf{wit}^{\cong}_{\boldsymbol{\tau}}(\mathsf{x}_B \vdash \mathsf{x}_C)\,. \tag{13.2}$$

This seems simple enough: from $\mathsf{x}^{\boldsymbol{\sigma}} \in \mathsf{wit}^{\cong}_{\boldsymbol{\sigma}}(\mathsf{x}_A \vdash \mathsf{x}_B)$ and $\mathsf{x}^{\boldsymbol{\tau}} \in \mathsf{wit}^{\cong}_{\boldsymbol{\tau}}(\mathsf{x}_B \vdash \mathsf{x}_C)$, we can pick representatives $x^{\boldsymbol{\sigma}} \in \mathsf{x}^{\boldsymbol{\sigma}}$, $x^{\boldsymbol{\tau}} \in \mathsf{x}^{\boldsymbol{\tau}}$ and a symmetry $\theta_B : x^{\boldsymbol{\sigma}}_B \cong_B x^{\boldsymbol{\tau}}_B$, and obtain an element of $\mathsf{wit}^{\cong}_{\boldsymbol{\tau}\odot\boldsymbol{\sigma}}(\mathsf{x}_A \vdash \mathsf{x}_C)$ as (the symmetry class of) $x^{\boldsymbol{\tau}} \odot_\theta x^{\boldsymbol{\sigma}}$ defined following Definition 7.4.6. It can be checked that any $\mathsf{x} \in \mathsf{wit}^{\cong}_{\boldsymbol{\tau}\odot\boldsymbol{\sigma}}(\mathsf{x}_A \vdash \mathsf{x}_C)$ can be obtained in this ways, and that $\mathsf{x}^{\boldsymbol{\sigma}} \in \mathsf{wit}^{\cong}_{\boldsymbol{\sigma}}(\mathsf{x}_A \vdash \mathsf{x}_B)$ and $\mathsf{x}^{\boldsymbol{\tau}} \in \mathsf{wit}^{\cong}_{\boldsymbol{\tau}}(\mathsf{x}_B \vdash \mathsf{x}_C)$ can be *uniquely* recovered from x. So this bijection seemed to work just fine, as expected!

Counting witnesses. This is how the collapse was written in [Castellan et al., 2018b], and it was the state of my understanding for about two years. But later on, when writing the details of the quantum version of this collapse with Marc, we realized that something was off – the more typed version of the collapse made it more apparent.

The issue is that in the reasoning above, we have defined

$$\mathsf{wit}^{\cong}_{\boldsymbol{\sigma}}(\mathsf{x}_A \vdash \mathsf{x}_B) \times \mathsf{wit}^{\cong}_{\boldsymbol{\tau}}(\mathsf{x}_B \vdash \mathsf{x}_C) \quad \to \quad \mathsf{wit}^{\cong}_{\boldsymbol{\tau}\odot\boldsymbol{\sigma}}(\mathsf{x}_A \vdash \mathsf{x}_C)$$

a function by working on concrete representatives, and we had not checked that the result does not depend on the representative. And of course it does, and it also depends

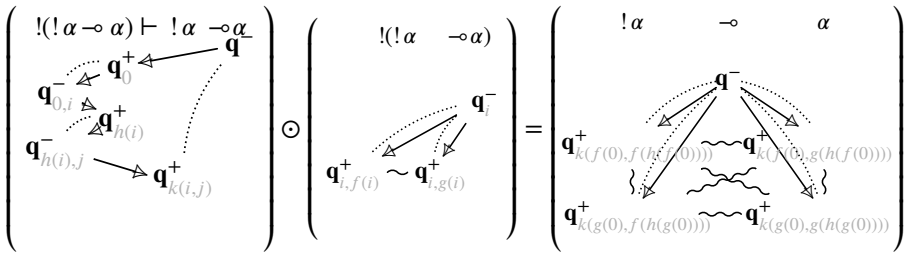


Figure 13.8: Composition of σ and τ

on the choice of the mediating symmetry θ_B – which in hindsight should have been obvious with some familiarity with the resource calculus, because θ_B somehow corresponds to the non-determinism of reduction in the resource calculus (see Section 13.4). As a matter of fact, (13.2) fails; let us see a counter-example:

Example 13.3.1. *Take the board α with one event $\mathbf{q}^-$, $\kappa_\alpha(\emptyset) = 1$, $\kappa_\alpha(\{\mathbf{q}^-\}) = 0$, and*

$$\sigma : !(!\alpha \multimap \alpha) \vdash !\alpha \multimap \alpha\,, \qquad \tau : !(!\alpha \multimap \alpha)$$

two strategies as in Figure 13.8. Their assignment of copy indices uses functions

$$f : \mathbb{N} \to \mathbb{N}, \qquad g : \mathbb{N} \to \mathbb{N}, \qquad h : \mathbb{N} \to \mathbb{N}, \qquad k : \mathbb{N}^2 \to \mathbb{N},$$

whose precise identity is irrelevant. Their composition unfolds as in Figure 13.8, yielding four pairwise conflicting, non-symmetric positive moves.

Enriching non-deterministic PCF *with a type α with no constant, the substitution*

$$(\lambda x^\alpha.\, f\,(f\,x))\ [(\lambda y^\alpha.\, y \mathbin{\textcircled{\scriptsize V}} y)/f]$$

gives a perfect syntactic counterpart to the composition in Figure 13.8. As there are two calls to the non-deterministic choice, this reduces to $\lambda x^\alpha.\, x$, but in four different ways. In the copy indices of each positive move of $\tau \odot \sigma$, one can read back how the two non-deterministic choices were resolved: the upper row corresponds to the first call yielding $\mathbf{q}^+_{i,f(i)}$, the leftmost column to the second call yielding $\mathbf{q}^+_{i,f(i)}$, and so on.

Reading back witnesses from the components and their composition, we have

$$\sharp\mathsf{wit}^{\cong}_{\tau\odot\sigma}\begin{pmatrix}\mathbf{q}^-\\ \vdots\\ \mathbf{q}^+\end{pmatrix} = 4\,, \qquad \sharp\mathsf{wit}^{\cong}_{\sigma}\begin{pmatrix}\mathbf{q}^- & \mathbf{q}^- & \mathbf{q}^-\\ \vdots & \vdots & \vdots\\ \mathbf{q}^+ & \mathbf{q}^+ & \mathbf{q}^+\end{pmatrix} = 1\,, \qquad \sharp\mathsf{wit}^{\cong}_{\tau}\begin{pmatrix}\mathbf{q}^- & \mathbf{q}^-\\ \vdots & \vdots\\ \mathbf{q}^+ & \mathbf{q}^+\end{pmatrix} = 3\,;$$

as σ and τ cannot synchronize on any other symmetry class, this contradicts (13.2)*.*

After staring at this example for a while, it seemed that the mistake was in the enumeration of the witnesses for $\boldsymbol{\tau}$. Indeed, up to symmetry, there are exactly *three* configurations of τ corresponding to two calls: *(1)* both choices may be resolved with $\mathbf{q}_{i,f(i)}$ (call this "left-left"); *(2)* both choices may be resolved with $\mathbf{q}_{i,g(i)}$ (call this "right-right");

and *(3)* we may have one of each. But arguably, *(3)* really identifies two cases: *(3a)* "left-right" and *(3b)* "right-left". Those configurations are symmetric, but it turns out that **TCG** is equipped with exactly the right tool to keep them distinct. Imagine fixed, for every $\mathsf{x}_A \in (\!|A|\!)$, a concrete representative $\underline{\mathsf{x}}_A \in \mathsf{x}_A$. Then, for $\sigma : A$ we set

$$\mathsf{wit}^+_\sigma(\mathsf{x}_A) = \{x^\sigma \in \mathscr{C}^+(\sigma) \mid \partial_\sigma\, x^\sigma \cong^+_A \underline{\mathsf{x}}_A\}$$

the set of **positive witnesses**[4] for x_A, *i.e.* those *concrete* configurations of σ for which there exists a *positive* symmetry to the chosen representative $\underline{\mathsf{x}}_A$. The *positivity* requirement morally fixes the identity of each Opponent move, factoring out Opponent reindexing but still letting us see the Player behaviour. We now have

$$\mathsf{wit}^+_\tau\left(\begin{array}{cc}\mathbf{q}^- & \mathbf{q}^-\\ \vdots & \vdots\\ \mathbf{q}^+ & \mathbf{q}^+\end{array}\right) = \left\{\left(\begin{array}{cc}\mathbf{q}^-_0 & \mathbf{q}^-_1\\ \vdots & \vdots\\ \mathbf{q}^+_{0,f(0)} & \mathbf{q}^+_{1,f(1)}\end{array}\right), \left(\begin{array}{cc}\mathbf{q}^-_0 & \mathbf{q}^-_1\\ \vdots & \vdots\\ \mathbf{q}^+_{0,g(0)} & \mathbf{q}^+_{1,g(1)}\end{array}\right), \left(\begin{array}{cc}\mathbf{q}^-_0 & \mathbf{q}^-_1\\ \vdots & \vdots\\ \mathbf{q}^+_{0,f(0)} & \mathbf{q}^+_{1,g(1)}\end{array}\right), \left(\begin{array}{cc}\mathbf{q}^-_0 & \mathbf{q}^-_1\\ \vdots & \vdots\\ \mathbf{q}^+_{0,g(0)} & \mathbf{q}^+_{1,f(1)}\end{array}\right)\right\}$$

a set of concrete configurations of cardinal 4, solving the mismatch.

In [Clairambault and Paquet, 2021] we detail this, which involves delving into an intricate analysis of symmetries in concurrent games. Thanks to this we are able to correct the interpretation-preserving functor $(\!|-|\!) : \mathbf{ProbNTCG\text{-}Vis} \to \overline{\mathbb{R}}_+\text{-}\mathbf{Rel}$ of [Castellan et al., 2018b]: the correct definition turns out to be given by

$$(\!|\sigma|\!)_{\mathsf{x}_A} = \sum_{x^\sigma \in \mathsf{wit}^+_\sigma(\mathsf{x}_A)} v_\sigma(x^\sigma)\,,$$

summing over *positive witnesses* rather than symmetry classes[5]. Beyond probabilistic weights, we prove this for games and relations weighted by arbitrary semirings of resources, modulo mild hypotheses. An interesting case is the interpretation of PCF in $\overline{\mathbb{N}}$-**Rel**, the relational model weighted by natural numbers. If $\vdash M : A$ is a term, to each $a \in [\![A]\!]$, $[\![M]\!]_a$ is a natural number, which might be arbitrarily high. So the relational model *counts* something, but what? It turns out from our study that the weighted relational model counts *positive witnesses* in the sense of concurrent game semantics.

Finally, in [Clairambault and de Visme, 2020], we also apply these ideas for the quantitative collapse of quantum strategies onto the quantum relational model.

13.3.2 Generalized species of structure

Interpreting programs in **TCG** allows us – for each point of the web in the sense of relational semantics – to replace a *natural number*, as given by the weighted relational model, with a *set of witnesses*. This is an example of what is increasingly being called *proof-relevant denotational semantics*; which beyond collecting behaviours, tracks, for each possible behaviour, a set of *witnesses*. The main example of such a model is the cartesian closed bicategory of *generalized species of structure* [Fiore et al., 2008].

[4] The cardinal of $\mathsf{wit}^+_\sigma(\mathsf{x}_A)$ does not depend on the representative, provided it satisfies a condition called *canonicity*, not covered here, see [Clairambault and Paquet, 2021].

[5] One can sum over symmetry classes, with correcting coefficients: $(\!|\sigma|\!)_{\mathsf{x}_A} = \sum_{x^\sigma \in \mathsf{wit}^{\cong}_\sigma(\mathsf{x}_A)} \frac{\sharp\mathcal{S}_-(\mathsf{x}_A)}{\sharp\mathcal{S}(\mathsf{x}^\sigma)} \cdot v_\sigma(\mathsf{x}^\sigma)$.

At the time (2021-2022), there was intense activity, in the quantitative semantics community, around species of structure. Roughly speaking, species of structure are a categorification of the relational model. The objects of **Rel**, *i.e.* sets, are replaced with categories. The morphisms of **Rel**, *i.e.* functions $\alpha : A \times B \to \{0,1\}$ are replaced with

$$\alpha : A^{\mathrm{op}} \times B \to \mathbf{Set}\,,$$

i.e. profunctors from A to B. The *bang* of **Rel**, which to any set associates its set of finite multisets, is replaced with the *free symmetric monoidal category*. Altogether, to a program $\vdash M : A$ and some $a \in A$, the interpretation of M associates a set $[\![M]\!]_a$ – notably, Federico Olimpieri proved [Olimpieri, 2021] that this set could be presented as the set of *derivations* of $M : a$ in a non-idempotent intersection type system.

As two "proof-relevant" denotational models, it was appealing to try to understand the links between concurrent games and generalized species of structure. As generalized species of structure replace the coefficient computed by the weighted relational model with a set, it is tempting to conjecture that the two agree – that the coefficient given by the weighted relational model is the cardinal of the witnesses given by generalized species. But it is easy to see that this is not true: for instance, for any set A we have

$$(\mathrm{id}_A)_{a,a'} = \begin{cases} 1 & \text{if } a = a', \\ 0 & \text{otherwise} \end{cases}$$

for the identity in the weighted relational model, compared with $(\mathrm{id}_A)_{a,a'} = A[a,a']$ given by the hom-functor, the identity in the bicategory of profunctors. We can plainly see that species of structures import morphisms of A as witnesses, whereas the weighted relational model does not. From the perspective of concurrent games these morphisms are nothing but the *symmetries* between configurations, so it seems that witnesses in the sense of generalized species of structure should correspond to witnesses in the sense of concurrent games *along with* explicit symmetries, *i.e.* for a strategy $\sigma : A \vdash B$,

$$(\!|\sigma|\!)(x_A, x_B) = \{(\theta_A^-, x^\sigma, \theta_B^+) \mid \theta_A^- : x_A \cong_A^- x_A^\sigma,\ x^\sigma \in \mathscr{C}^+(\sigma),\ \theta_B^+ : x_B^\sigma \cong_B^+ x_B\}\,.$$

Though defined on configurations, this can turned be into a profunctor

$$(\!|\sigma|\!) : \mathscr{S}(A)^{\mathrm{op}} \times \mathscr{S}(B) \to \mathbf{Set}$$

from the groupoid $\mathscr{S}(A)$ of symmetries of A, to that on B – the functorial action is an elaboration of Lemma 7.2.7. With significant effort with Hugo Paquet and Federico Olimpieri, we proved in [Clairambault et al., 2023a] that this yields a pseudofunctor from a cartesian closed bicategory of thin concurrent games (with visible strategies, so as to cope with deadlocks) into that of generalized species of structure. It turns out that up to isomorphism, generalized species of structure can be seen as computing positive witnesses in the sense of concurrent games, along with explicit positive symmetries! This is striking, as species of structure were invented between any of those. This also shows that the cardinality mismatch between the weighted relational model and generalized species of structure exactly amounts to the number of positive symmetries.

This is a new step in the relational collapses of games model, refining earlier work in that it is proof-relevant, and operates at the bicategorical level. It also emphasizes that the combinatorial core of proof-relevant models is their handling of symmetries, and there is an incredible wealth of mathematical structures to be worked out here, in which thin concurrent games play a central role. This is also strongly connected to the coefficients appearing in the Taylor expansion of λ-terms, see the next section.

13.4 Game Semantics as Taylor Expansion

This line of work is in collaboration with Lison Blondeau-Patissier, as part of her PhD thesis (2021-2024), and with Lionel Vaux Auclair.

13.4.1 Pointer concurrent games and positional injectivity

As we have seen, thin concurrent games provide an expressive representation of the interactive behaviour of programs, but at the cost of a significant technical complexity – a lot of which comes from copy indexing and uniformity. Could we conceivably simplify this by adopting an "Hyland-Ong" style representation, with pointers? It turns out that we can to an extent, losing some features such as non-deterministic branching, but gaining some such as a compelling connection with the Taylor expansion of λ-terms.

The idea is the following: instead of a concurrent strategy $\boldsymbol{\sigma}$, we retain only its set of symmetry classes of (+-covered) configurations, which must hence be defined directly, without appealing to all the technology of Part II. To do that we start with:

Definition 13.4.1. *Consider A an arena in the sense of Definition 3.1.3.*

*An **exploration**[6] $x \in \mathscr{E}(A)$ of arena A is $x = \langle |x|, \leq_x, \partial_x \rangle$ such that $\langle |x|, \leq_x \rangle$ is a finite forest, and the **display map** $\partial_x : |x| \to |A|$ is a function satisfying the conditions:*

minimality-respecting:	*for any $a \in |x|$, a is $\leq_x$-minimal iff $\partial_x(a)$ is $\leq_A$-minimal,*
causality-preserving:	*for all $a_1, a_2 \in |x|$, if $a_1 \rightarrowtriangle_x a_2$ then $\partial_x(a_1) \rightarrowtriangle_A \partial_x(a_2)$.*

If A is a mixed board, then explorations of $\underline{A}$ are analogous to configurations of A: they serve as concrete representatives for symmetry classes. A **symmetry** between explorations $x, y \in \mathscr{E}(A)$ is a bijection $\varphi : |x| \simeq |y|$ that preserves and reflects all structure. This lets us define **positions** as symmetry classes of explorations – if A is a mixed board without conflict, those are in one-to-one correspondence with $\mathscr{C}_{\cong}(A)$.

But one should not immediately quotient explorations to positions, because doing so blurs out the identity of individual moves, preventing us from enriching it with the *dynamic* causality from the program. Instead those are added on *explorations*:

Definition 13.4.2. *An **augmentation** on arena A is a tuple $\mathsf{q} = \langle |\mathsf{q}|, \leq_{\mathsf{q}}, \leq_{(\!|\mathsf{q}|\!)}, \partial_{\mathsf{q}} \rangle$, where*

[6] In the published papers, we call those *configurations*. Here, I use *exploration* instead to avoid the collision with the configurations of an event structure.

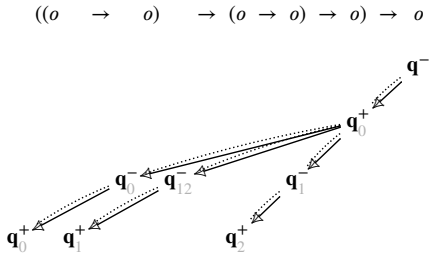


Figure 13.9: A configuration

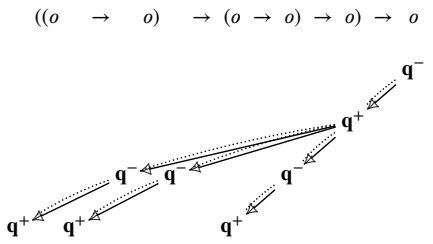


Figure 13.10: An isogmentation

$(\!|\mathsf{q}|\!) = \langle |\mathsf{q}|, \leq_{(\!|\mathsf{q}|\!)}, \partial_{\mathsf{q}} \rangle \in \mathscr{C}(A)$*, and* $\langle |\mathsf{q}|, \leq_{\mathsf{q}} \rangle$ *is a forest*[7] *satisfying:*

rule-abiding:	*for all* $a_1, a_2 \in |\mathsf{q}|$*, if* $a_1 \leq_{(\!|\mathsf{q}|\!)} a_2$*, then* $a_1 \leq_{\mathsf{q}} a_2$*,*
courteous:	*for all* $a_1 \rightarrowtriangle_{\mathsf{q}} a_2$*, if* $\mathrm{pol}(a_1) = +$ *or* $\mathrm{pol}(a_2) = -$*, then* $a_1 \rightarrowtriangle_{(\!|\mathsf{q}|\!)} a_2$*,*
deterministic:	*for all* $a^- \rightarrowtriangle_{\mathsf{q}} a_1^+$ *and* $a^- \rightarrowtriangle_{\mathsf{q}} a_2^+$*, then* $a_1 = a_2$*,*
+-covered:	*for all* $a \in |\mathsf{q}|$ *maximal in* q*, we have* $\mathrm{pol}(a) = +$*,*
negative:	*for all* $a \in \min(\mathsf{q})$*, we have* $\mathrm{pol}(a) = -$*,*

we then write $\mathsf{q} \in \mathscr{A}(A)$*, and call* $(\!|\mathsf{q}|\!) \in \mathscr{C}(A)$ *the* ***desequentialization*** *of* q*.*

An augmentation equips an exploration of the game with the causal dependency due to the program. In particular, if $\boldsymbol{\sigma} : A$ is a causal strategy on a mixed board A, then any $x^{\boldsymbol{\sigma}} \in \mathscr{C}(\boldsymbol{\sigma})$ gives rise to an augmentation $\mathsf{q} \in \mathscr{A}(A)$ on $\underline{A}$ by setting $|\mathsf{q}| = x^{\boldsymbol{\sigma}}$, $s_1 \leq_{\mathsf{q}} s_2$ iff $s_1 \leq_{\boldsymbol{\sigma}} s_2$, $s_1 \leq_{(\!|\mathsf{q}|\!)} s_2$ iff $\partial_{\boldsymbol{\sigma}}\, s_1 \leq_A \partial_{\boldsymbol{\sigma}}\, s_2$, and for all $s \in |\mathsf{q}|$, $\partial_{\mathsf{q}}\, s = \mathsf{lbl}_A(\partial_{\boldsymbol{\sigma}}\, s)$.

In thin concurrent games, all events have a distinct identity and yield a distinct image in the game with distinct copy indices, as illustrated in Figure 13.9; this rigidity is then tamed by 2-cells mediating between strategies. In contrast, in pointer concurrent games, individual events have no identity; just like elements of a multiset have no individual identity. This is illustrated in Figure 13.10, but it is not properly captured yet by Definition 13.4.2 – it remains to quotient out the identity of events. For that, a **symmetry** between augmentations q and p is a bijection $\varphi : |\mathsf{q}| \simeq |\mathsf{p}|$ preserving and reflecting all structure. Finally, an **isogmentation** is a symmetry class of augmentations; and then *strategies* should morally be certain aggregations of isogmentations.

Our original motivation in developing these definitions was not really to define a fully-fledged variant of concurrent games, but rather to obtain the tools we needed to prove an *injectivity* theorem [Blondeau-Patissier and Clairambault, 2021] along the following lines. Consider $\sigma, \tau : A$ two total finite innocent strategies in the Hyland-Ong sense. Each play $s \in \sigma$ reaches a *position* obtained by simply forgetting the chronological order; so that for each strategy we can retain its set of positions only. Now the theorem is that if σ and τ reach the same positions, then they are equal. Now proving

[7] For now, we have only explored pointer concurrent games in a sequential innocent setting – of course, there is no reason why the methodology would not extend further.

that involves a *pumping* technique similar to that used in traditional injectivity results in the linear logic literature [de Carvalho and de Falco, 2012], which needed understanding the effect of Opponent duplications on reached positions – and in turn, this invited a representation of plays factoring out Opponent's scheduling, *i.e.* isogmentations[8].

13.4.2 Isogmentations and the resource calculus

It is natural to see in isogmentations a variation of *resource terms*, the terms of the finitary calculus called *resource calculus* used as target for the Taylor expansion of λ-terms [Ehrhard and Regnier, 2008] – this observation was nothing new, the similarity between resource terms and plays up to Melliès' homotopy had already been remarked before [Melliès, 2006, Tsukada and Ong, 2016]. For instance, along this similarity, the isogmentation in Figure 13.10 could be naturally be written syntactically as

$$\lambda f^{o\to o}.\, f\,[\lambda x^o.\, x, \lambda x^o.\, x]\,[\lambda x^o.\, x]\,.$$

More precisely, there is a bijection between isogmentations on (the arena for) a simple type A, and β-normal resource terms in a simply-typed extensional resource calculus – moreover the correspondence is straightforward, closely following the standard finite definability argument for innocent strategies (see Theorem 3.2.14) – but this correspondence does not account for non-normal terms and for the dynamics of resource terms.

To account for these dynamics, one difficulty is that isogmentations do not easily compose. Indeed, we expect isogmentations $\mathsf{q} \in \mathscr{A}(A \vdash B)$ and $\mathsf{p} \in \mathscr{A}(B \vdash C)$ to be *composable* if, writing $(\!|\mathsf{q}|\!) = \mathsf{x}_A \vdash \mathsf{x}_B$ and $(\!|\mathsf{p}|\!) = \mathsf{y}_B \vdash \mathsf{y}_C$ for the corresponding positions, we have $\mathsf{x}_B = \mathsf{y}_B$ – so that we may write $(\!|\mathsf{q}|\!) = \mathsf{x}_A \vdash \mathsf{x}_B$ and $(\!|\mathsf{p}|\!) = \mathsf{x}_B \vdash \mathsf{x}_C$. In that case, up to symmetry we can pick concrete representatives for q and p which share the same events on B, letting us synchronize them. However, this idea bumps against an issue similar to that exposed in Example 13.3.1: the resulting composition depends on the representative chosen…To eliminate this dependency on the representative, we first define a composition $\mathsf{p} \odot_\theta \mathsf{q}$ through a symmetry $\theta : \mathsf{x}_B \cong_B \mathsf{y}_B$, and then set

$$\mathsf{p} \odot \mathsf{q} = \sum_{\theta : \mathsf{x}_B \cong_B \mathsf{y}_B} \mathsf{p} \odot_\theta \mathsf{q}$$

a *formal sum* of isogmentations. This observation – that the composition of isogmentations yields a formal sum of isogmentations, rather than a single isogmentation – is reminiscent of the fact that substitution is the resource calculus also yields a sum

$$s\langle [u_1, \dots, u_n]/x \rangle = \sum_{\pi \in S(n)} s\langle u_{\pi(1)}/x_1 \rangle \dots \langle u_{\pi(n)}/x_n \rangle$$

if x has n occurrences in s (the result is 0 otherwise) – where $x_1, \dots x_n$ is an enumeration of all occurrences of x in t. This prompted us to extend the correspondence between isogmentations and normal resource terms to the *dynamics* of the resource calculus.

[8] In other terms, isogmentations are plays up to Melliès' homotopy equivalence on plays [Melliès, 2006].

This is what we achieve in [Blondeau-Patissier et al., 2023b]: we construct an interpretation of (simply-typed, extensional) resource terms as **strategies**, *i.e.* formal sums of isogmentations weighted by non-negative rational coefficients. For normal forms, this coincides with the bijection between normal resource terms and isogmentations mentioned above. But for non-normal terms, we show that the interpretation is invariant under reduction – also accounting for the coefficients arising throughout reduction. The proof of this result is structured by a new categorical model we call a **resource category**. Resource categories axiomatize the structure available in our category of pointer concurrent games. Much of their structure resembles notions of differential categories [Blute et al., 2020], but without an exponential modality.

13.4.3 Taylor expansion, extensional resource terms

The correspondence between normal resource terms and isogmentations invites the conjecture that the game semantics of a λ-term *is* the normal form of its Taylor expansion.

But the Taylor expansion typically targets the *pure* λ-calculus, for which this is not quite true: game semantics is an extensional model, while the Taylor expansion of a λ-term follows its *Böhm tree* (*i.e.*, the normal form of the Taylor expansion of a λ-term M is the Taylor expansion of its Böhm tree, as proved in [Ehrhard and Regnier, 2003]. So what we did instead is to devise a variant of the untyped resource calculus, the *extensional resource calculus*, obtained by seeking a syntax for the isogmentations on the universal arena of [Ker et al., 2002]. The extensional resource calculus has a nice syntactic theory, and is the target of an *extensional Taylor expansion* that captures $\mathcal{H}^*$ [Blondeau-Patissier et al., 2023a]. At the date of writing, the correspondence of this extensional Taylor expansion with pointer concurrent games is still work in progress.

13.5 Operational Concurrent Games

This line of work is in collaboration with Simon Castellan.

Operationalizing a denotational model. As hopefully demonstrated in this monograph, thin concurrent games offer a very expressive intensional presentation of the interactive behaviours of programs. The work presented here already accounts for complex programming features, and there is no obstacle in principle to extend it further.

Yet, something is puzzling. More than any other denotational semantics, game semantics has a strong operational flavour. To the trained game semanticist, each play or configuration of the interpretation of a program matches an operational execution trace: it is this informal connection that guided our introduction to game semantics in Chapter 3. Yet, actually *proving* that a concrete play or configuration is in the interpretation is discouragingly – and embarrassingly – tricky: one must unfold the definition of the interpretation of a λ-term in a cartesian closed category, which is itself obtained as the Kleisli category of a model of intuitionistic linear logic. Overall, the interpretation of a program unfolds to a large algebraic expression in the language of the categorical model

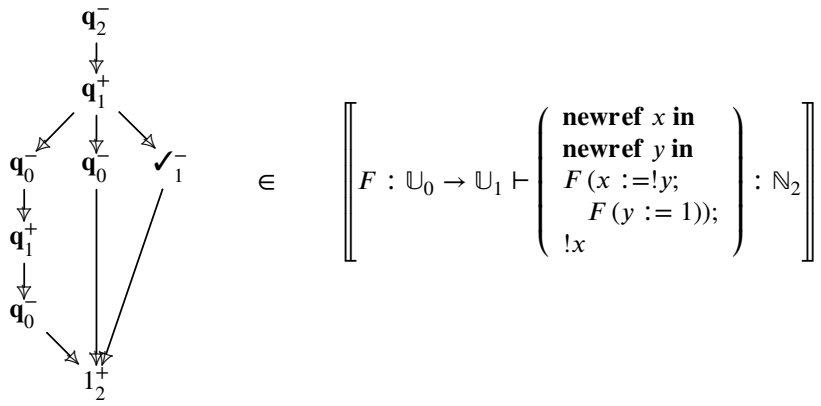


Figure 13.11: Is this a valid configuration for this program?

– for instance, the program of Figure 13.11 (partially) unfolds to an expression like

$$\mathsf{seq}\circ\left((\mathsf{ev}\circ((\llbracket F\rrbracket \otimes \llbracket x :=!y;\ F\,(y := 1)\rrbracket^{!})\circ(\delta \otimes \mathrm{id} \otimes \mathrm{id}))) \otimes \llbracket !x\rrbracket\right)\circ(\mathrm{id} \otimes \delta \otimes \mathrm{id})\circ(\mathrm{id} \otimes \mathsf{cell} \otimes \mathsf{cell}),$$

and answering the question in Figure 13.11 means putting forward an analogous configuration for each of the components of this expression, and proving they globally aggregate to the configuration claimed: this is not problematic in principle, but not efficient or even tractable in practice, so it is an obstacle to using game semantics to reason on concrete programs. Game semantics papers are full of examples, yet I feel confident in claiming that none but a few trivial ones have ever been proved formally[9].

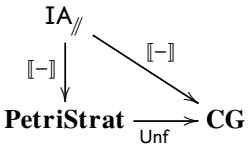
It seems we must choose between an interpretation that is compositional and one that it obtained directly operationally from the program. Or; we could try to devise an alternative – operational – method to generate the denotational object. Connecting operational and game semantics in that way is far from a new idea. Though it was clear from early work on game semantics [Danos et al., 1996] that interaction between strategies was related to execution by abstract machines, the idea to generate strategies by operational means was – to our knowledge – first suggested by Laird in his trace semantics for higher-order references [Laird, 2007] (with the explicit connection with game semantics later worked out by Jaber [Jaber, 2015]). In the 2010s, several works were proposed blurring the lines between operational and game semantics [Ghica and Tzevelekos, 2012, Levy and Staton, 2014], typically via LTSs dealing with open programs by sending and receiving messages from the environment; for these systems, compositionality is a theorem rather than a definition. But these developments do not yield strategies in the sense of previously established models. They also only deal with sequential deterministic programs, and it seems hard to extend LTS-based techniques to give a causal account of the execution of higher-order concurrent programs.

[9]This observation is meant to apply to all the game semantics literature, not just concurrent games.

But in fact, a powerful connection between operational and denotational semantics was already established significantly before the above, although not then presented as such: it is Baillot's result [Baillot, 1999] that Girard's *Geometry of Interaction* for IMELL generates the strategy obtained as its interpretation in so-called Abramsky-Jagadeesan-Malacaria (AJM) games [Abramsky et al., 2000]. We would like a similar correspondence for concurrent games as presented in this monograph. But how so? Geometry of Interaction is based on linear logic proof nets and Baillot's theorem relies on the rewriting theory of proof nets, but there are no such tools for $\mathtt{IA}_{/\!/}$.

Geometry of Interaction via Petri nets. In [Castellan and Clairambault, 2023] we ask the question: can we provide an analogue of Baillot's theorem for $\mathtt{IA}_{/\!/}$, *i.e.* a notion of execution for $\mathtt{IA}_{/\!/}$ programs that generates, just as a side effect of execution, the *same* concurrent strategy than that computed denotationally?

Our idea is the following: while there are no proof nets for $\mathtt{IA}_{/\!/}$, a proof net *equipped with the GoI token game* is an instance of an object that is very well-known in the verification and concurrency communities: a *coloured Petri net*. So we sidestep proof nets and aim to translate $\mathtt{IA}_{/\!/}$ programs directly into Petri nets. More precisely, we devise a notion we call a *Petri strategy*: a Petri strategy on game A is a (finite) coloured Petri net equipped with special transitions called *negative* and *positive* (on top of the old transitions now dubbed *neutral*). While neutral transitions have the same familiar mechanics as usual, negative and positive transitions are different: negative transitions have no precondition, instead triggered by an Opponent move in A and introducing a new token into the net. Dually, positive transitions have no postcondition, instead playing a positive move in A. Now, Petri nets satisfying adequate conditions support a notion of *unfolding* into event structures: each run has an implicit causal structure and those can be aggregated into an event structure. Accordingly, a *Petri strategy* is additionally required to unfold to a valid concurrent strategy. Finally, Petri strategies σ and τ can be easily composed, by putting them side by side and merging adequately matching negative and positive transitions. Altogether, we obtain a compositional interpretation of $\mathtt{IA}_{/\!/}$ into Petri strategies and an unfolding to concurrent strategies, such that



commutes. The interpretation of $\mathtt{IA}_{/\!/}$ into Petri nets is linear time, and can be regarded as a syntactic translation – a compilation – to an intermediate representation.

This is implemented and available at `https://ipatopetrinets.github.io/`. A screenshot of the application appears in Figure 13.12: on the right hand side we have the Petri strategy, with which the user can interact by clicking on available transitions. On the upper left corner is the program, and below is the configuration of the corresponding concurrent strategy reconstructed by the unfolding. The configuration is that of Figure 13.11 which, by our main result, proves that this is indeed a configuration of

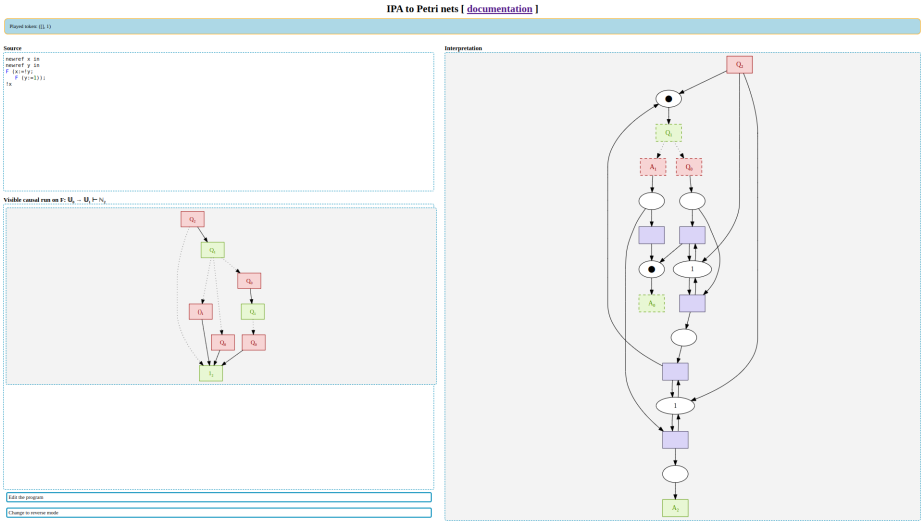


Figure 13.12: $\mathsf{IA}_{/\!/}$ to Petri nets

the concurrent strategy interpreting this program. Thanks to the causal nature of our interpretation, there is also a *reverse mode* that let us *undo* events, not necessarily in the opposite order in which they were played.

13.6 Miscellaneous

Finally, in this section, I give an account of other works that stand a bit on the side: though interesting, they are not part of the main thrust of our work on concurrent games.

13.6.1 Full Abstraction for Parallel-Or

This is joint work with Simon Castellan and Glynn Winskel.

In Chapter 10, I have presented our notion of parallel innocence, culminating in Chapter 12 to finite definability and intensional full abstraction for $\mathsf{PCF}_{/\!/}$. But in fact, our original objective was *not* $\mathsf{PCF}_{/\!/}$, but PCF plus *parallel-or*, the famous parallel primitive

$$\begin{aligned} \mathbf{por}\ \mathbf{ff}\ \mathbf{ff} &= \mathbf{ff} \\ \mathbf{por}\ \mathbf{tt}\ \bot &= \mathbf{tt} \\ \mathbf{por}\ \bot\ \mathbf{tt} &= \mathbf{tt} \end{aligned}$$

introduced by Plotkin in order to obtain a finite definability result with respect to Scott domains and continuous functions [Plotkin, 1977]. A central motivation in the original development of game semantics was precisely to ban **por** and obtain a cartesian closed category of sequential functions, and thus it felt deliciously ironic to attempt to close the

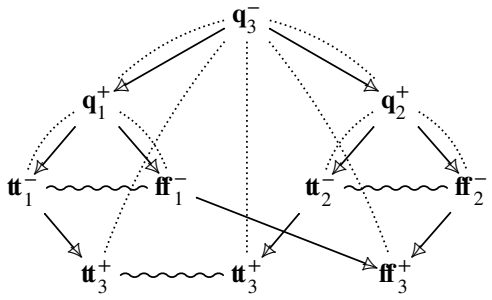


Figure 13.13: The concurrent strategy $\mathsf{por} : \mathbf{B}_1 \multimap \mathbf{B}_2 \multimap \mathbf{B}_3$

loop and capture the intensional behaviour of programs from PCF+**por**. Unfortunately, back in 2014, we bumped against a tricky problem and could not quite make this work.

Let us describe the problem. There is a natural concurrent strategy for **por**, described in Figure 13.13. When prompted by Opponent, the strategy plays two causally independent moves, prompting the evaluation of its two arguments in parallel. If both arguments return **ff**, then the strategy merges those two threads and returns **ff**. However, it suffices that *one* argument returns **tt** for the overall computation to return **tt**. What may come as a surprise is that although the extensional behaviour of **por** is functional (and thus deterministic), its intensional behaviour as given by this strategy is not. Indeed, each argument returning **tt** must cause an independent instance of the answer **tt** at toplevel. But those must conflict, since they match the same move in the game! In other words,

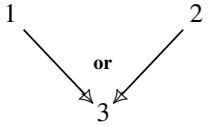
por tt tt

triggers an (invisible) race as the two threads compete to provide the same answer – operationally, **por tt tt** can be proved via two distinct rules, also witnessing this race.

How to formulate a notion of determinism accepting **por**, but refusing an actual nondeterministic choice? We could not figure out how to solve this back in 2014, hence reverted in [Castellan et al., 2015] to giving a model giving account of (causally deterministic) parallel evaluation in plain PCF.

Disjunctive causality. In [Castellan et al., 2017b] we solved that, constructing an intensionally fully abstract model for (an affine version of) PCF with **por**. The core of our solution is a variant of **CG** in which the semantics of **por** is *deterministic*.

The reasons why **por** is deterministic run quite deep: it boils down to the inability of event structures to express *disjunctive causality*, *i.e.* causal patterns such as



(13.3)

where the *same* event 3 may be caused *either* by 1 *or* by 2. But in fact, this is not a limitation of event structures in general, only of event structures of the kind we consider in this monograph, which are known as *prime event structures* in the literature. There are alternatives: for instance, *general event structures* [Winskel, 1980] allow events to be enabled in several distinct ways. We give an equivalent notion [Winskel, 1986]:

Definition 13.6.1. *A **configuration family** is a pair* $(|A|, A)$ *(often just written* A*) where* $|A|$ *is a set of events, and* A *is a set of **configurations**, finite subsets of* $|A|$ *satisfying:*

rooted:	$\emptyset \in A$
complete:	*for* $x, y \in A$*, if* $x \uparrow y$ *(they are **compatible**,* i.e. *there exists* $z \in A$ *such that* $x \subseteq z$ *and* $y \subseteq z$*), then* $x \cup y \in A$*.*
coincidence-free:	*if* $x \in A$*, for all distinct* $e_1, e_2 \in x$ *there exists* $y \subseteq x$ *such that* $y \in A$ *and* $e_1 \in y \Leftrightarrow e_2 \notin y$*.*

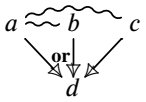
Configuration families do support disjunctive causality: the causal pattern of (13.3) is expressible with the configuration family with events $\{1, 2, 3\}$ and the configurations

$$\{\emptyset, \{1\}, \{2\}, \{1, 3\}, \{2, 3\}, \{1, 2, 3\}\},$$

and accordingly, *parallel-or* can be formalized as a configuration family that is *deterministic* in a suitable sense. So – why is this monograph entirely phrased in terms of configuration families, rather than prime event structures? The issue with configuration families is that they do not support *hiding* as is required in the composition of strategies:

Proposition 13.6.2. *Configuration families do* not *support hiding. There is a configuration family* A *and* $V \subseteq |A|$ s.t. $\{x \cap V \mid x \in A\}$ *is not a configuration family.*

Proof. Set $|A| = \{a, b, c, d\}$, and configurations those specified by the diagram



with $V = \{b, c, d\}$, *i.e.* we hide the event a. Then, the set of configurations obtained by hiding fails completeness: it contains $\{b\}, \{d\}, \{b, c, d\}$ but not $\{b, d\}$. □

Prime event structures support hiding because each event in the event structure post-hiding comes with a *unique, minimal* causal history in the event structure pre-hiding, a property that is lost here[10]. The failure of hiding comes from our inability to assign canonically, to any configuration of the hidden structure, a *witness* pre-hiding.

[10]One could object here that perhaps we could build concurrent games on something even more permissive than configuration families. But we must retain *coincidence-freeness*, because they are how we express the absence of causal loops (which distinguish concurrent games from relational semantics as illustrated in Section 10.4): any configuration should be reachable adding events one by one. Rooted coincidence-free families do support hiding, but they yield a non-associative composition of strategies.

Observational determinism. The key idea of our contribution is that we can recover hiding on configuration families, by requiring an additional *determinism* hypothesis:

Definition 13.6.3. *A **deterministic configuration family with polarities (dcfp)** is a configuration family A with $\mathrm{pol}_A : |A| \to \{-, 0, +\}$, satisfying the additional condition:*

$$\text{disjunctive deterministic:} \quad \textit{for all } x \subseteq_A^{\geq 0} y \textit{ and } x \subseteq z, \textit{ we have } y \cup z \in A.$$

where $x \subseteq_A^{\geq 0} y$ means that $x \subseteq y$ and $\mathrm{pol}_A(y \setminus x) \subseteq \{0, +\}$.

Here, the events with null polarities are those that we wish to hide, and indeed:

Proposition 13.6.4. *If A is a dcfp, then its **hiding** $A_\downarrow$, with events the set $V = \{a \in A \mid \mathrm{pol}(a) \neq 0\}$ and configurations those $x \cap V$ for $x \in A$, is still a dcfp.*

The key idea is as follows: a configuration $x \in A_\downarrow$ may have many different witnesses, *i.e.* configurations $y \in A$ such that $y \cap V = x$ – in general forming a family $(y_i)_{i \in I}$. But thanks to disjunctive determinism, all these witnesses are *compatible*, and

$$y = \bigcup_{i \in I} y_i \in A$$

is then a *canonical witness* for x. As in prime event structures, it is this ability to assign canonically to each $x \in A_\downarrow$ a witness in A that makes hiding go through. The situation here is somewhat dual to that for prime event structures: the canonical witness is the union of all witnesses, instead of the intersection as in the construction of **CG**.

Replaying the construction of **CG** based on the ideas above yields a compact closed category **Disj** of **disjunctive deterministic strategies**, where **por** admits a deterministic interpretation. In [Castellan et al., 2017b] we then proceed to build a hybrid model pairing **Disj** and **CG**, so that we have access to both disjunctive determinism and parallel innocence, letting us prove a finite definability result and intensional full abstraction.

13.6.2 A Semantic Proof of Herbrand's Theorem

This is joint work with Aurore Alcolei, Martin Hyland and Glynn Winskel.

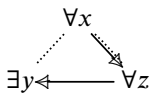
In a formula of (first-order) predicate logic, there is a clear causal dependency between instance of quantifiers that may be represented as a concurrent game, *e.g.*

$$[\![\forall x\, (\exists y\, P(x, y) \vee \forall z\, \neg P(x, z))]\!] \quad = \quad \begin{array}{ccc} & \forall x & \\ \exists y & & \forall z \end{array}$$

though this representation ignores for now all propositional connectives and atomic predicates. Then, a proof intuitively induces a further dependency between instances

of quantifiers that may be represented with a concurrent strategy; for instance we have

$$\left[\!\!\left[\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{\vdash P(x,z) \vee \neg P(x,z)}}{\vdash \exists y\, P(x,y), \neg P(x,z)}}{\vdash \exists y\, P(x,y), \forall z\, \neg P(x,z)}}{\vdash \exists y\, P(x,y) \vee \forall z\, \neg P(x,z)}}{\vdash \forall x (\exists y\, P(x,y) \vee \forall z\, \neg P(x,z))} \right]\!\!\right] =$$



witnessing the fact that the proof instantiates y with the eigenvariable z. The strategy only carries this causal dependency between quantifiers; the two players do not, for instance, play elements of a given structure in which the formula is evaluated, as would be the case in usual notions of games in the context of model theory.

We must still account for predicates and propositional connectives. To figure out whether such a strategy is correct, or *winning* in our terminology, we instantiate quantifiers following the dependency specified in the strategy, yielding a quantifier-free formula for which we assess whether it is a propositional tautology. In the example above, the causal dependency $\forall z \rightarrowtriangle \exists y$ in the strategy expresses that y should be replaced with z. Performing this substitution (and removing quantifiers), we obtain the formula

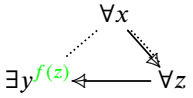
$$P(x,z) \vee \neg P(x,z)$$

which is a propositional tautology indeed, so the strategy is winning.

Strategies with witnesses. Beyond simply substituting one variable for another, a proof can instantiate an existential quantifier with a *first-order term*. Accordingly a strategy provides, for each positive move (existential quantifier), a first-order term whose free variables are the eigenvariables introduced by the universal quantifiers.

For instance, a variation of the example above could be

$$\left[\!\!\left[\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{\vdash P(x,f(z)) \vee \neg P(x,f(z))}}{\vdash \exists y\, P(x,y), \neg P(x,f(z))}}{\vdash \exists y\, P(x,y), \forall z\, \neg P(x,f(z))}}{\vdash \exists y\, P(x,y) \vee \forall z\, \neg P(x,f(z))}}{\vdash \forall x (\exists y\, P(x,y) \vee \forall z\, \neg P(x,f(z)))} \right]\!\!\right] =$$



decorating the quantifier $\exists y$ with the *witness* provided for y, here $f(z)$. The resulting instantiation of quantifiers yields a propositional tautology $P(x,f(z)) \vee \neg P(x,f(z))$.

Contraction. But classical proofs have access to the contraction rule, and as such may provide *several* witnesses for the same existential quantifier. So the interpretation of formulas is modified accordingly: instead of simply following the nesting of quantifiers,

it has countably many copies of each quantifier – though it is often more convenient to only draw the part of the game where only existential quantifiers are replicated.

For instance, the strategy for the usual proof of the Drinker's formula is:

$$\left[\!\!\left[\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{\vdash \neg P(c), P(y), \neg P(y) \vee \forall y\, P(y)}}{\vdash \neg P(c), P(y), \exists x\, (\neg P(x) \vee \forall y\, P(y))}}{\vdash \neg P(c), \forall y\, P(y), \exists x\, (\neg P(x) \vee \forall y\, P(y))}}{\vdash \neg P(c) \vee \forall y\, P(y), \exists x\, (\neg P(x) \vee \forall y\, P(y))}}{\vdash \exists x\, (\neg P(x) \vee \forall y\, P(y)), \exists x\, (\neg P(x) \vee \forall y\, P(y))}}{\vdash \exists x\, (\neg P(x) \vee \forall y\, P(y))} \right]\!\!\right] \quad = \quad \begin{array}{ccc} \exists x^{c} & & \exists x^{y} \\ \vdots & \nearrow & \vdots \\ \forall y & & \forall y \end{array}$$

assuming we have at least one constant symbol c – here the existential quantifier has two copies, witnessing the fact that the proof has a contraction. Instantiating quantifiers following this strategy, we obtain the quantifier-free formula

$$(\neg P(c) \vee P(y)) \vee (\neg P(y) \vee P(z))$$

which is a tautology – here as there are two instantiations for the existential quantifier, there are two copies of the subsequent sub-formula, linked with a disjunction.

In [Alcolei et al., 2018] we provide a denotational semantics for first-order classical proofs following this idea. For an existential formula $\exists x\, P(x)$, the interpretation of a proof corresponds exactly to a set of closed first-order terms $t_1, \dots, t_n$ such that $P(t_1) \vee \cdots \vee P(t_n)$ is a propositional tautology, recovering Herbrand's theorem. For more general formulas, we essentially recover Miller's expansion trees [Miller, 1987].

This work is original in that of all the developments considered in this monograph, this is the only one for which the games considered are not polarized (all minimal moves may not have the same polarity) and where causality is not alternating.

13.6.3 Revisiting Games Models of MALL

As mentioned in Section 6.5, (thin) concurrent games belong to a family of games model rejecting the hypothesis that plays should be totally chronologically ordered, initiated in the seminal paper [Abramsky and Melliès, 1999] and used in particular to construct a fully complete model for multiplicative additive linear logic.

Abramsky and Melliès' take on games is rather radical: their games have neither *moves* nor *plays*, but only a partial order of *positions*. More precisely, they are *dI-domains*, *i.e.* directed-complete, bounded-complete partial order satisfying two further axioms. If D is a dI-domain, we write $D^{\top}$ its extension with a top element $\top$ – it follows that $D^{\top}$ is a complete lattice. If E is an event structure (so that this automatically applies to a game A in the sense of Definition 6.1.3), then the domain $\mathscr{C}^{\infty}(E)$ of potentially infinite configurations is such a dI-domain. If A is a game, then $\mathscr{C}^{\infty}(A)$ is a new presentation of that game retaining only positions, forgetting individual moves.

Accordingly, a *strategy* in the sense of Abramsky and Melliès cannot play individual moves and acts on positions instead. Applied to a position x, a strategy σ *saturates* it, returning a new position that intuitively incorporates all moves that σ is prepared to play in x (or $\top$ if the strategy is undefined on the current position). More precisely, strategies are *continuous, stable closure operators*[11] – recall $f : D \to D'$ between dI-domains is **stable** iff for $x, y \in D$, if x, y are bounded then $f(x \wedge y) = f(x) \wedge f(y)$.

Definition 13.6.5. *A **closure-strategy** on dI-domain D, written $\sigma : D$, is a **continuous stable closure operator** on $D^\top$,* i.e. *a continuous function $\sigma : D^\top \to D^\top$ which is:*

extensive:	*for all $x \in D^\top$, $x \leq \sigma(x)$,*
idempotent:	*for all $x \in D^\top$, $\sigma(\sigma(x)) = \sigma(x)$),*
stable:	*there is a stable function $f : D \to D$ satisfying that for all $x \in D$ such that $\sigma(x) \neq \top$, $\sigma(x) = x \vee f(x)$.*

Stable implies *extensive*, however we state it separately as *extensive* and *idempotent* together they define a *closure operator*, a standard notion independently of stability.

The strategies may be composed; and Abramsky and Melliès construct a category **ClosOp** of dI-domains and closure-strategies, using which they build a fully complete model of linear logic. Naively, it might seem full completeness follows by "fixing" the non-associativity of composition in Blass games [Blass, 1992], and thus reinstating the equational laws expected of a model of MALL. In truth, the situation is more subtle: Abramsky and Melliès (implicitly) construct a model of a *polarized* version of MALL. It is then the *positionality* of strategies in **ClosOp** that allows the depolarization to MALL to induce a congruence, reinstating the desired equational laws and obtaining full completeness, as made more explicit in later work by Melliès [Melliès, 2005].

In [Clairambault, 2023], I revisit this story in the context of concurrent games on event structures. A lot of that paper consists in a synthesis of old ideas, already existing in the literature (though with a different technical underpinning).

From event structures to closure operators. But this article also has original contributions, with in particular the first proper treatment of the relationship between move-based game semantics and Abramsky and Melliès' model. Though the intuition seems clear, the definition is not so obvious (several incorrect attempts appear in the literature).

Given a (causally deterministic) concurrent strategy $\boldsymbol{\sigma} : A$, we must define

$$\mathsf{Clos}(\sigma) : \mathscr{C}^\infty(A) \to \mathscr{C}^\infty(A)$$

a closure-strategy. Given $x_A \in \mathscr{C}^\infty(A)$, by causal determinism, there is at most one $x^{\boldsymbol{\sigma}} \in \mathscr{C}^\infty(\boldsymbol{\sigma})$ such that $\partial_{\boldsymbol{\sigma}}\, x^{\boldsymbol{\sigma}} = x_A$. If we can find such $x^{\boldsymbol{\sigma}}$, then we may saturate it by taking a maximal $x^{\boldsymbol{\sigma}} \subseteq^+ y^{\boldsymbol{\sigma}}$, and set $\mathsf{Clos}(\boldsymbol{\sigma})(x_A) = y^{\boldsymbol{\sigma}}_A$. But what should we do if there is *no* $x^{\boldsymbol{\sigma}} \in \mathscr{C}^\infty(\boldsymbol{\sigma})$ such that $\partial_{\boldsymbol{\sigma}}\, x^{\boldsymbol{\sigma}} = x_A$? As it turns out, the correct answer is to play all positive moves of $\boldsymbol{\sigma}$ enabled in x_A, *even though* x_A might have positive moves that $\boldsymbol{\sigma}$ will never play. If doing so yields a conflict, then we return $\top$.

[11] Some additional conditions appear in the course of [Abramsky and Melliès, 1999], omitted here.

Chapter 14

Conclusions

14.1 The Work Done so Far

What do the following works have in common?

- Hyland and Ong's fully abstract games model for PCF [Hyland and Ong, 2000],
- Abramsky and McCusker's fully abstract games model for IA based on alternating non-innocent strategies [Abramsky and McCusker, 1996],
- Abramsky and Melliès' fully complete model for MALL based on closure operators [Abramsky and Melliès, 1999],
- The relational model and its syntactic presentation, non-idempotent intersection types [Girard, 1988],
- Ghica and Murawski's fully abstract model for $\mathsf{IA}_{/\!/}$ [Ghica et al., 2006],
- The weighted relational model [Laird et al., 2013],
- Alternating probabilistic game semantics [Danos and Harmer, 2000],
- Generalized species of structure [Fiore et al., 2008],
- The Taylor expansion of λ-terms [Ehrhard and Regnier, 2008],
- The GoI token machines [Danos and Regnier, 1996, Mackie, 1995].

The results presented in this monograph, accounting both for the work presented in details in Parts II and III and that presented more synthetically in Chapter 13, show that all these models are essentially obtained from concurrent games via forgetful operations: they are presentations of certain aspects of concurrent strategies. This list is of course a tiny fragment of denotational semantics, but this already gives a sign that we are getting closer to the *full interactive behaviour* mentioned in the introduction.

Our point is certainly not that this makes all these models obsolete. Concurrent games are too intensional for many purposes, and syntactic methods such as non-idempotent intersection types or Taylor expansion, by working directly on syntax, avoid the overhead brought by the categorical machinery of denotational semantics. More generally, different purposes call for different tools; concurrent games unifies these models but with the price of a high technical complexity which is not always desirable.

In contrast, we argue that this work strengthens all these models by inscribing them within a common semantic landscape and making them part of a joint mathematical theory. Many questions on their strengths, limitations and potential extensions can be attacked under the lense of concurrent games: for one example, how should non-idempotent intersection types and Taylor expansion of programs best deal with branching effects (such as non-determinism and parallelism) or quantitative effets (such as probabilities)? Concurrent strategies support all this and extend Taylor expansion conservatively, so the question above is reduced to the following: can concurrent games be described more syntactically, in the style of intersection types or Taylor expansion?

14.2 Open Problems

Next, here are some open problems around the theory presented in this monograph. Those are not necessarily long-term research objectives or directions (though some could be), but rather technical difficulties encountered during the technical developments, that we have not been able to solve and have had to avoid or work around.

14.2.1 Non-deterministic parallel innocence

In Chapter 10 we have developed the theory of *parallel innocent* causal strategies; but our notion of innocence (Definitions 10.1.8 and 10.1.11) smuggles in *causal determinism*, *i.e.* the absence of minimal conflicts between Player moves. But parallel innocence is intended to ban state and more generally side-channel interference between threads, there is no reason why it would necessarily enforce determinism. For instance, what are the strategies generated by $\mathsf{PCF}_{/\!/}$ plus a non-deterministic choice operator $\veebar$?

First of all, in the case of *sequential innocence* there is no problem, as spelled out in Definition 13.2.1: the problem appears only for non-deterministic *parallel* innocence. First, the conditions of Definition 13.2.1 do not work as-is: *locally sequential* is wrong for obvious reasons (as it forbids parallel branching); *locally conflicting* forbids parallel-or (see Figure 13.13) which should certainly be parallel innocent.

Recall that parallel innocence started with *pre-innocence* (Definition 10.1.8), which follows the idea that Player cannot causally merge threads forked by Opponent. It is natural to attempt to follow the same idea for conflict, and define:

Definition 14.2.1. *For A a board, a visible $\sigma : A$ is* ***#-pre-innocent*** *iff*

#-pre-innocence: *If* $m_1^+ \sim\!\!\!\sim_\sigma m_2^+ \in \sigma$ *and* $\rho_1 \twoheadrightarrow m_1^+, \rho_2 \twoheadrightarrow m_2^+ \in \mathsf{gcc}(\sigma)$,
then $\min(\rho_1) = \min(\rho_2)$ *and their least distinct moves are positive.*

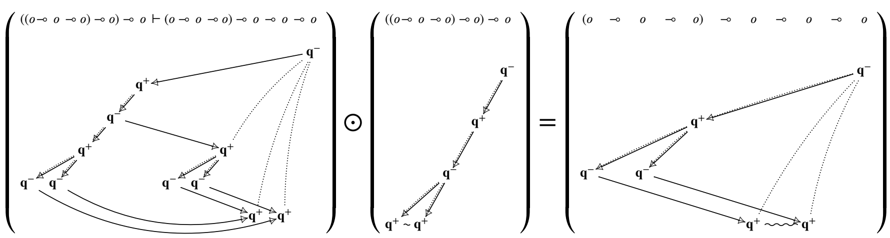


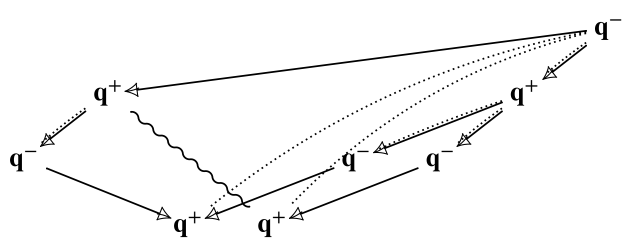
Figure 14.1: The subtlety of non-deterministic parallel innocence

In other words, *Player cannot impose conflict between threads forked by Opponent*.

It is quite intuitive that strategies obtained from $\mathsf{PCF}_{/\!/}$ plus $\oslash$ must satisfy this, as $\oslash$ can only be used to generate conflict between moves that share the same causal predecessors. And for quite some time, we were convinced we had a proof that this (in the presence of visibility) was stable under composition. With that condition, it is then tempting to define parallel innocence without causal determinism but with *#-pre-innocence*, so that it also covers non-deterministic parallel innocence.

However, #-pre-innocence is not stable under composition. The strategy

$$(o \multimap o) \vdash o \multimap o \multimap (o \multimap o \multimap o) \multimap o$$



is visible, pre-innocent and #-pre-innocent, but its composition with the identity on o is not, as the minimal conflict is transported to the two final positive moves, though the gccs leading to them are forked by Opponent. Therefore the open question is:

Open question 14.2.2. *Define a condition of non-deterministic parallel innocence.*

Non-deterministic parallel innocence should compose, and support the interpretation of $\mathsf{PCF}_{/\!/}$ *with* $\oslash$*, and of parallel-or. Additionally, non-deterministic parallel innocent strategies which are causally deterministic should be parallel innocent in the sense of this monograph. Non-deterministic parallel innocent strategies which are* sequential *should be non-deterministic sequential innocent as in Definition 13.2.1.*

In other word, the problem is to *disentangle* non-determinism and parallelism.

Note that non-deterministic parallel innocence is not going to be conservative with respect to deterministic parallel innocence (which probably means that we do not have

deterministic parallel innocence quite right either) as illustrated in Figure 14.1. By composing a deterministic parallel innocent strategy with a sequential non-deterministic innocent strategy (that is obviously definable in a non-deterministic linear λ-calculus with non-deterministic choice), we break #-pre-innocence. Something is wrong with the deterministic parallel innocent strategy of Figure 14.1, but it seems that it cannot be banned with simple, local forbidden patterns as in pre-innocence and #-pre-innocence; our best guess is a global correctness criterion in the style of linear logic proof nets.

Once equipped with a satisfactory notion of non-deterministic innocence, it would be natural to further seek a corresponding extension of the sequential probabilistic innocence of Definition 13.2.2 to a full parallel probabilistic innocence.

14.2.2 Proper determinism

An issue strongly related to proper non-deterministic parallel innocence, is that of determinism. As we have seen, $\mathsf{IA}_{/\!/}$ is a genuinely non-deterministic language, due to the combination of parallelism and state causing races in the memory. Accordingly, in Part III, our conditions of *parallel innocence* and *sequentiality* each had to reimpose determinism; but this was done is radically different ways.

Recall indeed *causal determinism*, a part of parallel innocence:

Definition 10.1.1. *Consider $\boldsymbol{\sigma} : A \vdash B$ a causal strategy.*
*We say $\boldsymbol{\sigma}$ is **causally deterministic** if for all $s \sim\!\!\!\sim_{\boldsymbol{\sigma}} s'$ in $\boldsymbol{\sigma}$, $\mathrm{pol}(s) = \mathrm{pol}(s') = -$.*

And recall also *sequential determinism*, a part of sequentiality:

Definition 11.1.4. *A strategy $\boldsymbol{\sigma} \in \mathbf{NTCG}(A, B)$ is **sequential** if:*

$$\begin{array}{rl} \text{pointed:} & \text{if } s \in \boldsymbol{\sigma}\text{, there is a unique } \mathrm{init}(s) \leq_{\boldsymbol{\sigma}} s \text{ minimal in } \boldsymbol{\sigma}, \\ \text{sequential determinism:} & \text{if } tn_1^+, tn_2^+ \in \Updownarrow\!\uparrow\text{-}\mathscr{L}(\boldsymbol{\sigma})\text{, then } n_1 = n_2, \\ \text{sequential visibility:} & \text{if } s \in \Updownarrow\!\uparrow\text{-}\mathscr{L}(\boldsymbol{\sigma})\text{, writing } \ulcorner s \urcorner = s_1 \ldots s_n \text{ and } 1 \leq j \leq n, \\ & \text{if } s_j \text{ is non-minimal then there is } 1 \leq i < j \text{ s.t. } \mathsf{j}(s_j) = s_i, \end{array}$$

where the justifier $\mathsf{j}(m)$ *of $m \in \boldsymbol{\sigma}$ refers to Definition 10.2.1.*

Open question 14.2.3. *Define a* proper notion of determinism*, unifying these two.*

Note that such a proper determinism *should not be compositional* – indeed both IA and $\mathsf{PCF}_{/\!/}$ are deterministic languages, but their union is not. However, proper determinism should become compositional in conjunction with either non-deterministic parallel innocence and non-deterministic sequentiality.

As a reasonable starting point, it might be that in the presence of (an appropriate notion of) non-deterministic parallel innocence, *sequential determinism* implies causal determinism. Indeed, proving that amounts to showing that all Player immediate conflicts are reachable in an alternating linearization, which seems plausible, if conflicting patterns are anything like the *globules* of Definition 10.5.1.

14.2.3 All well-bracketings

Rather than the clear statement of an open problem, this is a description of an unfortunate state of affairs in this monograph: namely, the status of well-bracketing – understood broadly as a constraint on the function call/return discipline. In this monograph, we have seen four different mechanisms akin to well-bracketing.

Firstly, of course, we have seen the traditional **chronological well-bracketing** of Section 3.2.2, explicitly requiring that calls and returns, ordered chronologically, are... well-bracketed. This is of course very natural, though restricted to sequential plays.

Secondly, in non-alternating games we have seen **logical well-bracketing** (Section 5.1.1), also used in concurrent games when constructing $\multimap$-$\mathbf{Strat}$ (Section 9.3.1). Logical well-bracketing asks that a function may only call its arguments *before* it returns, and may only return once all called arguments have returned. A logically well-bracketed play may not be chronologically well-bracketed even if alternating (Figure 5.3).

Next, in the definition of **NTCG** we have deployed *winning conditions* (Section 8.2.2) imported from [Melliès and Tabareau, 2007]. This **positional well-bracketing**, avoiding the explicit disctinction of Questions and Answers, treats well-bracketing as a linearity constraint: a call may only be answered once. The *winning* mechanism enforces that winning strategies are never the first to break this linearity constraint, which hits surprisingly close to chronological well-bracketing at least for sequential innocent total strategies. In general, positional well-bracketing does not subsume the other more interactive notions, and *e.g.* cannot guarantee finite definability result. On the other hand, it is by far the best way to formulate the relational collapse of Section 10.4.

Finally, we have seen *globularity*, which is a **causal well-bracketing**: it ensures that the forking of merging of threads also respects the call/return discipline for parallel innocent strategies, in that a strategy may not merge threads with calls still pending since their fork. Again this is not guaranteed by our previous notions, bit is nevertheless required for the definability result of Chapter 12.

This profusion of subtly different notions of well-bracketing is messy and unsatisfactory. It feels that they could be unified in principle, but at the cost of requiring importing even more intensional information onto strategies, such as the *concurrency pointers* of [Laird, 2001b] to distinguish causal links coming from the term structure from those going through side-channel communication – this may allow us to extend causal well-bracketing beyond parallel innocent strategies. But does this actually work? And is it worth making the model even more complex? We leave this as an exercise to the reader.

14.2.4 Observing causality

Typically, concurrent games models are not fully abstract; as argued in Section 1.6 the strength of concurrent games is not to capture observational equivalence, but to record intensional information. Nevertheless, it is natural to wonder under which circumstances this causal information might be observed. From full abstraction results such as Theorem 5.1.12, this seems to boil down to asking when causal strategies may be entirely recovered (up to positive iso) from their non-alternating plays.

As presented in Section 13.1.1, the causal information *cannot* be reconstructed from plays in general. But we think it might be under the right circumstances: if each move can be replicated and the strategy is parallel innocent, then in principle causality may be observed by duplicating Opponent moves and observing the triggered duplications of Player moves. Indeed in examples, it is usually easy to define plays that separate distinct parallel innocent strategies. This brings us to the conjecture:

Conjecture 14.2.4. *Consider A a mixed board such that no moves are answers, and $\sigma, \tau : A$ two (possibly finite) parallel innocent strategies.*
If $\circlearrowleft\text{-}\mathsf{Unf}(\sigma) = \circlearrowleft\text{-}\mathsf{Unf}(\tau)$, then $\sigma \approx \tau$.

If this was the case, it would likely entail that two terms in $\mathsf{PCF}_{/\!/}$ yield *isomorphic* strategies if and only if they cannot be distinguished by contexts from $\mathsf{IA}_{/\!/}$ extended with a control operator. However, one should not underestimate this problem: given a non-alternating play s, there may in principle be several parts of the strategy causing it – thus it is hard to pin down what one individual play teaches us about a strategy.

Perhaps the right path to that conjecture is via another:

Conjecture 14.2.5. *Consider A a mixed board such that no moves are answers, and $\sigma, \tau : A$ two (possibly finite) parallel innocent strategies.*
If $(\!|\sigma|\!) = (\!|\tau|\!)$, then $\sigma \approx \tau$.

In [Blondeau-Patissier and Clairambault, 2021], we have already proved this for finite sequential innocent strategies. It would imply Conjecture 14.2.4, since the relational collapse can be extracted from the plays. The proof method for the sequential innocent case seems to extend in principle, but it becomes very technical and so far we have not managed to work out the details.

14.3 Perspectives and Future Directions

While the open problems above may tie loose ends and offer good avenues for short-term technical developments, they do not give long-term overarching research directions for the future. While we believe that the work reported here gives valuable steps in the direction presented in our introduction, the long-term goal is to use concurrent games to help make denotational semantics a connected mathematical theory. This suggests a *wealth* of research directions for the future, and we now introduce some of them.

The cube of branching structure. The core contribution of this monograph is the *disentangling of parallelism and state*, leading to a semantic square in the style of that presented in Section 3.3.3. This fits together parallelism and state in a single model, but why only those? After all, there are multiple other programming features studied in the game semantics litterature – to cite only a few: control operators, exceptions, higher-order references, coroutines, probabilistic choice, etc. In principle, all of those semantics could be reformulated within concurrent games, but it does not seem such a good research strategy to do it exhaustively for all of them – if only because such a massive undertaking would prevent us from investigating other interesting questions.

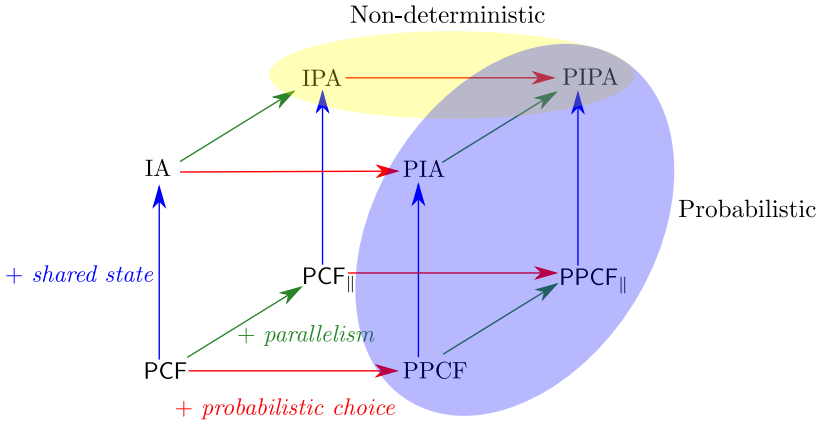


Figure 14.2: The cube of branching structures

So we introduce the following criterion. Some of the features above change the rules following which control flows through the program, but do not change the *geometry* of the control flow – this includes control operators, exceptions, higher-order references and coroutines: adding any of those to a sequential deterministic programming language yields a sequential deterministic programming language.

In contrast, some programming features heavily affect the geometric shape of execution: in this monograph, we have investigated in depth *parallelism*, and also *non-determinism* arising from the combination of parallelism and state. *Non-determinism* makes execution a tree, *parallelism* makes it a partial order, and the two together turn execution into an *event structure*. This makes concurrent games based on event structures perfect for this combination; we argue it is their ability to represent explicitly this *branching structure* that made possible the results presented in Part III. We expect giving a proper treatment of a new branching structure is a more fundamental problem than dealing with other programming features: after all, most of the non-branching effects mentioned above did *not* require changing much of the basic sequential deterministic game semantics canvas; unlike the results presented in this monograph.

There is a major branching structure that is not properly covered within this monograph: *probabilistic choice*. By itself, it differs from non-deterministic choice by the addition of quantitative weigths representing probabilities (as described in Section 13.2). But more fundamentally, it is the branching structure in the presence of all three of parallelism, non-determinism and probabilistic choice that poses the real challenge. This combination of branching effects is embodied by the cube in Figure 14.2.

Its left hand face is that studied in Part III. Its front face seems easy to handle working from the results of Section 13.2. The real challenge is in dealing with the all three effects together, which requires us to elaborate concurrent strategies to deal with the combination of non-determinism and probabilistic choice – perhaps based on de Visme's *mixed event structures* [de Visme, 2019]. But besides having to develop a new theory of con-

current games, constructing a semantic cube to match Figure 14.2 means we *must* solve the open problems of Section 14.2.1 and give proper notions of probabilistic and non-deterministic parallel innocence. So this is a long-term objective that will require many developments before we can properly tackle it.

Call-by-value and evaluation order. A major element missing from all the development presented in this monograph, is evaluation order. All the languages considered here are call-by-name. This is mainly for historical reasons, because we connect to earlier work in the literature which is set in call-by-name. There is no obstacle, in principle, to replaying the results presented here in call-by-value: in particular, it has been known for a long time how to build games models for call-by-value languages [Abramsky and McCusker, 1997, Honda and Yoshida, 1999]. Our point here is not to redo all the previous work in call-by-value, as this would be a major undertaking offering comparatively little conceptual insight – though some of it does need to be redone.

There are several frameworks for languages and models encompassing both call-by-name and call-by-value: the main example is of course Levy's call-by-push-value [Levy, 1999] which focuses on the impact of evaluation order on effects, but also others such as the *bang calculus* (inspired from Girard's translations of intuitionistic logic into linear logic), which instead captures evaluation order via its impact on resource consumption [Ehrhard and Guerrieri, 2016]. Or finally, system L [Curien et al., 2016], encompassing both. This latter system, to our knowledge, does not yet have a concrete games model, which is unfortunate since it is the family of models that gives a concrete account of both effects and resources. A concurrent games analysis of such systems could be enlightening and also help us understand how *parallelism* fits into this picture.

In fact, there is currently increasing research activity revisiting classical topics (such as the pure λ-calculus, solvability, Böhm trees etc) in call-by-value. It seems that adequate notions of infinite normal forms in call-by-value include permutative equivalences inducing the structure of a DAG, rather than simply a tree [Accattoli et al., 2023], meaning that it might naturally yield a parallel innocent concurrent strategy. A natural question is that of full abstraction for the pure λ-calculus in call-by-value, which is still missing; drawing inspirations from the results of this monograph this also offers enticing areas of investigation, connecting concurrent games with Taylor expansion [Kerinec et al., 2020] and intersection types [Arrial et al., 2023] in call-by-value.

Finally, a major objective is to extend our geometry of interaction multi-token machine presented in Section 13.5 to a call-by-value language – this is a prerequisite before we can seriously investigate if it might be useful in practice. We have preliminary ideas about how such a machine should work exploiting subtle memoization mechanisms in Petri nets, but much ground work is needed: in call-by-name, the validity of the machine rests on the adequate concurrent games model of $\mathsf{IA}_{/\!/}$ presented in Chapter 9, and no such model exists yet for a higher-order call-by-value stateful concurrent language.

Taylor expansion and intersection types. As reported in Section 13.4.3, for the pure λ-calculus, concurrent games offer a representation of programs very close to that proposed by the Taylor expansion of λ-terms [Ehrhard and Regnier, 2008] – more pre-

cisely, concurrent strategies form what could be regarded as a very rigid Taylor expansion. With respect to game semantics, the Taylor expansion of λ-terms has the advantage of being a syntactic method. This makes it much easier to deploy, and with a much tighter connection with the original syntax. In return, concurrent games are much more general, supporting with in the same framework many additional programming features including non-determinism, concurrency, probabilistic choice, shared memory, etc.

This suggests a number of natural questions: to what extent could concurrent games models be presented syntactically beyond the pure λ-calculus? How does it compare with existing extensions of the Taylor expansion, such as for non-deterministic choice [Olimpieri and Vaux Auclair, 2022] or probabilistic choice [Dal Lago and Leventis, 2019]? Can we use concurrent games as a guide to construct intersection type systems or Taylor expansions for effectful languages (such as $\mathsf{IA}_{/\!/}$), giving us the expressiveness of concurrent games models without all the accompanying technical complications?

Categorification. Games models lack uniformity; while they often formalize similar ideas and follow a common methodology, different models usually rest on very different technical underpinnings. The work presented in this monograph attempts to cope with this state of affairs by linking the models together, providing functorial bridges that make explicit how they relate to concurrent games. But a complementary methodology is to capture the construction of various games models as instances of one common categorical construction; notable works in this direction include [Eberhart and Hirschowitz, 2018] and more recently, Melliès' *template games* [Melliès, 2019a, Melliès, 2019b]. However, for the time being, concurrent games evade this categorical unification. Melliès' original template games lack mechanisms to eliminate deadlocking synchronizations – a new, more elaborate version [Melliès, 2021] does capture the *interaction* of concurrent strategies (see Proposition 6.2.13), but not the composition, for subtle reasons. So the question remains open: how should template games be generalized so as to encompass concurrent games?

A strongly related topic is the proper categorical treatment of *symmetries* in concurrent games. Recently [Clairambault and Forest, 2023], we constructed a cartesian closed bicategory of certain *spans of groupoids*, reproducing categorically much of the structure of symmetries in thin concurrent games (but, like template games, also failing to account for deadlocks in compositions). It is our conviction that a proper categorification of game semantics should rest on thin spans of groupoids. The link with template games is fascinating, as the two models use completely different mechanisms to account for replication of resources – thin spans are *thin* as in thin concurrent games, while template games are *saturated* (as in Section 7.1.2). Another fundamental question is the link with distributors and generalized species of structure; we give some elements in [Clairambault et al., 2023b] but this should be completed to account for inclusion of configurations and not just symmetries, and should also include a proper treatment of the linear bicategorical (or double-categorical) structure.

Thin spans of groupoids really are *thin concurrent games without concurrent games*. It seems that many of our passed achievements with thin concurrent games (typically, the enrichment with quantitative valuations and the corresponding relational collapses,

up to generalized species of structure) are actually independent from the basic game mechanisms, and should really be carried out in thin spans of groupoids instead. We should spell this out and investigate extensions made possible by the fact that we are no longer hindered by the intricate combinatorics of strategy composition.

Qualitative concurrent games. Thin concurrent games are a *quantitative* model, in the sense that they record explicitly the *multiplicity* of resource usage – even more, each individual move has a separate identity, formalized as a copy index. In categorified relational models such as distributors, it is possible to change the exponential modality to make it *qualitative*; this is done by changing the *morphisms* in $!A$ so as to allow contraction and weakening. In principle, the same could be done in thin concurrent games, by generalizing *symmetries* to encompass more general relations between configurations that may not preserve multiplicity. There are fundamental questions to solve regarding how such relations behave with respect to polarity. A first concrete objective would be to construct such a generalization of thin concurrent games, and deduce a new, more concrete proof of Ehrhard's *extensional collapse* theorem [Ehrhard, 2012] linking qualitative and quantitative models.

One interest of such a concrete proof is that we expect it to generalize in a situation with *infinite multiplicities*. More precisely, we aim to generalize the relational collapse presented in this monograph to a collapse to the *infinitary relational model* [Grellois and Melliès, 2015], also accounting for infinite configurations. Paired with an infinitary version of Ehrhard's extensional collapse, we expect this is the key ingredient to a purely semantic proof of the decidability of the higher-order model-checking problem, which would be a beautiful illustration of the strength of thin concurrent games.

Bibliography

[Abramsky, 2003] Abramsky, S. (2003). Sequentiality vs. concurrency in games and logic. *Math. Struct. Comput. Sci.*, 13(4):531–565.

[Abramsky et al., 1998] Abramsky, S., Honda, K., and McCusker, G. (1998). A fully abstract game semantics for general references. In *Thirteenth Annual IEEE Symposium on Logic in Computer Science, Indianapolis, Indiana, USA, June 21-24, 1998*, pages 334–344. IEEE Computer Society.

[Abramsky et al., 2000] Abramsky, S., Jagadeesan, R., and Malacaria, P. (2000). Full abstraction for PCF. *Inf. Comput.*, 163(2):409–470.

[Abramsky and McCusker, 1996] Abramsky, S. and McCusker, G. (1996). Linearity, sharing and state: a fully abstract game semantics for idealized algol with active expressions. *Electron. Notes Theor. Comput. Sci.*, 3:2–14.

[Abramsky and McCusker, 1997] Abramsky, S. and McCusker, G. (1997). Call-by-value games. In *CSL*, volume 1414 of *Lecture Notes in Computer Science*, pages 1–17. Springer.

[Abramsky and Melliès, 1999] Abramsky, S. and Melliès, P. (1999). Concurrent games and full completeness. In *LICS*, pages 431–442. IEEE Computer Society.

[Accattoli et al., 2023] Accattoli, B., Faggian, C., and Lancelot, A. (2023). Normal form bisimulations by value. *CoRR*, abs/2303.08161.

[Alcolei et al., 2018] Alcolei, A., Clairambault, P., Hyland, M., and Winskel, G. (2018). The true concurrency of herbrand's theorem. In *CSL*, volume 119 of *LIPIcs*, pages 5:1–5:22. Schloss Dagstuhl - Leibniz-Zentrum für Informatik.

[Alcolei et al., 2019] Alcolei, A., Clairambault, P., and Laurent, O. (2019). Resource-tracking concurrent games. In *FoSSaCS*, volume 11425 of *Lecture Notes in Computer Science*, pages 27–44. Springer.

[Altenkirch et al., 2010] Altenkirch, T., Chapman, J., and Uustalu, T. (2010). Monads need not be endofunctors. In *International Conference on Foundations of Software Science and Computational Structures*, pages 297–311. Springer.

[Arnold and Niwiński, 2001] Arnold, A. and Niwiński, D. (2001). *Rudiments of μ-calculus*. Elsevier.

[Arrial et al., 2023] Arrial, V., Guerrieri, G., and Kesner, D. (2023). Quantitative inhabitation for different lambda calculi in a unifying framework. *Proc. ACM Program. Lang.*, 7(POPL):1483–1513.

[Baillot, 1999] Baillot, P. (1999). *Approches dynamiques en sémantique de la logique linéaire: jeux et géométrie de l'interaction*. PhD thesis, Aix-Marseille 2.

[Baillot et al., 1997a] Baillot, P., Danos, V., Ehrhard, T., and Regnier, L. (1997a). Believe it or not, ajm's games model is a model of classical linear logic. In *LICS*, pages 68–75. IEEE Computer Society.

[Baillot et al., 1997b] Baillot, P., Danos, V., Ehrhard, T., and Regnier, L. (1997b). Timeless games. In *Computer Science Logic, 11th International Workshop, CSL '97, Annual Conference of the EACSL, Aarhus, Denmark, August 23-29, 1997, Selected Papers*, pages 56–77.

[Blass, 1992] Blass, A. (1992). A game semantics for linear logic. *Ann. Pure Appl. Log.*, 56(1-3):183–220.

[Blondeau-Patissier and Clairambault, 2021] Blondeau-Patissier, L. and Clairambault, P. (2021). Positional injectivity for innocent strategies. In Kobayashi, N., editor, *6th International Conference on Formal Structures for Computation and Deduction, FSCD 2021, July 17-24, 2021, Buenos Aires, Argentina (Virtual Conference)*, volume 195 of *LIPIcs*, pages 17:1–17:22. Schloss Dagstuhl - Leibniz-Zentrum für Informatik.

[Blondeau-Patissier et al., 2023a] Blondeau-Patissier, L., Clairambault, P., and Vaux Auclair, L. (2023a). Extensional taylor expansion. *CoRR*, abs/2305.08489.

[Blondeau-Patissier et al., 2023b] Blondeau-Patissier, L., Clairambault, P., and Vaux Auclair, L. (2023b). Strategies as resource terms, and their categorical semantics. In Gaboardi, M. and van Raamsdonk, F., editors, *8th International Conference on Formal Structures for Computation and Deduction, FSCD 2023, July 3-6, 2023, Rome, Italy*, volume 260 of *LIPIcs*, pages 13:1–13:22. Schloss Dagstuhl - Leibniz-Zentrum für Informatik.

[Blute et al., 2020] Blute, R., Cockett, J. R. B., Lemay, J. P., and Seely, R. A. G. (2020). Differential categories revisited. *Appl. Categorical Struct.*, 28(2):171–235.

[Boudes, 2009] Boudes, P. (2009). Thick subtrees, games and experiments. In *Typed Lambda Calculi and Applications, 9th International Conference, TLCA 2009, Brasilia, Brazil, July 1-3, 2009. Proceedings*, pages 65–79.

[Calderon and McCusker, 2010] Calderon, A. C. and McCusker, G. (2010). Understanding game semantics through coherence spaces. *Electr. Notes Theor. Comput. Sci.*, 265:231–244.

[Cartwright et al., 1994] Cartwright, R., Curien, P., and Felleisen, M. (1994). Fully abstract semantics for observably sequential languages. *Inf. Comput.*, 111(2):297–401.

[Castellan, 2015] Castellan, S. (2015). La stratégie de la fourchette. In *Vingt-sixièmes Journées Francophones des Langages Applicatifs (JFLA 2015)*.

[Castellan and Clairambault, 2016] Castellan, S. and Clairambault, P. (2016). Causality vs. interleavings in concurrent game semantics. In *CONCUR*, volume 59 of *LIPIcs*, pages 32:1–32:14. Schloss Dagstuhl - Leibniz-Zentrum für Informatik.

[Castellan and Clairambault, 2021] Castellan, S. and Clairambault, P. (2021). Disentangling parallelism and interference in game semantics. Submitted.

[Castellan and Clairambault, 2023] Castellan, S. and Clairambault, P. (2023). The geometry of causality: Multi-token geometry of interaction and its causal unfolding. *Proc. ACM Program. Lang.*, 7(POPL):689–717.

[Castellan et al., 2018a] Castellan, S., Clairambault, P., Hayman, J., and Winskel, G. (2018a). Non-angelic concurrent game semantics. In Baier, C. and Lago, U. D., editors, *Foundations of Software Science and Computation Structures - 21st International Conference, FOSSACS 2018, Held as Part of the European Joint Conferences on Theory and Practice of Software, ETAPS 2018, Thessaloniki, Greece, April 14-20, 2018, Proceedings*, volume 10803 of *Lecture Notes in Computer Science*, pages 3–19. Springer.

[Castellan et al., 2018b] Castellan, S., Clairambault, P., Paquet, H., and Winskel, G. (2018b). The concurrent game semantics of probabilistic PCF. In *LICS*, pages 215–224. ACM.

[Castellan et al., 2017a] Castellan, S., Clairambault, P., Rideau, S., and Winskel, G. (2017a). Games and strategies as event structures. *Log. Methods Comput. Sci.*, 13(3).

[Castellan et al., 2014] Castellan, S., Clairambault, P., and Winskel, G. (2014). Symmetry in concurrent games. In Henzinger, T. A. and Miller, D., editors, *Joint Meeting of the Twenty-Third EACSL Annual Conference on Computer Science Logic (CSL) and the Twenty-Ninth Annual ACM/IEEE Symposium on Logic in Computer Science (LICS), CSL-LICS '14, Vienna, Austria, July 14 - 18, 2014*, pages 28:1–28:10. ACM.

[Castellan et al., 2015] Castellan, S., Clairambault, P., and Winskel, G. (2015). The parallel intensionally fully abstract games model of PCF. In *30th Annual ACM/IEEE Symposium on Logic in Computer Science, LICS 2015, Kyoto, Japan, July 6-10, 2015*, pages 232–243. IEEE Computer Society.

[Castellan et al., 2017b] Castellan, S., Clairambault, P., and Winskel, G. (2017b). Observably deterministic concurrent strategies and intensional full abstraction for parallel-or. In *FSCD*, volume 84 of *LIPIcs*, pages 12:1–12:16. Schloss Dagstuhl - Leibniz-Zentrum für Informatik.

[Castellan et al., 2019] Castellan, S., Clairambault, P., and Winskel, G. (2019). Thin games with symmetry and concurrent hyland-ong games. *Log. Methods Comput. Sci.*, 15(1).

[Chappe et al., 2023] Chappe, N., He, P., Henrio, L., Zakowski, Y., and Zdancewic, S. (2023). Choice trees: Representing nondeterministic, recursive, and impure programs in coq. *Proc. ACM Program. Lang.*, 7(POPL):1770–1800.

[Clairambault, 2010] Clairambault, P. (2010). *Logique et Interaction : une Étude Sémantique de la Totalité. (Logic and Interaction : a Semantic Study of Totality)*. PhD thesis, Paris Diderot University, France.

[Clairambault, 2023] Clairambault, P. (2023). A tale of additives and concurrency in game semantics. In *Samson Abramsky on Logic and Structure in Computer Science and Beyond*, pages 363–414. Springer.

[Clairambault and de Visme, 2020] Clairambault, P. and de Visme, M. (2020). Full abstraction for the quantum lambda-calculus. *Proc. ACM Program. Lang.*, 4(POPL):63:1–63:28.

[Clairambault et al., 2019] Clairambault, P., de Visme, M., and Winskel, G. (2019). Game semantics for quantum programming. *Proc. ACM Program. Lang.*, 3(POPL):32:1–32:29.

[Clairambault and Forest, 2023] Clairambault, P. and Forest, S. (2023). The cartesian closed bicategory of thin spans of groupoids. In *LICS*, pages 1–13.

[Clairambault et al., 2012] Clairambault, P., Gutierrez, J., and Winskel, G. (2012). The winning ways of concurrent games. In *LICS*, pages 235–244. IEEE Computer Society.

[Clairambault and Harmer, 2010] Clairambault, P. and Harmer, R. (2010). Totality in arena games. *Ann. Pure Appl. Logic*, 161(5):673–689.

[Clairambault et al., 2023a] Clairambault, P., Olimpieri, F., and Paquet, H. (2023a). From thin concurrent games to generalized species of structure. In *Proceedings of the 38th Annual ACM/IEEE Symposium on Logic in Computer Science, LICS 2023, Boston, US, June 26–29, 2023*. To appear.

[Clairambault et al., 2023b] Clairambault, P., Olimpieri, F., and Paquet, H. (2023b). From thin concurrent games to generalized species of structures. In *LICS*, pages 1–14.

[Clairambault and Paquet, 2021] Clairambault, P. and Paquet, H. (2021). The quantitative collapse of concurrent games with symmetry. *CoRR*, abs/2107.03155.

[Coquand, 1995] Coquand, T. (1995). A semantics of evidence for classical arithmetic. *J. Symb. Log.*, 60(1):325–337.

[Curien and Faggian, 2005] Curien, P. and Faggian, C. (2005). L-nets, strategies and proof-nets. In *CSL*, volume 3634 of *Lecture Notes in Computer Science*, pages 167–183. Springer.

[Curien et al., 2016] Curien, P., Fiore, M. P., and Munch-Maccagnoni, G. (2016). A theory of effects and resources: adjunction models and polarised calculi. In *POPL*, pages 44–56. ACM.

[Dal Lago and Leventis, 2019] Dal Lago, U. and Leventis, T. (2019). On the taylor expansion of probabilistic lambda-terms. In Geuvers, H., editor, *4th International Conference on Formal Structures for Computation and Deduction, FSCD 2019, June 24-30, 2019, Dortmund, Germany*, volume 131 of *LIPIcs*, pages 13:1–13:16. Schloss Dagstuhl - Leibniz-Zentrum für Informatik.

[Danos and Ehrhard, 2011] Danos, V. and Ehrhard, T. (2011). Probabilistic coherence spaces as a model of higher-order probabilistic computation. *Inf. Comput.*, 209(6):966–991.

[Danos and Harmer, 2000] Danos, V. and Harmer, R. (2000). Probabilistic game semantics. In *LICS*, pages 204–213. IEEE Computer Society.

[Danos et al., 1996] Danos, V., Herbelin, H., and Regnier, L. (1996). Game semantics & abstract machines. In *Proceedings, 11th Annual IEEE Symposium on Logic in Computer Science, New Brunswick, New Jersey, USA, July 27-30, 1996*, pages 394–405. IEEE Computer Society.

[Danos and Regnier, 1996] Danos, V. and Regnier, L. (1996). Reversible, irreversible and optimal lambda-machines. In *Linear Logic Tokyo Meeting*, volume 3 of *Electronic Notes in Theoretical Computer Science*, pages 40–60. Elsevier.

[de Carvalho and de Falco, 2012] de Carvalho, D. and de Falco, L. T. (2012). The relational model is injective for multiplicative exponential linear logic (without weakenings). *Ann. Pure Appl. Log.*, 163(9):1210–1236.

[de Visme, 2019] de Visme, M. (2019). Event structures for mixed choice. In Fokkink, W. J. and van Glabbeek, R., editors, *30th International Conference on Concurrency Theory, CONCUR 2019, August 27-30, 2019, Amsterdam, the Netherlands*, volume 140 of *LIPIcs*, pages 11:1–11:16. Schloss Dagstuhl - Leibniz-Zentrum für Informatik.

[Dixon et al., 2021a] Dixon, A., Lazic, R., Murawski, A. S., and Walukiewicz, I. (2021a). Leafy automata for higher-order concurrency. In *FoSSaCS*, volume 12650 of *Lecture Notes in Computer Science*, pages 184–204. Springer.

[Dixon et al., 2021b] Dixon, A., Lazic, R., Murawski, A. S., and Walukiewicz, I. (2021b). Verifying higher-order concurrency with data automata. In *LICS*, pages 1–13. IEEE.

[Eberhart and Hirschowitz, 2018] Eberhart, C. and Hirschowitz, T. (2018). What's in a game?: A theory of game models. In Dawar, A. and Grädel, E., editors, *Proceedings of the 33rd Annual ACM/IEEE Symposium on Logic in Computer Science, LICS 2018, Oxford, UK, July 09-12, 2018*, pages 374–383. ACM.

[Eberhart et al., 2017] Eberhart, C., Hirschowitz, T., and Seiller, T. (2017). An intensionally fully-abstract sheaf model for π(expanded version). *Log. Methods Comput. Sci.*, 13(4).

[Ehrhard, 2012] Ehrhard, T. (2012). The scott model of linear logic is the extensional collapse of its relational model. *Theor. Comput. Sci.*, 424:20–45.

[Ehrhard and Guerrieri, 2016] Ehrhard, T. and Guerrieri, G. (2016). The bang calculus: an untyped lambda-calculus generalizing call-by-name and call-by-value. In *PPDP*, pages 174–187. ACM.

[Ehrhard et al., 2018] Ehrhard, T., Pagani, M., and Tasson, C. (2018). Full abstraction for probabilistic PCF. *J. ACM*, 65(4):23:1–23:44.

[Ehrhard and Regnier, 2003] Ehrhard, T. and Regnier, L. (2003). The differential lambda-calculus. *Theor. Comput. Sci.*, 309(1-3):1–41.

[Ehrhard and Regnier, 2008] Ehrhard, T. and Regnier, L. (2008). Uniformity and the taylor expansion of ordinary lambda-terms. *Theor. Comput. Sci.*, 403(2-3):347–372.

[Faggian and Hyland, 2002] Faggian, C. and Hyland, M. (2002). Designs, disputes and strategies. In *CSL*, volume 2471 of *Lecture Notes in Computer Science*, pages 442–457. Springer.

[Faggian and Maurel, 2005] Faggian, C. and Maurel, F. (2005). Ludics nets, a game model of concurrent interaction. In *LICS*, pages 376–385. IEEE Computer Society.

[Faggian and Piccolo, 2009] Faggian, C. and Piccolo, M. (2009). Partial orders, event structures and linear strategies. In *TLCA*, volume 5608 of *Lecture Notes in Computer Science*, pages 95–111. Springer.

[Fiore et al., 2008] Fiore, M., Gambino, N., Hyland, M., and Winskel, G. (2008). The cartesian closed bicategory of generalised species of structures. *Journal of the London Mathematical Society*, 77(1):203–220.

[Gabbay and Ghica, 2012] Gabbay, M. and Ghica, D. R. (2012). Game semantics in the nominal model. In *MFPS*, volume 286 of *Electronic Notes in Theoretical Computer Science*, pages 173–189. Elsevier.

[Ghica, 2005] Ghica, D. R. (2005). Slot games: a quantitative model of computation. In Palsberg, J. and Abadi, M., editors, *Proceedings of the 32nd ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages, POPL 2005, Long Beach, California, USA, January 12-14, 2005*, pages 85–97. ACM.

[Ghica and McCusker, 2003] Ghica, D. R. and McCusker, G. (2003). The regular-language semantics of second-order idealized Algol. *Theor. Comput. Sci.*, 309(1-3):469–502.

[Ghica and Murawski, 2004] Ghica, D. R. and Murawski, A. S. (2004). Angelic semantics of fine-grained concurrency. In *FoSSaCS*, volume 2987 of *Lecture Notes in Computer Science*, pages 211–225. Springer.

[Ghica and Murawski, 2008] Ghica, D. R. and Murawski, A. S. (2008). Angelic semantics of fine-grained concurrency. *Ann. Pure Appl. Log.*, 151(2-3):89–114.

[Ghica et al., 2006] Ghica, D. R., Murawski, A. S., and Ong, C. L. (2006). Syntactic control of concurrency. *Theor. Comput. Sci.*, 350(2-3):234–251.

[Ghica and Tzevelekos, 2012] Ghica, D. R. and Tzevelekos, N. (2012). A system-level game semantics. In Berger, U. and Mislove, M. W., editors, *Proceedings of the 28th*

Conference on the Mathematical Foundations of Programming Semantics, MFPS 2012, Bath, UK, June 6-9, 2012, volume 286 of *Electronic Notes in Theoretical Computer Science*, pages 191–211. Elsevier.

[Girard, 1987] Girard, J. (1987). Linear logic. *Theor. Comput. Sci.*, 50:1–102.

[Girard, 1988] Girard, J. (1988). Normal functors, power series and λ-calculus. *Ann. Pure Appl. Log.*, 37(2):129–177.

[Girard, 2001] Girard, J. (2001). Locus solum: From the rules of logic to the logic of rules. *Math. Struct. Comput. Sci.*, 11(3):301–506.

[Girard, 1989] Girard, J.-Y. (1989). Geometry of Interaction 1: Interpretation of System F. In *Studies in Logic and the Foundations of Mathematics*, volume 127, pages 221–260. Elsevier.

[Goyet, 2013] Goyet, A. (2013). The lambda lambda-bar calculus: a dual calculus for unconstrained strategies. In *POPL*, pages 155–166. ACM.

[Grellois and Melliès, 2015] Grellois, C. and Melliès, P. (2015). An infinitary model of linear logic. In Pitts, A. M., editor, *Foundations of Software Science and Computation Structures - 18th International Conference, FoSSaCS 2015, Held as Part of the European Joint Conferences on Theory and Practice of Software, ETAPS 2015, London, UK, April 11-18, 2015. Proceedings*, volume 9034 of *Lecture Notes in Computer Science*, pages 41–55. Springer.

[Griffin, 1990] Griffin, T. (1990). A formulae-as-types notion of control. In Allen, F. E., editor, *Conference Record of the Seventeenth Annual ACM Symposium on Principles of Programming Languages, San Francisco, California, USA, January 1990*, pages 47–58. ACM Press.

[Harmer, 1999] Harmer, R. (1999). *Games and full abstraction for non-deterministic languages*. PhD thesis, Imperial College London, UK.

[Harmer, 2004] Harmer, R. (2004). Innocent game semantics. *Lecture notes*, 2007.

[Harmer et al., 2007] Harmer, R., Hyland, M., and Melliès, P. (2007). Categorical combinatorics for innocent strategies. In *LICS*, pages 379–388. IEEE Computer Society.

[Harmer and McCusker, 1999] Harmer, R. and McCusker, G. (1999). A fully abstract game semantics for finite nondeterminism. In *14th Annual IEEE Symposium on Logic in Computer Science, Trento, Italy, July 2-5, 1999*, pages 422–430. IEEE Computer Society.

[Honda and Yoshida, 1999] Honda, K. and Yoshida, N. (1999). Game-theoretic analysis of call-by-value computation. *Theor. Comput. Sci.*, 221(1-2):393–456.

[Hyland and Ong, 2000] Hyland, J. M. E. and Ong, C. L. (2000). On full abstraction for PCF: I, II, and III. *Inf. Comput.*, 163(2):285–408.

[Hyland, 1997] Hyland, M. (1997). Game semantics. *Semantics and Logics of Computation*, 14:131.

[Hyland and Schalk, 2002] Hyland, M. and Schalk, A. (2002). Games on graphs and sequentially realizable functionals. In *17th IEEE Symposium on Logic in Computer Science (LICS 2002), 22-25 July 2002, Copenhagen, Denmark, Proceedings*, pages 257–264. IEEE Computer Society.

[Jaber, 2015] Jaber, G. (2015). Operational nominal game semantics. In Pitts, A. M., editor, *Foundations of Software Science and Computation Structures - 18th International Conference, FoSSaCS 2015, Held as Part of the European Joint Conferences on Theory and Practice of Software, ETAPS 2015, London, UK, April 11-18, 2015. Proceedings*, volume 9034 of *Lecture Notes in Computer Science*, pages 264–278. Springer.

[Jacq and Melliès, 2018] Jacq, C. and Melliès, P. (2018). Categorical combinatorics for non deterministic strategies on simple games. In Baier, C. and Lago, U. D., editors, *Foundations of Software Science and Computation Structures - 21st International Conference, FOSSACS 2018, Held as Part of the European Joint Conferences on Theory and Practice of Software, ETAPS 2018, Thessaloniki, Greece, April 14-20, 2018, Proceedings*, volume 10803 of *Lecture Notes in Computer Science*, pages 39–70. Springer.

[Joyal, 1977] Joyal, A. (1977). Remarques sur la théorie des jeux à deux personnes. *Gazette des sciences mathématiques du Québec*, 1(4):46–52.

[Kelly and Laplaza, 1980] Kelly, G. M. and Laplaza, M. L. (1980). Coherence for compact closed categories. *Journal of pure and applied algebra*, 19:193–213.

[Ker et al., 2002] Ker, A. D., Nickau, H., and Ong, C. L. (2002). Innocent game models of untyped lambda-calculus. *Theor. Comput. Sci.*, 272(1-2):247–292.

[Kerinec et al., 2020] Kerinec, A., Manzonetto, G., and Pagani, M. (2020). Revisiting call-by-value böhm trees in light of their taylor expansion. *Log. Methods Comput. Sci.*, 16(3).

[Koenig and Shao, 2020] Koenig, J. and Shao, Z. (2020). Refinement-based game semantics for certified abstraction layers. In *LICS*, pages 633–647. ACM.

[Krebbers et al., 2014] Krebbers, R., Leroy, X., and Wiedijk, F. (2014). Formal C semantics: Compcert and the C standard. In Klein, G. and Gamboa, R., editors, *Interactive Theorem Proving - 5th International Conference, ITP 2014, Held as Part of the Vienna Summer of Logic, VSL 2014, Vienna, Austria, July 14-17, 2014. Proceedings*, volume 8558 of *Lecture Notes in Computer Science*, pages 543–548. Springer.

[Laird, 1997] Laird, J. (1997). Full abstraction for functional languages with control. In *LICS*, pages 58–67. IEEE Computer Society.

[Laird, 2001a] Laird, J. (2001a). A fully abstract game semantics of local exceptions. In *LICS*, pages 105–114. IEEE Computer Society.

[Laird, 2001b] Laird, J. (2001b). A game semantics of idealized CSP. In *MFPS*, volume 45 of *Electronic Notes in Theoretical Computer Science*, pages 232–257. Elsevier.

[Laird, 2002] Laird, J. (2002). A categorical semantics of higher order store. In *CTCS*, volume 69 of *Electronic Notes in Theoretical Computer Science*, pages 209–226. Elsevier.

[Laird, 2007] Laird, J. (2007). A fully abstract trace semantics for general references. In *ICALP*, volume 4596 of *Lecture Notes in Computer Science*, pages 667–679. Springer.

[Laird et al., 2013] Laird, J., Manzonetto, G., McCusker, G., and Pagani, M. (2013). Weighted relational models of typed lambda-calculi. In *LICS*, pages 301–310. IEEE Computer Society.

[Lambek and Scott, 1988] Lambek, J. and Scott, P. J. (1988). *Introduction to higher-order categorical logic*, volume 7. Cambridge University Press.

[Laurent, 2002] Laurent, O. (2002). Polarized games. In *LICS*, page 265. IEEE Computer Society.

[Leroy, 2009] Leroy, X. (2009). Formal verification of a realistic compiler. *Commun. ACM*, 52(7):107–115.

[Levy, 1999] Levy, P. B. (1999). Call-by-push-value: A subsuming paradigm. In *TLCA*, volume 1581 of *Lecture Notes in Computer Science*, pages 228–242. Springer.

[Levy, 2014] Levy, P. B. (2014). Transition systems over games. Talk at the *Chocola* seminar.

[Levy and Staton, 2014] Levy, P. B. and Staton, S. (2014). Transition systems over games. In *Joint Meeting of the Twenty-Third EACSL Annual Conference on Computer Science Logic (CSL) and the Twenty-Ninth Annual ACM/IEEE Symposium on Logic in Computer Science (LICS), CSL-LICS '14, Vienna, Austria, July 14 - 18, 2014*, pages 64:1–64:10.

[Loader, 2001] Loader, R. (2001). Finitary PCF is not decidable. *Theor. Comput. Sci.*, 266(1-2):341–364.

[Mackie, 1995] Mackie, I. (1995). The geometry of interaction machine. In *POPL*, pages 198–208. ACM Press.

[McCusker, 2003] McCusker, G. (2003). On the semantics of the bad-variable constructor in algol-like languages. In *MFPS*, volume 83 of *Electronic Notes in Theoretical Computer Science*, pages 169–186. Elsevier.

[Melliès, 2004a] Melliès, P. (2004a). Asynchronous games 2: The true concurrency of innocence. In *CONCUR*, volume 3170 of *Lecture Notes in Computer Science*, pages 448–465. Springer.

[Melliès, 2004b] Melliès, P. (2004b). Asynchronous games 3 an innocent model of linear logic. In *CTCS*, volume 122 of *Electronic Notes in Theoretical Computer Science*, pages 171–192. Elsevier.

[Melliès, 2005] Melliès, P. (2005). Asynchronous games 4: A fully complete model of propositional linear logic. In *LICS*, pages 386–395. IEEE Computer Society.

[Melliès, 2006] Melliès, P. (2006). Asynchronous games 2: The true concurrency of innocence. *Theor. Comput. Sci.*, 358(2-3):200–228.

[Melliès, 2019a] Melliès, P. (2019a). Categorical combinatorics of scheduling and synchronization in game semantics. *Proc. ACM Program. Lang.*, 3(POPL):23:1–23:30.

[Melliès, 2019b] Melliès, P. (2019b). Template games and differential linear logic. In *34th Annual ACM/IEEE Symposium on Logic in Computer Science, LICS 2019, Vancouver, BC, Canada, June 24-27, 2019*, pages 1–13. IEEE.

[Melliès, 2021] Melliès, P. (2021). Asynchronous template games and the gray tensor product of 2-categories. In *36th Annual ACM/IEEE Symposium on Logic in Computer Science, LICS 2021, Rome, Italy, June 29 - July 2, 2021*, pages 1–13. IEEE.

[Melliès and Mimram, 2007] Melliès, P. and Mimram, S. (2007). Asynchronous games: Innocence without alternation. In *CONCUR*, volume 4703 of *Lecture Notes in Computer Science*, pages 395–411. Springer.

[Melliès and Tabareau, 2007] Melliès, P. and Tabareau, N. (2007). Resource modalities in game semantics. In *LICS*, pages 389–398. IEEE Computer Society.

[Melliès, 2003] Melliès, P.-A. (2003). Asynchronous games 1: Uniformity by group invariance.

[Melliès, 2009] Melliès, P.-A. (2009). Categorical semantics of linear logic. *Panoramas et synthèses*, 27:15–215.

[Miller, 1987] Miller, D. A. (1987). A compact representation of proofs. *Stud Logica*, 46(4):347–370.

[Mimram, 2008] Mimram, S. (2008). *Sémantique des jeux asynchrones et réécriture 2-dimensionnelle. (Asynchronous Game Semantics and 2-dimensional Rewriting Systems)*. PhD thesis, Paris Diderot University, France.

[Moggi, 1991] Moggi, E. (1991). Notions of computation and monads. *Inf. Comput.*, 93(1):55–92.

[Murawski, 2007] Murawski, A. S. (2007). Bad variables under control. In *CSL*, volume 4646 of *Lecture Notes in Computer Science*, pages 558–572. Springer.

[Murawski, 2010] Murawski, A. S. (2010). Full abstraction without synchronization primitives. In *MFPS*, volume 265 of *Electronic Notes in Theoretical Computer Science*, pages 423–436. Elsevier.

[Murawski and Tzevelekos, 2013] Murawski, A. S. and Tzevelekos, N. (2013). Deconstructing general references via game semantics. In *FoSSaCS*, volume 7794 of *Lecture Notes in Computer Science*, pages 241–256. Springer.

[New et al., 2016] New, M. S., Bowman, W. J., and Ahmed, A. (2016). Fully abstract compilation via universal embedding. In Garrigue, J., Keller, G., and Sumii, E., editors, *Proceedings of the 21st ACM SIGPLAN International Conference on Functional Programming, ICFP 2016, Nara, Japan, September 18-22, 2016*, pages 103–116. ACM.

[Nickau, 1994] Nickau, H. (1994). Hereditarily sequential functionals. In Nerode, A. and Matiyasevich, Y. V., editors, *Proceedings of the Symposium on Logical Foundations of Computer Science: Logic at St. Petersburg*, volume 813 of *Lecture Notes in Computer Science*, pages 253–264. Springer Verlag.

[O'Hearn, 2015] O'Hearn, P. W. (2015). From categorical logic to facebook engineering. In *LICS*, pages 17–20. IEEE Computer Society.

[Olimpieri, 2021] Olimpieri, F. (2021). Intersection type distributors. In *36th Annual ACM/IEEE Symposium on Logic in Computer Science, LICS 2021, Rome, Italy, June 29 - July 2, 2021*, pages 1–15. IEEE.

[Olimpieri and Vaux Auclair, 2022] Olimpieri, F. and Vaux Auclair, L. (2022). On the taylor expansion of λ-terms and the groupoid structure of their rigid approximants. *Log. Methods Comput. Sci.*, 18(1).

[Pagani et al., 2014] Pagani, M., Selinger, P., and Valiron, B. (2014). Applying quantitative semantics to higher-order quantum computing. In *POPL*, pages 647–658. ACM.

[Paquet, 2020] Paquet, H. (2020). Probabilistic concurrent game semantics. Technical report, University of Cambridge, Computer Laboratory.

[Paquet, 2023] Paquet, H. (2023). Bi-invariance for uniform strategies on event structures. *Electronic Notes in Theoretical Informatics and Computer Science*, 1.

[Plotkin, 1977] Plotkin, G. D. (1977). LCF considered as a programming language. *Theor. Comput. Sci.*, 5(3):223–255.

[Plotkin, 1981] Plotkin, G. D. (1981). Post-graduate lecture notes in advanced domain theory (incorporating the\pisa notes"). *Dept. of Computer Science, Univ. of Edinburgh*.

[Rideau and Winskel, 2011] Rideau, S. and Winskel, G. (2011). Concurrent strategies. In *Proceedings of the 26th Annual IEEE Symposium on Logic in Computer Science, LICS 2011, June 21-24, 2011, Toronto, Ontario, Canada*, pages 409–418. IEEE Computer Society.

[Sakayori and Tsukada, 2017] Sakayori, K. and Tsukada, T. (2017). A truly concurrent game model of the asynchronous \pi -calculus. In *FoSSaCS*, volume 10203 of *Lecture Notes in Computer Science*, pages 389–406.

[Saunders-Evans and Winskel, 2006] Saunders-Evans, L. and Winskel, G. (2006). Event structure spans for nondeterministic dataflow. In *EXPRESS*, volume 175 of *Electronic Notes in Theoretical Computer Science*, pages 109–129. Elsevier.

[Scott and Strachey, 1971] Scott, D. S. and Strachey, C. (1971). *Toward a mathematical semantics for computer languages*, volume 1. Oxford University Computing Laboratory, Programming Research Group Oxford.

[Selinger, 1997] Selinger, P. (1997). First-order axioms for asynchrony. In Mazurkiewicz, A. W. and Winkowski, J., editors, *CONCUR '97: Concurrency The-*

ory, 8th International Conference, Warsaw, Poland, July 1-4, 1997, Proceedings, volume 1243 of *Lecture Notes in Computer Science*, pages 376–390. Springer.

[Stewart et al., 2015] Stewart, G., Beringer, L., Cuellar, S., and Appel, A. W. (2015). Compositional compcert. In *POPL*, pages 275–287. ACM.

[Tsukada and Ong, 2015] Tsukada, T. and Ong, C. L. (2015). Nondeterminism in game semantics via sheaves. In *30th Annual ACM/IEEE Symposium on Logic in Computer Science, LICS 2015, Kyoto, Japan, July 6-10, 2015*, pages 220–231. IEEE Computer Society.

[Tsukada and Ong, 2016] Tsukada, T. and Ong, C. L. (2016). Plays as resource terms via non-idempotent intersection types. In Grohe, M., Koskinen, E., and Shankar, N., editors, *Proceedings of the 31st Annual ACM/IEEE Symposium on Logic in Computer Science, LICS '16, New York, NY, USA, July 5-8, 2016*, pages 237–246. ACM.

[Turi and Plotkin, 1997] Turi, D. and Plotkin, G. D. (1997). Towards a mathematical operational semantics. In *LICS*, pages 280–291. IEEE Computer Society.

[Tzevelekos, 2009] Tzevelekos, N. (2009). Full abstraction for nominal general references. *Log. Methods Comput. Sci.*, 5(3).

[Vale et al., 2023] Vale, A. O., Shao, Z., and Chen, Y. (2023). A compositional theory of linearizability. *Proc. ACM Program. Lang.*, 7(POPL):1089–1120.

[Winskel, 1980] Winskel, G. (1980). *Events in computation*. PhD Thesis, Edinburgh University.

[Winskel, 1982] Winskel, G. (1982). Event structure semantics for CCS and related languages. In Nielsen, M. and Schmidt, E. M., editors, *Automata, Languages and Programming, 9th Colloquium, Aarhus, Denmark, July 12-16, 1982, Proceedings*, volume 140 of *Lecture Notes in Computer Science*, pages 561–576. Springer.

[Winskel, 1986] Winskel, G. (1986). Event structures. In Brauer, W., Reisig, W., and Rozenberg, G., editors, *Petri Nets: Central Models and Their Properties, Advances in Petri Nets 1986, Part II, Proceedings of an Advanced Course, Bad Honnef, Germany, 8-19 September 1986*, volume 255 of *Lecture Notes in Computer Science*, pages 325–392. Springer.

[Winskel, 1993] Winskel, G. (1993). *The formal semantics of programming languages: an introduction*. MIT press.

[Winskel, 2007] Winskel, G. (2007). Event structures with symmetry. *Electron. Notes Theor. Comput. Sci.*, 172:611–652.

[Winskel, 2013a] Winskel, G. (2013a). Distributed probabilistic and quantum strategies. In *MFPS*, volume 298 of *Electronic Notes in Theoretical Computer Science*, pages 403–425. Elsevier.

[Winskel, 2013b] Winskel, G. (2013b). Strategies as profunctors. In Pfenning, F., editor, *Foundations of Software Science and Computation Structures - 16th International Conference, FOSSACS 2013, Held as Part of the European Joint Conferences*

on Theory and Practice of Software, ETAPS 2013, Rome, Italy, March 16-24, 2013. Proceedings, volume 7794 of *Lecture Notes in Computer Science*, pages 418–433. Springer.

[Xia et al., 2020] Xia, L., Zakowski, Y., He, P., Hur, C., Malecha, G., Pierce, B. C., and Zdancewic, S. (2020). Interaction trees: representing recursive and impure programs in coq. *Proc. ACM Program. Lang.*, 4(POPL):51:1–51:32.

Part V

Appendices

Appendix A

Complements on Relative Seely Categories

Like many other works in denotational semantics, the developments appearing in this monograph target programming languages with a non-linear type system, whose models are certain cartesian closed categories. But as is also common in denotational semantics, our cartesian closed categories are constructed following a linear decomposition, from a model of intuitionistic linear logic. A standard categorical structure used for this purpose is that of a *Seely category* [Melliès, 2009]: it is a symmetric monoidal closed category $\mathcal{C}$ also equipped with a cartesian structure along with a comonad ! on $\mathcal{C}$ and

$$!A \otimes !B \cong !(A \& B)\,, \qquad\qquad 1 \cong !\top\,,$$

natural isos called the *Seely isomorphisms* satisfying a few coherence conditions.

It *is* possible to build a genuine Seely category of games and concurrent strategies, and this is done for instance in [Castellan and Clairambault, 2021]. But the construction is inelegant: as argued in Section 8.2, handling types such as $o \multimap (o \otimes o)$ with a tensor on the right hand side complicates the shape of games and weakens the connection with the relational model. So we shall ban such types and work not with Seely categories, but with *relative* Seely categories, defined and developed here[1].

In Section A.1, we first define relative Seely category, then construct the cartesian closed Kleisli category of a Seely category. Then, in Section A.2 we shall define functors between relative Seely categories, and show how they lift to the Kleisli category.

[1]More precisely, the monograph builds on *relative Seely $\sim$-category*, where every law holds up to an equivalence relation $\sim$, which is also preserved by all constructions on morphisms – but we omit it in this appendix as it is completely orthogonal to the development.

A.1 Definition and Kleisli Category

Definition. It seems natural to first restate the definition from Section 8.2.1:

Definition 8.2.1. *A **relative Seely category** is a symmetric monoidal category $(\mathcal{C}, \otimes, 1)$ equipped with a full subcategory $\mathcal{C}_s$ together with the following data and axioms:*

- *$\mathcal{C}_s$ has finite products $(\&, \top)$ preserved by the inclusion functor $J : \mathcal{C}_s \hookrightarrow \mathcal{C}$.*
- *For every $B \in \mathcal{C}$ there is a functor $B \multimap - : \mathcal{C}_s \to \mathcal{C}_s$, such that there is*

$$\Lambda(-) : \mathcal{C}(A \otimes B, S) \simeq \mathcal{C}(A, B \multimap S).$$

  a bijection natural in $A \in \mathcal{C}$ and $S \in \mathcal{C}_s$.

- *There is a J-relative comonad $! : \mathcal{C}_s \to \mathcal{C}$: we have, for every $S \in \mathcal{C}_s$, an object $!S \in \mathcal{C}$ and a **dereliction** morphism $\mathsf{der}_S : !S \to S$, and for every $\sigma : !S \to T$, a **promotion** $\sigma^! : !S \to !T$, subject to three axioms [Altenkirch et al., 2010]:*

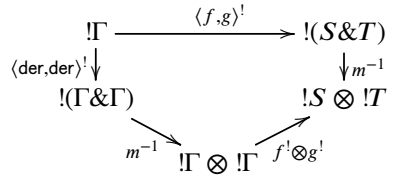
$$\begin{array}{rcll} \mathsf{der}_T \circ \sigma^! & = & \sigma & (\sigma : !S \to T) \\ \mathsf{der}_S^! & = & \mathrm{id}_{!S} & (S \in \mathcal{C}_s) \\ (\tau \circ \sigma^!)^! & = & \tau^! \circ \sigma^! & (\sigma : !S \to T, \tau : T \to U)\,, \end{array}$$

  which make $! : \mathcal{C}_s \to \mathcal{C}$ a functor, via $!\sigma = (\sigma \circ \mathsf{der}_S)^!$ for $\sigma : S \to T$.

- *The functor $! : \mathcal{C}_s \to \mathcal{C}$ is symmetric strong monoidal $(\mathcal{C}_s, \&, \top) \to (\mathcal{C}, \otimes, 1)$, so*

$$m_0 : 1 \to !\top \qquad\qquad m_{S,T} : !S \otimes !T \to !(S\&T)$$

  are natural isos, additional compatible with promotion: the diagram



  commutes for all $\Gamma, S, T \in \mathcal{C}_s$, $f \in \mathcal{C}(!\Gamma, S), g \in \mathcal{C}(!\Gamma, T)$.

As we shall annotate with ! the components of the constructed Kleisli category, we prefer to write the promotion as $\sigma^\dagger$ from now on (rather than $\sigma^!$), to avoid the collision.

Kleisli category. We define the Kleisli category $\mathcal{C}_!$ as follows. Its *objects* are simply those of $\mathcal{C}_s$. A *morphism from S to T* is $f : !S \to T$ in $\mathcal{C}$. The *identity* on object S is $\mathrm{id}_S^\dagger = \mathsf{der}_S : !S \to S$. The *composition* of $\sigma \in \mathcal{C}_!(S,T)$ and $\tau \in \mathcal{C}_!(T,U)$ is

$$\tau \circ^! \sigma \quad = \quad \tau \circ \sigma^\dagger\,.$$

It is completely straightforward that this data defines a category; the equations to be verified directly boil down to the axioms of relative comonads above. Thus we have:

Proposition A.1.1. *The data above forms a category $\mathcal{C}_!$, the **Kleisli category** of !.*

Cartesian closed structure. Obviously, $\top$ is still terminal in $\mathcal{C}_!$. If $S, T \in \mathcal{C}_s$, their *product* is $S\&T \in \mathcal{C}_s$. The corresponding *projections* are (in $\mathcal{C}$):

$$\begin{array}{rcccccc} \pi^!_S & = & \pi_S \circ \mathsf{der}_{S\&T} & : & !(S\&T) & \to & S \\ \pi^!_T & = & \pi_T \circ \mathsf{der}_{S\&T} & : & !(S\&T) & \to & T\,, \end{array}$$

and the *pairing* of $\sigma : !\Gamma \to S$ and $\tau : !\Gamma \to T$ is $\langle \sigma, \tau \rangle_! = \langle \sigma, \tau \rangle$. It is easy to check:

Proposition A.1.2. *This data equips $\mathcal{C}_!$ with a cartesian structure.*

Likewise, given $S, T \in \mathcal{C}_s$, their *arrow* is $S \Rightarrow T = !S \multimap T$. The *evaluation* is

$$\mathsf{ev}^{\Rightarrow}_{S,T} \quad = \quad !((!S \multimap T)\&S) \overset{m^{-1}_{!S\to T,S}}{\longrightarrow} !(!S \multimap T) \otimes !S \overset{\mathsf{der}_{!S\multimap T}\otimes !S}{\longrightarrow} (!S \multimap T) \otimes !S \overset{\mathsf{ev}_{!S,T}}{\longrightarrow} T$$

and given $\sigma : !(S\&T) \to U$, its *currying* is $\Lambda^!_{S,T,U} = \Lambda_{!S,!T,U}(\sigma \circ m_{S,T})$ which is in $\mathcal{C}_!(S, !T \multimap U)$ as required. Again, it is an elementary verification that:

Theorem A.1.3. *This data equips $\mathcal{C}_!$ with a cartesian closed structure.*

A.2 Relative Seely Functors

Theorem A.1.3 is our main tool to construct cartesian closed categories in this monograph; but we also need an accompanying tool to construct cartesian closed functors from an appropriate notion of functor between relative Seely categories.

Definition. Let us start by defining relative Seely functors.

Definition A.2.1. *Let $\mathcal{C}, \mathcal{D}$ be relative Seely categories. A **relative Seely functor** $\mathcal{C} \to \mathcal{D}$ is a functor $F : \mathcal{C} \to \mathcal{D}$ which restricts to $F : \mathcal{C}_s \to \mathcal{D}_s$, equipped with:*

- *For every $A, B \in \mathcal{C}$, natural isomorphisms morphisms*

$$\begin{array}{rcccc} t^{\otimes}_{A,B} & : & FA \otimes FB & \cong & F(A \otimes B) \\ t^{1} & : & 1 & \cong & F1; \end{array}$$

  making $(F, t^{\otimes}, t^{1})$ a strong symmetric monoidal functor $(\mathcal{C}, \otimes, 1) \to (\mathcal{D}, \otimes, 1)$;

- *for every $S, T \in \mathcal{C}_s$, isomorphisms*

$$\begin{array}{rcccc} t^{\&}_{S,T} & : & FS\&FT & \cong & F(S\&T) \\ t^{\top} & : & \top & \cong & F\top; \end{array}$$

- *For every $A \in \mathcal{C}$ and $S \in \mathcal{C}_s$, an isomorphism*

$$t^{\multimap}_{A,S} \quad : \quad FA \multimap FS \quad \cong \quad F(A \multimap S);$$

- *For every $S \in \mathcal{C}_s$, an isomorphism*

$$t^{!}_{S} \quad : \quad !FS \quad \cong \quad F\,!S;$$

satisfying the coherence axioms of Figure A.1, and the naturality-like condition that for every $S, T \in \mathcal{C}_s$ and $\sigma : !S \to T$, the following diagram commutes:

$$\begin{array}{ccc} !FS & \xrightarrow{(F\sigma \circ t^!_S)^\dagger} & !FT \\ {\scriptstyle t^!_S}\downarrow & & \downarrow{\scriptstyle t^!_T} \\ F\,!S & \xrightarrow[F(\sigma^\dagger)]{} & F\,!T \end{array}$$

Lifting to the Kleisli categories. Given a relative Seely functor $F : \mathcal{C} \to \mathcal{D}$, we define its **lifting** to a functor $F_! : \mathcal{C}_! \to \mathcal{D}_!$ that we shall prove to be cartesian closed.

On objects, we simply set $F_!(S) = FS$. On morphisms, for $\sigma : !S \to T$, we set

$$F_!\,\sigma = F\sigma \circ t^!_S : !FS \to FT\,.$$

From this data, a routine calculation ensures that we have:

Proposition A.2.2. *The above data yields a functor $F_! : \mathcal{C}_! \to \mathcal{D}_!$.*

Next, we show that this functor is cartesian. By definition, this means that $F\top$ is terminal – which is clear from $t^\top$ – and that for all $S, T \in \mathcal{C}_!$, the canonical map

$$\langle F_!\pi^!_S, F_!\pi^!_T\rangle_! \in \mathcal{D}_!(F(S\&T), FS\&FT)$$

obtained via the cartesian structures of $\mathcal{C}_!$ and $\mathcal{D}_!$, is invertible. Now using our coherence laws, this morphism simplifies to $\langle F\pi_S \circ \mathsf{der}_{S,T}, F\pi_T \circ \mathsf{der}_{S,T}\rangle : !F(S\&T) \to FS\&FT$. And indeed, the inverse we seek can be constructed as

$$r^\&_{S,T} \quad = \quad t^\&_{S,T} \circ \mathsf{der}_{FS\&FT} \quad : \quad !(FS\&FT) \quad \to \quad F(S\&T)\,;$$

that it is an inverse is a direct verification via our coherence laws, which ensure that $t^\&_{S,T}$ is an inverse in $\mathcal{D}$ to the canonical map $\langle F(\pi_S), F(\pi_T)\rangle : F(S\&T) \to FS\&FT$ given by the cartesian closed structures in $\mathcal{C}$ and $\mathcal{D}$.

Likewise, proving that $F_!$ is cartesian closed means showing, for all $S, T \in \mathcal{C}_!$, that

$$\Lambda^!(F_!(\mathsf{ev}^{\Rightarrow}_{S,T}) \circ {}^!r^{\Rightarrow}_{S,T}) \quad \in \quad \mathcal{C}_!(F(S \Rightarrow T), FS \Rightarrow FT)$$

obtained via the cartesian closed structures of $\mathcal{C}_!$ and $\mathcal{D}_!$, has an inverse. But via our coherence rules, this morphism simplifies to the composition in $\mathcal{D}$:

$$!F(!S \multimap T) \overset{\mathsf{der}_{F(!S\multimap T)}}{\longrightarrow} F(!S \multimap T) \overset{(t^{\multimap}_{!S,T})^{-1}}{\longrightarrow} F(!S) \multimap FT \overset{t^!_S \multimap FT}{\longrightarrow} !FS \multimap FT\,,$$

where we use that as usual, relative closure informs a contravariant functor $- \multimap S : \mathcal{D}^{\mathrm{op}} \to \mathcal{D}$ for all $S \in \mathcal{D}_s$, defined as $\sigma \multimap S = \Lambda(\mathsf{ev}_{B,S} \circ ((B \multimap S) \otimes \sigma)$ for any $\sigma : A \to B$ in $\mathcal{D}$. Now, the morphism above has an inverse in $\mathcal{D}_!$, constructed as

$$r^{\Rightarrow}_{S,T} \quad = \quad t^{\multimap}_{!S,T} \circ ((t^!_S)^{-1} \multimap FT) \circ \mathsf{der}_{FS\Rightarrow FT} \quad : \quad !(!FS \multimap FT) \quad \to \quad F(!S \multimap T)$$

concluding the proof of the following statement:

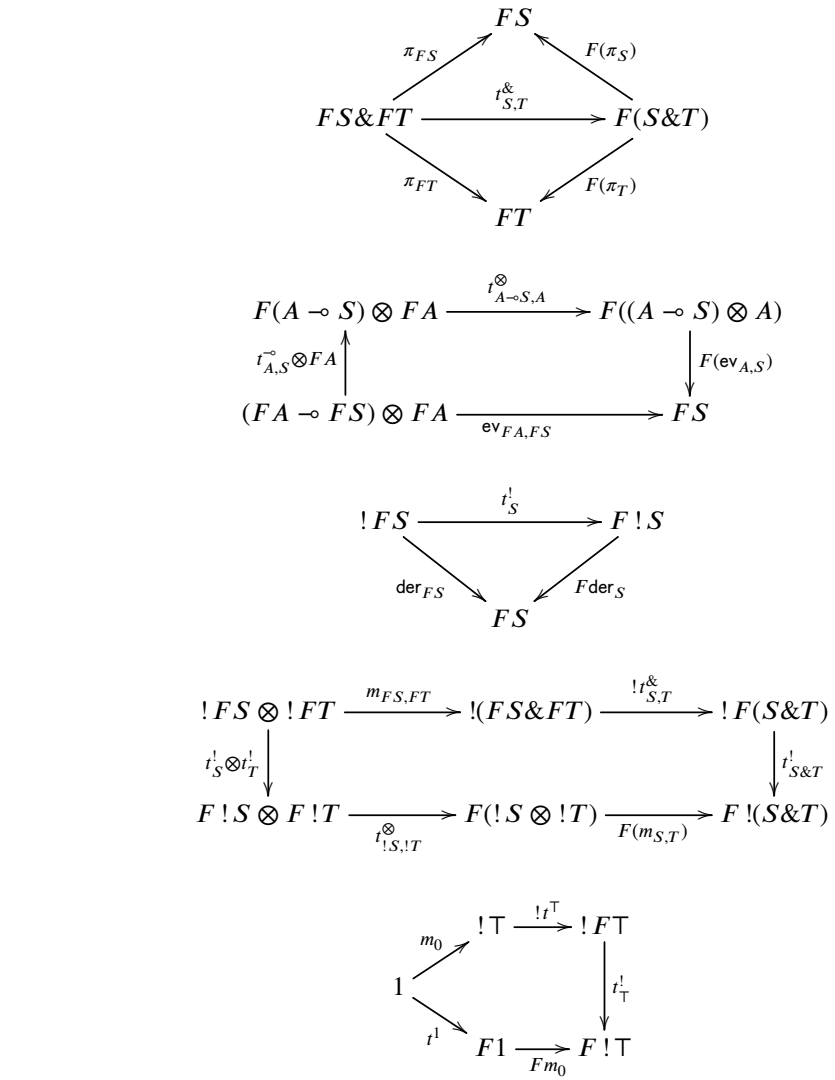


Figure A.1: Coherence diagrams for relative Seely functors

Theorem A.2.3. *Consider $F : \mathcal{C} \to \mathcal{D}$ a relative Seely functor.*
Then, the functor $F_! : \mathcal{C}_! \to \mathcal{D}_!$ constructed above is cartesian closed.

All the axioms of relative Seely functors get used in the proof, except for the last coherence diagram of Figure A.1, which appears unnecessary to ensure that a relative Seely functor lifts to a cartesian closed functor between the Kleisli categories. We include it nonetheless, as it seems likely to be necessary if we ever wish to show that relative Seely functors preserve the interpretation of a linear/non-linear type system.