



HAL
open science

Games and Strategies as Event Structures

Simon Castellan, Pierre Clairambault, Silvain Rideau, Glynn Winskel

► **To cite this version:**

Simon Castellan, Pierre Clairambault, Silvain Rideau, Glynn Winskel. Games and Strategies as Event Structures. Logical Methods in Computer Science, 2017, 10.23638/LMCS-13(3:35)2017 . hal-01302713v3

HAL Id: hal-01302713

<https://hal.science/hal-01302713v3>

Submitted on 16 May 2017

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.

GAMES AND STRATEGIES AS EVENT STRUCTURES

SIMON CASTELLAN, PIERRE CLAIRAMBAULT, SILVAIN RIDEAU, AND GLYNN WINSKEL

Univ Lyon, CNRS, ENS de Lyon, UCB Lyon 1, LIP
e-mail address: simon.castellan@ens-lyon.fr

Univ Lyon, CNRS, ENS de Lyon, UCB Lyon 1, LIP
e-mail address: pierre.clairambault@ens-lyon.fr

Department of Mathematics, UC Berkeley
e-mail address: silvain.rideau@berkeley.edu

Computer Laboratory, University of Cambridge
e-mail address: glynn.winskel@cl.cam.ac.uk

ABSTRACT. In 2011, Rideau and Winskel introduced *concurrent games and strategies as event structures*, generalizing prior work on causal formulations of games. In this paper we give a detailed, self-contained and slightly-updated account of the results of Rideau and Winskel: a notion of pre-strategy based on event structures; a characterisation of those pre-strategies (deemed *strategies*) which are preserved by composition with a copycat strategy; and the construction of a bicategory of these strategies. Furthermore, we prove that the corresponding category has a compact closed structure, and hence forms the basis for the semantics of concurrent higher-order computation.

1. INTRODUCTION

Games are ubiquitous. They appear in many areas, such as economics, logic, and computer science. They provide a valuable language in which one can model situations where the evolution of a system is determined by the choices of several agents. The agents are players performing moves according to rules that model the situation at hand, and the evolution of the system follows from the sequence of moves reflecting the decisions of the players. The outcome of the game might be a payoff for each player, a successful refutation of a logical formula, a bug exposed in a program – or, in some instances, we might just be interested in the play itself as a description of the evolution of a system. Although games can in general involve many players, they often (as in this paper) focus on two players: Player (Proponent, Eloïse, Verifier, . . .) and Opponent (Vbélard, Spoiler, . . .) each one defending their interests while subject to attacks of the other. Two-player games are perfectly suited to the representation of *open systems*: one player plays for the system at hand, while the other is deemed *external* and plays for the environment. Player may be regarded as representing a *team* of players, whose interaction yields the system under study.

1998 ACM Subject Classification: F.3.2.

Key words and phrases: Games, event structures, concurrency.

In their traditional formulation, games are highly sequential: the behaviour of a game determines a tree of which the nodes are the positions and the branches describe the different choices available to a player. The interaction between the players results in the selection of a potentially infinite branch of the game tree. Most of the time, each position belongs to exactly one player and the other has to wait until a move is played. Often, the game also obeys the condition of *alternation* where players are additionally required to play in turns.

Despite this sequential nature, one also would like to use games to represent situations that are concurrent or distributed, *e.g.* several systems running in parallel, possibly with synchronizations or shared resources. Of course such concurrent applications of games exist, but it is worth pointing out that in the overwhelming majority of cases concurrency is represented indirectly via the interleaving, or linearization, of atomic actions of the participants. Rather than using a notion of game that does justice to the distributed nature of the system, a tree-based, inherently sequential representation is opted for, where a branch is a total ordering of the implicitly partially ordered evolution of the system. In other words, concurrency is modelled by removing alternation, but the basic tree-based understanding remains unquestioned. Of course, that representation has been useful and sufficiently accurate to a large extent, and a significant and successful body of work follows from this choice. But we believe nonetheless that a more precise *causal* representation is to be preferred. Our reasons and a further discussion on this point can be found in Section 2.

However, causal representations of concurrent processes have a richer structure than trees, and require more elaborate tools to be dealt with properly. It was not clear at first on what mathematical formalism one should rely on for this endeavour. The first causal foundations for concurrent games emerged in the late nineties in the game semantics community; due to Abramsky and Melliès [AM99], they were used to build a fully complete model of multiplicative additive linear logic (MALL). The idea was to switch from a tree to a *domain* of positions, and formulate (deterministic) strategies as closure operators on this domain. Later, Melliès and Mimram [MM07] connected this position-based approach to a more traditional play-based formulation in the framework of asynchronous games – in this setting (deterministic) strategies were manipulated as traditional sets of plays, but with closure properties ensuring an underlying causal order between moves. In parallel, Faggian and Piccolo [FP09] had developed a setting where the (deterministic) strategies were manipulated explicitly as partial orders, rather than the partial order being recovered *a posteriori*. Finally, in 2011 Rideau and Winskel [RW11] generalized all prior work by proposing a setting where (non-deterministic) strategies are described as event structures, thus benefiting from a body of work on event structure models for concurrency.

The present paper aims to be a detailed and self-contained introduction to this latter formulation of concurrent games: it covers details and extends the results of [RW11]. In Section 2, we start with a gentle introduction to the basic ideas behind the representation of concurrent processes as event structures, with an eye towards the application to games. In this setting, both games and “pre-strategies” playing on them are event structures, with a pre-strategy being essentially an event structure labelled by moves of the game. But pre-strategies, thought of as prototypical strategies for Player, are too expressive: they may impose unreasonable constraints on Opponent, and can behave in ways that are not consistent with their standing for interaction in an asynchronous distributed environment. As an answer to this, strategies are introduced in Section 3 as the pre-strategies that are preserved under composition with an *asynchronous forwarder*, formalized as a copycat strategy. This provides an adequately robust notion of strategy on an event structure, and

a non-deterministic generalization of the earlier notions of concurrent strategies mentioned above. We prove the main result of [RW11]: that strategies are exactly the pre-strategies obeying conditions called *receptivity* and *courtesy*. The paper [RW11] also constructed a *bicategory* of concurrent games and strategies between them, akin to Joyal’s category of Conway games [Joy77]. In Section 4, we give a detailed proof of that result. Finally in Section 5 we show that just as Joyal’s category, our category is compact closed and can provide a basis for games-based models of higher-order computation. In Section 6, we conclude.

1.0.1. *Other related work.* Many other notions of games for concurrency have appeared in the literature.

In the verification community, “concurrent games” [dAH00, dAHK07] refer to variations of Blackwell games [Mar98]: there is a tree (or a graph) of positions. The game is played in rounds: at each round, both players select their behaviour from a pool of possible actions. This selection is independent, and with no information on the other player’s choice. The next position is decided as a function of both player’s choices. In contrast to our setting, their focus is on enforcing the independence of the two players in each round, rather than describing a general concurrent computation. In particular, plays are still totally ordered. Games in event structures are closer to the games played in Zielonka automata [GGMW13], which could be unfolded to event structures. However, our focus is more on the unfoldings themselves, and on their compositional structure.

Through our focus on compositionality, we are very close to the notions of games for concurrency studied in the semantics community [Lai01, GM08]. Just as we do, they form categories of games and strategies where concurrent processes can be modelled. However, these models are based on interleavings rather than partial orders: rather than opting for a primitive representation of concurrency based on partial orders, they represent the execution of a concurrent process via the non-deterministic schedulings of its possible actions.

Finally, in a different direction, let us cite the “playgrounds” of Hirschowitz *et al* [HP12, Hir13], and the multi-token Geometry of Interaction of Dal Lago *et al* [LFHY14]. Both formalisms aim at providing a non interleaving-based representation of concurrent processes and of their execution. They should both relate to our approach, in the sense that from their settings one could extract an event structure, which is arguably more abstract and syntax-independent than the models used there.

2. EVENT STRUCTURES, GAMES AND PRE-STRATEGIES

In this section we introduce the basic notions underlying our development, from event structures to pre-strategies represented by them.

2.1. Events for concurrent and distributed systems.

2.1.1. *Causality and independence.* It is common to describe the evolution of a process or system by listing its *events*, *i.e.* the observable actions occurring through time. For instance, one could describe an interaction with a coffee vending machine as a sequence:

$$\mathbf{coin} \cdot \mathbf{coffee}$$

that we call a **trace**, where **coin** represents the action of inserting a coin in the machine, and **coffee** represents the action of getting a coffee. In fact, the input/output behaviour of the vending machine may be modelled by the set:

$$\text{Coffee} = \{\epsilon, \mathbf{coin}, \mathbf{coin} \cdot \mathbf{coffee}\}$$

where ϵ is the empty sequence (and with possibly more iterations of the interaction if one is not interested in a one-use coffee vending machine). Nearby the coffee machine, there is a tea machine modelled by:

$$\text{Tea} = \{\epsilon, \mathbf{coin}', \mathbf{coin}' \cdot \mathbf{tea}\}$$

where we use **coin'** to distinguish it from **coin**.

The two machines may be interacted with in parallel – one may for instance pay for a coffee, then, while waiting for the machine to deliver, also pay for a tea, and then obtain both. This behaviour may be represented as **coin** · **coin'** · **coffee** · **tea**. In fact, the system formed by both machines can be modelled as:

$$\begin{aligned} &\{\epsilon, \mathbf{coin}, \mathbf{coin} \cdot \mathbf{coin}', \mathbf{coin} \cdot \mathbf{coffee}, \mathbf{coin} \cdot \mathbf{coin}' \cdot \mathbf{coffee}, \mathbf{coin} \cdot \mathbf{coffee} \cdot \mathbf{coin}', \\ &\mathbf{coin} \cdot \mathbf{coin}' \cdot \mathbf{tea}, \mathbf{coin} \cdot \mathbf{coin}' \cdot \mathbf{coffee} \cdot \mathbf{tea}, \mathbf{coin} \cdot \mathbf{coin}' \cdot \mathbf{tea} \cdot \mathbf{coffee}, \\ &\mathbf{coin} \cdot \mathbf{coffee} \cdot \mathbf{coin}' \cdot \mathbf{tea}, \mathbf{coin}', \mathbf{coin}' \cdot \mathbf{coin}, \mathbf{coin}' \cdot \mathbf{tea}, \mathbf{coin}' \cdot \mathbf{coin} \cdot \mathbf{tea}, \\ &\mathbf{coin}' \cdot \mathbf{tea} \cdot \mathbf{coin}, \mathbf{coin}' \cdot \mathbf{coin} \cdot \mathbf{tea} \cdot \mathbf{coffee}, \mathbf{coin}' \cdot \mathbf{coin} \cdot \mathbf{coffee}, \\ &\mathbf{coin}' \cdot \mathbf{coin} \cdot \mathbf{coffee} \cdot \mathbf{tea}, \mathbf{coin}' \cdot \mathbf{tea} \cdot \mathbf{coin} \cdot \mathbf{coffee}\} \end{aligned}$$

This follows the so-called *interleaving-based* approach to modelling concurrent and parallel systems: that two independent processes interacted with in parallel should behave as the set of interleavings of the traces of the original processes. This approach proved powerful and versatile, and provides the basis for most developments on models of concurrency.

However, it suffers from some drawbacks. To cite two of them: (1) as should appear clearly in our example, this representation gets exponentially bigger than the original system – this is the so-called *state explosion problem*, which is a main challenge in interleaving-based model-checking of concurrent systems, (2) it is unreadable, and obfuscates the key information of which events *depend* on which events. Instead of the large set of traces above, one would like to manage with only the *partial order* generating it displayed in Figure 1, for which the set of traces above is the set of all linearizations. This idea is far from new:

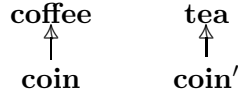


FIGURE 1. Partial order semantics for the coffee and tea machines

advocated first by Petri, it is known as the *independence*, *partial order*, *causal*, or *truly concurrent* approach to models of concurrency. Although causal models yield smaller and more intuitive representations of the dynamics of a concurrent process, they can be quite

subtle to manipulate and manage. Operations that are straightforward for interleaving-based approaches, and say based on simple induction, can be more mathematically involved when carried out in a partial-ordered setting.

2.1.2. *Event structures.* Our example above is purely deterministic: it appears visibly in the partial order of Figure 1 that no irreversible choice is ever made in the evolution of the system. Whatever order the events of a prefix of the partial order of Figure 1 appear, they can be completed to the maximal set $\{\mathbf{coin}, \mathbf{coffee}, \mathbf{coin}', \mathbf{tea}\}$. In this sense the order in which these events occur is irrelevant. To express non-determinism, one needs to enrich the partial order. A natural way to do that is to follow Winskel [Win86] and add a *consistency* relation on top of the partial order, as follows.

Definition 2.1 (Event structures). An **event structure** (**es** for short) is a tuple $(E, \leq_E, \text{Con}_E)$ where E is a set of events, \leq_E is a partial order on E called **causality** and Con_E is a non-empty set of finite subsets of E called **consistency**, such that:

$$\begin{aligned} \forall e \in E, [e] = \{e' \in E \mid e' \leq_E e\} \text{ is finite,} \\ \forall e \in E, \{e\} \in \text{Con}_E, \\ \forall X \in \text{Con}_E, \forall Y \subseteq X, Y \in \text{Con}_E, \\ \forall X \in \text{Con}_E, \forall e \in X, \forall e' \leq_E e, X \cup \{e'\} \in \text{Con}_E \end{aligned}$$

We will often omit the subscripts in \leq_E, Con_E if they are obvious from the context, and use E both for the event structure and its underlying set of events.

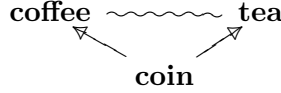
If $X \subseteq E$ is in Con , then we say that it is **consistent**, and its events may occur together. The states of an event structure E , called **configurations**, are the sets $x \subseteq E$ that are both consistent (in the sense that every finite subset belongs to Con_E) and **down-closed** (*i.e.* for all $e \in x$, for all $e' \leq e$, one has $e' \in x$). Here we shall work exclusively with finite configurations, those finite sets $x \subseteq E$ that are both consistent and down-closed; the set of such configurations of E is written $\mathcal{C}(E)$, and is partially ordered by inclusion. Configurations with a maximal element are called **prime configurations**, they are those of the form $[e]$ for $e \in E$. We will also use the notation $[e] = [e] \setminus \{e\}$. Between configurations, the **covering relation** $x \dashv\!\! \dashv y$ means that y is obtained from x by adding exactly one event: y is an **atomic extension** of x . We might also write $x \dashv\!\! \dashv^e$ to mean that $e \notin x$ and $x \cup \{e\} \in \mathcal{C}(E)$; this says that the event e is enabled at x . It is easy to prove that the relation \subseteq between configurations is the transitive reflexive closure of $\dashv\!\! \dashv$; in fact $\dashv\!\! \dashv$ is its transitive reduction.

When drawing event structures, we will not portray the full partial order \leq but its transitive reduction; the **immediate causality** generating it defined as $e \rightarrow e'$ whenever $e < e'$ and for any $e \leq e'' \leq e'$, either $e = e''$ or $e'' = e'$. Finally, we say two events e, e' are **concurrent** when they are consistent and incomparable for \leq_E .

Event structures can express non binary conflict, *e.g.* one can have three events $\{1, 2, 3\}$ with consistent subsets those with at most two elements: all events are pairwise consistent, but not the three of them together. This extra generality makes for a smooth theory, but in many examples consistency is equivalently described by a complementary irreflexive binary *conflict* relation \sharp , that relates any two events that *cannot* occur together, *i.e.* $X \in \text{Con}$ iff for all $e, e' \in X$, $\neg(e \sharp e')$. It follows then from the axioms of event structures that if $e \sharp e'$ and $e' \leq e''$, then $e \sharp e''$ as well – we call this conflict **inherited**. A conflict $e \sharp e'$ that is not

inherited is called **minimal**, and represented as $e \sim e'$. In order to alleviate the notation, when drawing event structures with binary conflict we only represent minimal conflicts.

As an example, consider a (less popular) variant of the coffee machine above: when a coin is inserted it will produce a tea or a coffee, nondeterministically. The corresponding event structure can be represented as follows:



Its configurations are $\{\{\emptyset\}, \{\text{coin}\}, \{\text{coin}, \text{coffee}\}, \{\text{coin}, \text{tea}\}\}$. We will never get *both* tea and coffee even though both are enabled by **coin**.

2.1.3. Simple parallel composition. Whereas using traces the operation of putting two systems in parallel without communication or interaction was the source of a combinatorial explosion, in event structures it only consists in putting two event structures side by side. For instance, the event structure of Figure 1 is obtained in a transparent way from event structures for the coffee and tea machines. Generally:

Definition 2.2. Given two event structures E and F their **simple parallel composition** (or just *parallel composition* for short) $E \parallel F$ is defined as the event structure comprising:

- *Events:* $\{0\} \times E \cup \{1\} \times F$ (tagged disjoint union of E and F),
- *Causality:* $(i, c) \leq_{E \parallel F} (j, c')$ when $i = j = 0$ and $c \leq_E c'$ or $i = j = 1$ and $c \leq_F c'$,
- *Consistency* defined as:

$$X \in \text{Con}_{E \parallel F} \text{ iff } \{a \mid (0, a) \in X\} \in \text{Con}_E \ \& \ \{b \mid (1, b) \in X\} \in \text{Con}_F$$

Thus, $E \parallel F$ is E and F put side-by-side with no causality or conflict between them. As a result, configurations of $E \parallel F$ can be easily described in terms of those of E and F – namely there is a canonical order-isomorphism $\mathcal{C}(E \parallel F) \cong \mathcal{C}(E) \times \mathcal{C}(F)$ (where configurations are ordered by inclusion). We will denote by $x \parallel y \in \mathcal{C}(E \parallel F)$ the configuration corresponding to $(x, y) \in \mathcal{C}(E) \times \mathcal{C}(F)$. When denoting events of a parallel composition $E_1 \parallel E_2$, we will not always write the explicit injections (as in $(0, e)$ or $(1, e)$). Instead, we will often annotate or name the events so as to disambiguate the components they belong to (as in *e.g.* e_1, e_2).

2.1.4. Conjunctive causality and projection. In this setting of event structures causality is *conjunctive* rather than *disjunctive*: states/configurations need to be down-closed, so for an event to occur it is required that *all* of its dependencies have occurred before. For instance, in the event structure of Figure 2 the user needs to *both* insert a coin and press a button in order to get a drink (inserting a coin and pressing both buttons results in a non-deterministic choice).

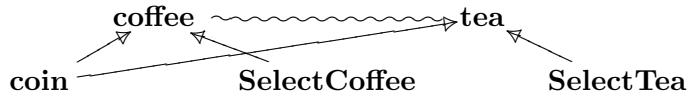


FIGURE 2. An event structure for a vending machine with selection

Plain event structures cannot express that an event may occur for two distinct, independent reasons – such as in saying that **coffee** can be obtained through a **coin** or through an override mechanism. In event structures, expressing that would require two distinct events **coffee** and **coffee'**, with different causal histories. The apparent limitation that each event has a unique, unambiguous causal history enables us to perform the following *projection* operation¹:

Definition 2.3. If E is an event structure and $V \subseteq E$ is a subset of events, then the **projection** $E \downarrow V$ has V as events, and causality and consistency directly inherited from V : if $e_1, e_2 \in V$ then $e_1 \leq_{E \downarrow V} e_2$ iff $e_1 \leq_E e_2$, and for X a finite subset of V , $X \in \text{Con}_{E \downarrow V}$ iff $X \in \text{Con}_E$.

In other words, the projection $E \downarrow V$ is obtained by considering the events not in V to be invisible: they occur silently, and are not observable anymore. Because causality is conjunctive, for an event $e \in E \downarrow V$ there is never any ambiguity as to what events caused it in E . Each configuration $y \in \mathcal{C}(E)$ projects to $y \cap V \in \mathcal{C}(E \downarrow V)$ – reciprocally, any $x \in \mathcal{C}(E \downarrow V)$ has a minimal **witness** $[x]_E = \{e' \in E \mid e' \leq_E e \in x\} \in \mathcal{C}(E)$, yielding a bijection:

$$\begin{array}{ccc} \mathcal{C}(E \downarrow V) & \cong & \{x \in \mathcal{C}(E) \mid \forall e \in x \text{ maximal, } e \in V\} \\ x & \mapsto & [x]_E \\ y \cap V & \leftarrow & y \end{array}$$

that preserves and reflects inclusion. This feature will be key to the *hiding* step of the composition of strategies, introduced later.

2.1.5. *Polarity and pre-strategies.* We now move towards games. We consider *two-player games* between Player (considered as having positive polarity) and Opponent (considered as having negative polarity). Each event is equipped with a polarity, indicating which player has the responsibility to play it.

Definition 2.4. An **event structure with polarities** (esp for short) is an event structure A along with a function

$$\text{pol}_A : A \rightarrow \{-, +\}$$

associating to each event a polarity.

When introducing events of an esp A , we might annotate them in order to indicate their polarity. For instance, in “let $a^- \in A$ ”, a ranges over all events of A of negative polarity. For configurations $x, y \in \mathcal{C}(A)$, we will write $x \subseteq^- y$ if $x \subseteq y$ and all events in $y \setminus x$ are negative; $x \subseteq^+ y$ is defined dually. For an esp A , we will write A^\perp for its **dual**, *i.e.* A with the same data, except for the polarity which is reversed.

We define **games** to be simply esps. The terms *game* and *esp* will however not be used interchangeably: we will use *games* for those esps used to specify the interface at which two players interact (*strategies* will also be certain esps). For instance, one could model the interface of the vending machine above by saying that Player plays according to the program of the coffee machine, Opponent plays for the user, and the game describes the observable actions through which they interact on the physical device. Following this idea, the game for the physical interface of the coffee machine would have events

¹General event structures [Win80] avoid this restriction, though they do not support a reasonable projection operation.

$\{\mathbf{coin}^-, \mathbf{SelectCoffee}^-, \mathbf{SelectTea}^-, \mathbf{coffee}^+, \mathbf{tea}^+\}$, for causality the discrete partial order (*i.e.* the order contains only the reflexive pairs), and all sets consistent. This game is a discrete partial order, but in general games can feature non-trivial causality and consistency.

The *strategy* for Player would then describe the behaviour of the vending machine at this interface, represented as an event structure as well (such as Figure 2). Both *games* and *strategies* are expressed as esps; they will nonetheless play very different roles in the development. Following this idea, we now define *pre-strategies* – *strategies*, defined later, will be subject to further conditions.

Definition 2.5. A **pre-strategy** on a game A is an esp S labelled by A , that is, a function $\sigma : S \rightarrow A$ which:

- (1) Obeys the rules of the game (preserves configurations):

$$\forall x \in \mathcal{C}(S), \sigma x \in \mathcal{C}(A)$$

- (2) Plays linearly (local injectivity):

$$\forall s, s' \in x \in \mathcal{C}(S), \sigma s = \sigma s' \implies s = s'$$

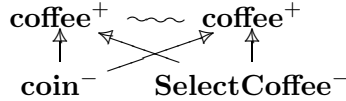
- (3) Preserves polarity:

$$\forall s \in S, \text{pol}_A(\sigma s) = \text{pol}_S(s)$$

Note that σ does not need to preserve the order. On the other hand, as we will see later (Lemma 2.13) it follows from these axioms that it always reflects it for consistent events.

As announced, a pre-strategy on A is an esp S along with a labelling function $\sigma : S \rightarrow A$. The esp structure on A brings constraints, that the labelling function has to respect. It is easy to check that the event structure of Figure 2 is a pre-strategy on the game $M = \{\mathbf{coin}^-, \mathbf{SelectCoffee}^-, \mathbf{SelectTea}^-, \mathbf{coffee}^+, \mathbf{tea}^+\}$ (with trivial causality and all subsets consistent), with the obvious map to it given by the labels. In the rest of this paper, when drawing pre-strategies we will follow the presentation of Figure 2: we will draw the event structure S , with events written as their image via σ .

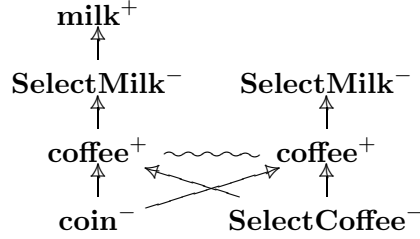
Remark 2.6. A pre-strategy $\sigma : S \rightarrow A$ is *locally injective* but may not be *injective*: there may well be several incompatible events in S mapping to the same event in A . This is true even for two events $s_1, s_2 \in S$ sharing the same causal history, *e.g.* $[s_1] = [s_2]$. For instance, the following variant of Figure 2 represents a valid pre-strategy on the game M above.



The intuition here is that the coffee machine secretly tosses a coin (it has plenty of those after all), but the result does not change its behaviour: it serves the coffee nonetheless. The intensional information of this nondeterministic choice, despite being unobservable, can be recorded by our semantics. This intensionality means that our model will not validate the idempotency law for nondeterministic choice: the event structures $e \rightsquigarrow e$ and e are not isomorphic.

Though it is not covered in this paper, one may opt for a variant of our setting that does validate idempotency (see *e.g.* [CC16]). But as a consequence we would also lose the branching information. To illustrate this, consider the following variant of the pre-strategy

above, playing on an extension of M popular in the United Kingdom.



In this version, the user has the possibility of requesting milk after the coffee is served. But in the pre-strategy above, they may not always get it: something may go wrong in the machine before the coffee is served, leading to a state where **SelectMilk**⁻ is ineffective and the user frustrated. The highly intensional, non-idempotent representation of concurrency we opt for in this paper allows us to record this branching information. In settings where this branching information is not relevant, such as in [CC16], it can be easily forgotten.

Besides being natural (we hope) as a first tentative definition of strategies, the pre-strategies defined above match the standard notion of *maps* between event structures.

Definition 2.7. If E, F are event structures, a (total) **map of event structures** from E to F is a function on events $f : E \rightarrow F$ satisfying (1) and (2) above.

The identity function on E is a map of event structures and those are stable under composition; in other words there is a category \mathcal{E} of event structures and maps.

Later on we will also consider *partial* maps between event structures (Definition 4.2), but throughout this paper all maps are considered total unless explicitly said otherwise. We also have a category \mathcal{EP} of esps, and maps *preserving polarities* – technically pre-strategies are exactly maps of esps. We keep a distinguished terminology, because in the sequel we will encounter maps of esps that it is unwise to regard as pre-strategies.

We note in passing that simple parallel composition extends to esps by defining the polarity of $A \parallel B$ as $\text{pol}_{A \parallel B}(0, a) = \text{pol}_A(a)$ and $\text{pol}_{A \parallel B}(1, b) = \text{pol}_B(b)$ – this entails that parallel composition commutes with the duality operation, *i.e.* $(A \parallel B)^\perp = A^\perp \parallel B^\perp$. Two pre-strategies $\sigma : S \rightarrow A$ and $\tau : T \rightarrow B$ playing respectively on A and B can be combined to form a pre-strategy $\sigma \parallel \tau : S \parallel T \rightarrow A \parallel B$ defined by $(\sigma \parallel \tau)(0, a) = (0, \sigma(a))$ and $(\sigma \parallel \tau)(1, b) = (1, \tau(b))$. In fact with this definition, simple parallel composition acts functorially on maps of es and esp and equip the categories \mathcal{E} and \mathcal{EP} with the structure of a symmetric monoidal category (with the empty event structure 1 as unit).

At this point, the reader may find confusing the fact that although there are polarities in games and pre-strategies, these are not taken into special account in the definition of pre-strategies (besides their preservation by the labeling function). This makes pre-strategies more powerful than perhaps wished: a pre-strategy may constrain the external Opponent beyond the rules of the game. Taking $S = 1$ the empty event structure, and writing \ominus for the esp with just one negative event, the empty map $S \rightarrow \ominus$ is a valid pre-strategy. As S has no counterpart for the unique negative move in the game, this pre-strategy fails to acknowledge Opponent's right to play it. This is in contradiction with the idea that Opponent's available actions should only depend on the game, and not be controllable by Player. We will see in Section 3 other ways in which pre-strategies may constrain Opponent in unintended ways.

This is because the current definition is an intermediate step, towards the notion of strategy introduced in Section 3 that will take polarity more carefully into account. Whereas pre-strategies axiomatize the polarity-agnostic description of the evolution of a concurrent process on an interface, strategies will satisfy polarity-specific constraints, *e.g.* a strategy cannot prevent its opponent from playing a move enabled in the game. But for the remainder of this section, polarities are present only to set the stage for Section 3.

2.2. Interaction of pre-strategies. Pre-strategies playing on A^\perp are pre-strategies for Opponent or *counter pre-strategies*. Given a pre-strategy $\sigma : S \rightarrow A$ and a counter pre-strategy $\tau : T \rightarrow A^\perp$, we proceed to explain how they *interact* with each other. As σ and τ have opposite expectations for the polarity of events in A , their interaction will drop all information about polarities, *i.e.* it will be a *map of event structures* (in \mathcal{E}):

$$\sigma \wedge \tau : S \wedge T \rightarrow A$$

where A is silently coerced to an event structure *without polarities*. We regard it again as an event structure $S \wedge T$ describing the causal structure of actions that both σ and τ agree to, along with a labeling $\sigma \wedge \tau$ of $S \wedge T$ by events of (a polarity-agnostic version of) A .

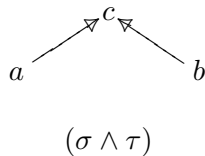
In fact, the *definition* of interaction makes use neither of the polarity information in S, T , or A , nor of its preservation by σ and τ . So we will define it in this section as an operation which for two maps $\sigma : S \rightarrow A$ and $\tau : T \rightarrow A$ in \mathcal{E} , yields a map $\sigma \wedge \tau : S \wedge T \rightarrow A$ in \mathcal{E} . Polarities will only become relevant again in Section 2.3, when we define composition.

As we will see, interaction is very close to the product of event structures used in [Win82, Win86] to interpret the synchronising parallel composition of CCS (we will see that it corresponds to a *pullback* in \mathcal{E}).

2.2.1. Secured bijections. The interaction of σ and τ should follow the behaviour that σ and τ agree on: in a given state, it should be ready to play $c \in A$ whenever σ and τ are. In particular, this means that an event $c \in A$ played by σ and τ should be played in their interaction only after all the dependencies in S and T are satisfied. For instance the interaction of the following two event structures labelled on the interface $A = a b c$ (consisting in three concurrent events)

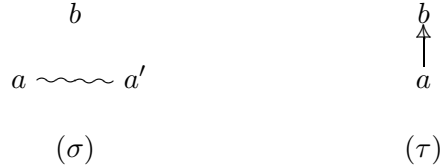


should give rise to the interaction $\sigma \wedge \tau$:

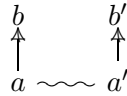


with immediate causal links imported from both S and T . Similarly, a set of events should be consistent in the interaction when the corresponding projections in S and T are.

At this point, one is tempted to define the events of $S \wedge T$ as *synchronized events*: pairs $(s, t) \in S \times T$ such that $\sigma s = \tau t$. This works correctly when the maps σ and τ are injective but fails in general. For instance, consider the interaction of the two labelled event structures:



Here, σ has two copies a and a' of the event $a \in A$ (by local injectivity, the two copies must be in conflict) and τ plays b after a . However, because σ has two ways of playing a , the interaction has two possible causal histories for b : either after $(a, a) \in S \times T$ or after $(a', a) \in S \times T$. Since in event structures, each event comes with a unique causal history, those two histories for b must correspond to *two different events* in $S \wedge T$, which should therefore look like:



We see that $S \wedge T$ has four events, whereas there are only three possible synchronized pairs: (a, a) , (a', a) and (b, b) – thus events of $S \wedge T$ will be more than just pairs.

Our approach will be to construct the desired event structure $S \wedge T$ indirectly via the set of configurations that we wish it to have. In the example above, configurations of the diagram *are* in one-to-one correspondence with synchronized configurations: pairs $(x, y) \in \mathcal{C}(S) \times \mathcal{C}(T)$ such that $\sigma x = \tau y$. By local injectivity, in such a situation σ and τ induce a bijection $\varphi : x \simeq \sigma x = \tau y \simeq y$ that is not order preserving in general (we use the notation \simeq , as opposed to \cong , to insist on the fact that although $x, \sigma x, \tau y$ and y are canonically partially ordered by \leq_S, \leq_A, \leq_T , these bijections do not preserve this order). Note that its graph is a set of synchronized (paired) events as above.

Such bijections will be used to represent *configurations* of the interaction. But as configurations of an event structure (yet to be defined), the graphs of these bijections should be ordered as well. As shown above, the order on $S \wedge T$ should be inherited from that of S and T . However, the transitive closure of the relation induced by the orders of S and T is, in general, not an order. For instance in the following picture



there is a *deadlock*: σ (the dealer) waits for the money to be delivered before presenting the drug while τ (the buyer) waits for the drug before offering the dollars. Their interaction should be empty as in the empty configuration there is no common event that σ and τ are both ready to play. This is reflected by the fact that on the bijection

$\{(\mathbf{Money}, \mathbf{Money}), (\mathbf{Drug}, \mathbf{Drug})\}$ the preorder induced by S and T is not an order: it has a loop. To eliminate such loops, we introduce *secured bijections*:

Definition 2.8 (Secured bijection). A **secured bijection** between two (finite) orders (q, \leq_q) and $(q', \leq_{q'})$ is a bijection $\varphi : q \simeq q'$ such that the reflexive and transitive closure of the following relation on the graph of φ is an order:

$$(a, b) \triangleleft (a', b') \quad \text{when} \quad a <_q a' \text{ or } b <_{q'} b'$$

Secured bijections need not preserve the order but they do not contradict it: if $a <_q b$ then $\varphi b \not<_{q'} \varphi a$ as otherwise this would constitute a cycle.

Equivalently, secured bijections are those bijections satisfying a reachability property akin to one of configurations of event structures, which can always be reached from the empty configuration by successive additions of events. We invite the reader to check the following lemma, which is useful in forging an intuition on the role of the notion.

Lemma 2.9. Let (q, \leq_q) and $(q', \leq_{q'})$. Then a bijection $\varphi : q \simeq q'$ is secured, iff there is a sequence of (graphs of) bijections:

$$(\varphi_0 : x_0 \simeq y_0) \xrightarrow{(a_1, b_1)} (\varphi_1 : x_1 \simeq y_1) \xrightarrow{(a_2, b_2)} \dots \xrightarrow{(a_n, b_n)} (\varphi_n : x_n \simeq y_n)$$

such that φ_0 is the empty bijection, $\varphi_n = \varphi$, and for all $0 \leq i \leq n$, $x_i \in \mathcal{C}(q)$ and $y_i \in \mathcal{C}(q')$ (*i.e.* they are down-closed).

Secured bijections can be used to give a very concise description of the desired states of $S \wedge T$: write $\mathcal{B}_{\sigma, \tau}^{\text{sec}}$ for the following set, ordered by inclusion.

$$\mathcal{B}_{\sigma, \tau}^{\text{sec}} = \{\varphi \mid \varphi : x \xrightarrow{\sigma} \sigma x = \tau y \xrightarrow{\tau} y \text{ is secured, with } x \in \mathcal{C}(S), y \in \mathcal{C}(T)\}.$$

Since secured bijections are by definition equipped with a canonical order, the elements of $\mathcal{B}_{\sigma, \tau}^{\text{sec}}$ can be seen as ordered sets. Immediate causal links in a secured bijection are related to those of the underlying orders:

Lemma 2.10. Let $\varphi : q \simeq q'$ be a secured bijection. If we have $(a, b) \rightarrow_{\varphi} (a', b')$ then either $a \rightarrow_q a'$ or $b \rightarrow_{q'} b'$.

Proof. From $(a, b) \rightarrow_{\varphi} (a', b')$ we deduce $(a, b) \triangleleft (a', b')$. Hence either $a <_q a'$ or $b <_{q'} b'$. Assume for instance $a <_q a'$. If we do not have $a \rightarrow_q a'$ then there exists $a_0 \in q$ such that $a < a_0 < a'$. Then $(a_0, \varphi a_0) \in \varphi$ and we have $(a, b) <_{\varphi} (a_0, \varphi a_0) <_{\varphi} (a', b')$ contradicting the hypothesis. \square

2.2.2. Prime secured bijections. The order $(\mathcal{B}_{\sigma, \tau}^{\text{sec}}, \subseteq)$ is (up to isomorphism) the order of configurations of the event structure we are looking for. We can now reconstruct an event structure whose order of configurations matches this order: events are identified as the *prime* secured bijections, *i.e.* those with a top (*i.e.* greatest) synchronized event (s, t) . In other words there will be an event for each synchronized pair (s, t) along with a consistent causal history for it, *i.e.* a prime secured bijection with (s, t) as top element. In particular, if there is none (because of a cycle), it would not appear in the interaction. With these ingredients we can form an event structure:

Definition 2.11 (Interaction of pre-strategies). Let $\sigma : S \rightarrow A$ and $\tau : T \rightarrow A$ be maps of event structures. We define the event structure $S \wedge T$ as follows:

- *Events*: those elements of $\mathcal{B}_{\sigma,\tau}^{\text{sec}}$ that have a top event,
- *Causality*: inclusion of graphs,
- *Consistency*: a finite set X of (graphs of) secured bijections is consistent when its union is still (the graph of) a secured bijection in $\mathcal{B}_{\sigma,\tau}^{\text{sec}}$.

We invite the reader to apply this definition on the examples at the beginning of Section 2.2.1, and check that we obtain the event structures announced.

It is routine to check that $S \wedge T$ is an event structure such that $\mathcal{C}(S \wedge T)$ is order-isomorphic to $\mathcal{B}_{\sigma,\tau}^{\text{sec}}$:

Lemma 2.12. For each configuration $x \in \mathcal{C}(S \wedge T)$, then $\varphi_x = \cup x : x_S \simeq x_T \in \mathcal{B}_{\sigma,\tau}^{\text{sec}}$ is a secured bijection. Moreover, this assignment is such that

$$\begin{aligned} \varphi_x &\rightarrow x \\ (s, t) &\mapsto [(s, t)]_{\varphi_x} \end{aligned}$$

is an order-isomorphism $\varphi_x \cong x$, where $[(s, t)]_{\varphi_x}$ denotes the down-closure of (s, t) inside the ordered set φ_x . Moreover, the mapping $x \mapsto \varphi_x$ defines an order isomorphism $\mathcal{C}(S \wedge T) \cong \mathcal{B}_{\sigma,\tau}^{\text{sec}}$.

Proof. Let $x \in \mathcal{C}(S \wedge T)$. By definition of consistency in $S \wedge T$, $\cup x$ is the graph of a secured bijection $\varphi_x \in \mathcal{B}_{\sigma,\tau}^{\text{sec}}$. Any element of x is a secured bijection with a maximal element (s, t) , and hence is $[(s, t)]_{\varphi_x}$. Thus, $[(s, t)]_{\varphi_x} \mapsto (s, t)$ defines an order-isomorphism $x \cong \varphi_x$. This yields a map $\mathcal{C}(S \wedge T) \rightarrow \mathcal{B}_{\sigma,\tau}^{\text{sec}}$. The converse maps a secured bijection φ to the set of elements of $S \wedge T$ included in φ . \square

By local injectivity of σ and τ , a secured bijection $\varphi : x \simeq y$ is entirely determined by x and y . Therefore, $\mathcal{C}(S \wedge T)$ is also order-isomorphic to the set of pairs $(x, y) \in \mathcal{C}(S) \times \mathcal{C}(T)$ such that $\sigma x = \tau y$ and such that the induced bijection between x and y is secured, partially ordered by componentwise inclusion – we will use this description later on in the proofs.

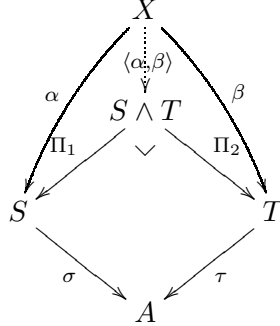
2.2.3. The interaction pullback. Events of $S \wedge T$ have the form φ with a top element (s, t) . The mappings $\Pi_1 : \varphi \mapsto s$ and $\Pi_2 : \varphi \mapsto t$ induce maps of event structures $S \wedge T \rightarrow S$ and $S \wedge T \rightarrow T$ that make the following diagram commute:

$$\begin{array}{ccc} & S \wedge T & \\ \Pi_1 \swarrow & & \searrow \Pi_2 \\ S & & T \\ \sigma \searrow & & \swarrow \tau \\ & A & \end{array}$$

Writing π_i for the (set-theoretic) projections, by Lemma 2.12, for every $x \in \mathcal{C}(S \wedge T)$ we have

$$\pi_1 \varphi_x = \Pi_1 x$$

as $\pi_1(s, t) = s = \Pi_1[(s, t)]_{\varphi_x}$ and similarly for π_2 and Π_2 . Those maps furthermore satisfy a universal property making formal the intuition of a “generalized intersection”: $(S \wedge T, \Pi_1, \Pi_2)$ is the *pullback* of $\sigma : S \rightarrow A$ and $\tau : T \rightarrow A$, meaning that that the above diagram commutes and for each map of event structures $\alpha : X \rightarrow S$ and $\beta : X \rightarrow T$ satisfying



there is a unique map $\langle \alpha, \beta \rangle : X \rightarrow S \wedge T$ such that $\Pi_1 \circ \langle \alpha, \beta \rangle = \alpha$ and $\Pi_2 \circ \langle \alpha, \beta \rangle = \beta$.

To construct $\langle \alpha, \beta \rangle$, we will need the following lemma stating the precise sense in which maps of event structures reflect the causal order:

Lemma 2.13. Let $f : A \rightarrow B$ be a map of event structures and $a, b \in A$ such that $\{a, b\}$ is consistent. If $f(a) \leq f(b)$ then $a \leq b$.

Proof. Since f is a map of event structures, $f[b]$ is down-closed as a configuration of B . Since $f(a) \leq f(b) \in f[b]$ by hypothesis, it follows that $f(a) \in f[b]$ and thus $f(a) = f(c)$ for some $c \leq b$. Since $\{a, b\}$ is consistent so is $\{a, b, c\}$ and local injectivity implies $a = c \leq b$ as desired. \square

We can now prove that our construction yields a pullback:

Lemma 2.14 (The interaction is a pullback). Let $\sigma : S \rightarrow A$ and $\tau : T \rightarrow A$ be maps of event structures. The triple $(S \wedge T, \Pi_1, \Pi_2)$ is a pullback for σ and τ .

Proof. We have already noticed that the inner square commutes.

Existence of $\langle \alpha, \beta \rangle$: Assume we have an event structure X with two maps $\alpha : X \rightarrow S$ and $\beta : X \rightarrow T$ such that $\sigma \circ \alpha = \tau \circ \beta$. Let $a \in X$. The bijection (by local injectivity of α, β):

$$\varphi_a = \{(\alpha a', \beta a') \mid a' \leq_X a\} : \alpha[a] \simeq \beta[a]$$

is secured as a consequence of Lemma 2.13, as a cycle in it would be reflected to X . Define $\langle \alpha, \beta \rangle(a) = [(\alpha(a), \beta(a))]_{\varphi_a}$ to be the secured bijection obtained as the down-closure of $(\alpha(a), \beta(a))$ inside the canonical order on the graph of φ_a : it has a maximal event by construction, and thus is an event of $S \wedge T$. It is a good exercise to check that this function defines a map of event structures; which makes the two triangles commute.

Uniqueness of $\langle \alpha, \beta \rangle$: Assume we have another map $\psi : X \rightarrow S \wedge T$ making the two triangles commute. We will check that $\langle \alpha, \beta \rangle$ and ψ have the same action on configurations, which will imply (by Lemma 2.15 below) that they are the same. Let $z \in \mathcal{C}(X)$. Its image through ψ and $\langle \alpha, \beta \rangle$ are (under the order-isomorphism $\mathcal{C}(S \wedge T) \cong \mathcal{B}_{\sigma, \tau}^{\text{sec}}$) secured bijections $\varphi : x \simeq y$ and $\varphi' : x' \simeq y'$. But by commutation of the two triangles in the pullback we must have $x = x' = \alpha z$ and $y = y' = \beta z$, thus $\varphi = \varphi'$ (as φ is uniquely determined from x, y by local injectivity). \square

In the proof of uniqueness, we only compared the maps by their action on configurations and deduced they were equal on events. This is justified by the following simple fact, that will be useful later on:

Lemma 2.15. Let $f, g : A \rightarrow B$ be parallel maps of event structures such that for all configuration $x \in \mathcal{C}(A)$ we have $fx = gx$. Then $f = g$.

Proof. Let $a \in A$. Write $[a]$ for the configuration $[a] \setminus \{a\}$. By hypothesis we have $f[a] = g[a]$ and $f[a] = g[a]$ as sets, thus $\{f(a)\} = f[a] \setminus f[a] = g[a] \setminus g[a] = \{g(a)\}$ and hence $f(a) = g(a)$. \square

2.3. Composition of pre-strategies. Building on our understanding of the interaction of pre-strategies as a pullback, we can now proceed to define the notion of composition, which is of critical importance in particular for the application of our games to semantics of programming languages. For that we need to define what is a pre-strategy σ from game A to game B , and given also τ from B to C , what is $\tau \odot \sigma$ from A to C .

Following Joyal [Joy77], we will define a pre-strategy from A to B to be simply a pre-strategy on the composite game $A^\perp \parallel B$. Let us show how to compose such pre-strategies. From $\sigma : S \rightarrow A^\perp \parallel B$ and $\tau : T \rightarrow B^\perp \parallel C$, we need to build a pre-strategy $\tau \odot \sigma$ on the game $A^\perp \parallel C$. Note that from such a notion of composition we can recover a notion of application when A is the empty event structure 1 . As usual in game semantics, composition is defined in two steps: firstly, we construct the *interaction* of the two strategies as an event structure where the two strategies communicate freely. Secondly, the internal synchronisation steps are *hidden* away. We will now detail these two steps.

To illustrate them, let \mathbb{B} be the game $\mathbf{tt}^+ \sim \mathbf{ff}^+$ of booleans (two conflicting positive events). Consider the following pre-strategies σ and τ respectively playing on $1^\perp \parallel \mathbb{B}_1$ and $\mathbb{B}_1^\perp \parallel \mathbb{B}_2$ (where indices are just there to disambiguate otherwise identical copies of \mathbb{B}):

$$\begin{array}{ccc}
 \mathbf{tt}_1^+ & \sim & \mathbf{ff}_1^+ \\
 & & \\
 & & \text{(}\sigma\text{)} \\
 & & \\
 \mathbf{ff}_2^+ & & \mathbf{tt}_2^+ \\
 \uparrow & & \uparrow \\
 \mathbf{tt}_1^- & \sim & \mathbf{ff}_1^- \\
 & & \\
 & & \text{(}\tau\text{)}
 \end{array}$$

The pre-strategy σ performs a nondeterministic choice: it can either play true or false. Likewise, τ computes the negation of a boolean: when Opponent plays true or false on \mathbb{B}_1 it answers the negation of that in \mathbb{B}_2 .

2.3.1. Interaction. Ignoring the polarities, σ and τ are maps of event structures $S \rightarrow A \parallel B$ and $T \rightarrow B \parallel C$. They do not play on the same game so it is not possible to make them interact directly. To solve this problem we pad them out with identity maps in order to get pre-strategies on $A \parallel B \parallel C$.

Thus we consider $\sigma \parallel \text{id}_C : S \parallel C \rightarrow A \parallel B \parallel C$ and $\text{id}_A \parallel \tau : A \parallel T \rightarrow A \parallel B \parallel C$. Since the identity map on any A accepts all possible behaviour appearing in A , only σ and τ give constraints on A and C respectively. In our example, the interaction is:

$$\begin{array}{ccc}
\mathbf{ff}_2 & & \mathbf{tt}_2 \\
\uparrow & & \uparrow \\
\mathbf{tt}_1 & \text{~~~~~} & \mathbf{ff}_1 \\
& (\sigma \parallel \text{id}_C) \wedge (\text{id}_A \parallel \tau) &
\end{array}$$

This interaction will be written $\tau \otimes \sigma : T \otimes S \rightarrow A \parallel B \parallel C$. (Note the change of order from $(\sigma \parallel \text{id}_C) \wedge (\text{id}_A \parallel \tau)$ to $\tau \otimes \sigma$, which reflects the standard notation for composition. In particular, when $A = C = 1$, $\sigma \wedge \tau$ is the same as $\tau \otimes \sigma$.)

2.3.2. Hiding. From $\tau \otimes \sigma : T \otimes S \rightarrow A \parallel B \parallel C$ we need to obtain a map to $A \parallel C$. For an event $p \in T \otimes S$ we say that it is **visible** if it maps to A or C , **invisible** otherwise. Let us write V for the set of visible events of $T \otimes S$.

We now obtain the composition by hiding invisible events: formally, $T \odot S = (T \otimes S) \downarrow V$. The obvious function $\tau \odot \sigma : T \odot S \rightarrow A \parallel C$, got as the restriction of $\tau \otimes \sigma$, defines a map of event structures. Polarities on $T \odot S$ are inherited from those of $A^\perp \parallel C$ to make $\tau \odot \sigma$ a pre-strategy on $A^\perp \parallel C$. In our example this yields the pre-strategy on \mathbb{B} (notice the inheritance of conflict – the conflict between \mathbf{ff}_2 and \mathbf{tt}_2 becomes minimal after hiding):

$$\begin{array}{ccc}
\mathbf{ff}_2^+ & \text{~~~~~} & \mathbf{tt}_2^+ \\
& \tau \odot \sigma &
\end{array}$$

We get back the original nondeterministic boolean – the non-deterministic boolean is invariant under negation. But in what sense is it the same, exactly?

2.3.3. Isomorphisms of pre-strategies. They are not equal (set-theoretically) because the underlying sets are not the same, but they are *isomorphic*:

Definition 2.16 (Isomorphism of pre-strategies). Let $\sigma : S \rightarrow A$ and $\tau : T \rightarrow A$ be two pre-strategies on a common game A . An **isomorphism** between σ and τ is an isomorphism of event structures $\phi : S \cong T$ commuting with the action on the game:

$$\begin{array}{ccc}
S & \xrightarrow{\phi} & T \\
\searrow \sigma & & \swarrow \tau \\
& A &
\end{array}$$

In this case, we write $\phi : \sigma \cong \tau$ or simply $\sigma \cong \tau$ when the specific ϕ does not matter.

Isomorphism is the most precise equivalence that makes sense on pre-strategies: two isomorphic pre-strategies have the same intensional behaviour.

Constructing isomorphisms at the level of events can be sometimes cumbersome especially in the case when the event structures are generated from an order of configurations as is the case for the interaction (Section 2.2). Fortunately, order-isomorphisms between configurations of event structures induce isomorphisms on the event structures (*cf.* [NPW81]).

Lemma 2.17. Let A and B be event structures. For every order-isomorphism $\varphi : \mathcal{C}(A) \cong \mathcal{C}(B)$ there is a unique isomorphism of event structures $\hat{\varphi} : A \cong B$ satisfying $\hat{\varphi}(x) = \varphi(x)$ for every configuration $x \in \mathcal{C}(A)$. This induces a bijective correspondence between order-isomorphisms $\mathcal{C}(A) \cong \mathcal{C}(B)$ and isomorphisms of event structures $A \cong B$.

Proof. Since it is an order-isomorphism, φ preserves the covering relation on configurations. We define $\hat{\varphi}$ through this property: indeed for all $a \in A$ we have $[a] \text{--} \subset [a]$, therefore $\varphi[a] \text{--} \subset \varphi[a]$; let us write $\hat{\varphi}a$ for the event added in this covering. In order to establish that $\hat{\varphi}$ is a map of event structures whose image of configurations matches φ , the key property will be that for all $x \text{--} \subset x \cup \{a\}$, the event added in $\varphi x \text{--} \subset \varphi(x \cup \{a\})$ is indeed $\hat{\varphi}a$.

For that, we remark that φ preserves commuting squares of coverings of the form:

$$\begin{array}{ccc} y_1 & \xrightarrow{a_1} & z \\ a_2 \downarrow & & a_2 \downarrow \\ x & \xrightarrow{a_1} & y_2 \end{array}$$

in the sense that their images are squares where parallel arrows correspond to the same event. Since φ preserves $\text{--} \subset$, the image of the square as above is:

$$\begin{array}{ccc} \varphi y_1 & \xrightarrow{b'_1} & \varphi z \\ b_2 \downarrow & & b'_2 \downarrow \\ \varphi x & \xrightarrow{b_1} & \varphi y_2 \end{array}$$

If $\varphi y_1 = \varphi y_2$ then $y_1 = y_2$ so $a_1 = a_2$ and $a_1 \in y_1$, contradicting $y_1 \xrightarrow{a_1}$. So since $\varphi y_1 \neq \varphi y_2$ we have $b_2 \neq b_1$, therefore $b_1 = b'_1$ and $b_2 = b'_2$.

Now, by induction on x we prove that $\hat{\varphi}x = \varphi x$. Clearly $\varphi \emptyset = \hat{\varphi} \emptyset = \emptyset$. Now take $x \xrightarrow{a} y$, and write $\varphi x \xrightarrow{b} \varphi y$. We have covering diagrams as below:

$$\begin{array}{ccc} x & \xrightarrow{a} & y \\ \downarrow & & \downarrow \\ \vdots & & \vdots \\ \downarrow & & \downarrow \\ [a] & \xrightarrow{a} & [a] \end{array} \quad \mapsto \quad \begin{array}{ccc} \varphi x & \xrightarrow{b} & \varphi y \\ \downarrow & & \downarrow \\ \vdots & & \vdots \\ \downarrow & & \downarrow \\ \varphi[a] & \xrightarrow{\hat{\varphi}a} & \varphi[a] \end{array}$$

where the left hand side diagram decomposes into commuting squares of coverings where all horizontal coverings add a . Since those are preserved, it follows that $b = \hat{\varphi}a$. Hence $\hat{\varphi}y = \hat{\varphi}x \cup \{\hat{\varphi}a\} = \varphi x \cup \{b\} = \varphi y$.

Obviously it follows that $\hat{\varphi}$ preserves configurations. It is also locally injective since x and $\varphi x = \hat{\varphi}x$ have the same cardinal (as φ preserves coverings). Thus $\hat{\varphi}$ is a map of event structures. From Lemma 2.15 it follows that $\hat{\varphi}$ and $\hat{\varphi}^{-1}$ are inverses.

Uniqueness is obvious by Lemma 2.15 and the bijective correspondence follows. \square

It will follow from the developments of Section 4 that up to this notion of isomorphism of pre-strategies, composition is associative:

Proposition 2.18. Let $\sigma : S \rightarrow A^\perp \parallel B$, $\tau : T \rightarrow B^\perp \parallel C$ and $\rho : U \rightarrow C^\perp \parallel D$ be pre-strategies. Then, there is an isomorphism $\alpha_{\sigma,\tau,\rho} : (U \odot T) \odot S \rightarrow U \odot (T \odot S)$ making the following diagram commute:

$$\begin{array}{ccc}
 (U \odot T) \odot S & \xrightarrow{\alpha_{\sigma,\tau,\rho}} & U \odot (T \odot S) \\
 \searrow^{(\rho \odot \tau) \odot \sigma} & & \swarrow_{\rho \odot (\tau \odot \sigma)} \\
 & A^\perp \parallel D &
 \end{array}$$

Proof. The isomorphism is constructed in Section 4.3. □

To accompany this associative composition, the next section will start by introducing a *copycat pre-strategy*, that serves as a candidate for an identity. The copycat pre-strategy

$$\mathfrak{c}_A : \mathbb{C}_A \rightarrow A^\perp \parallel A$$

is an *asynchronous forwarder*: every negative move on one side triggers the corresponding positive move on the other side. It is idempotent, but we will see that it is not an identity: pre-strategies and copycat *do not form a category*. We will define *strategies* as those pre-strategies for which copycat is an identity, and characterise them concretely.

3. STRATEGIES

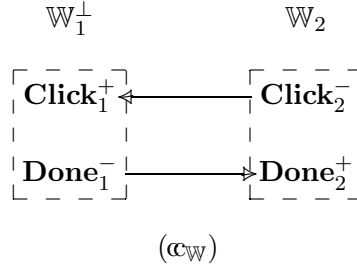
As previously hinted at, pre-strategies currently take little account of polarity, and hence have an unreasonable expressive power: they can for instance constrain the order in which Opponent plays their moves, or prevent them from playing at all. We have encountered just before Section 2.2 a simple example of that: the empty pre-strategy on the game \ominus with just one negative move. By not acknowledging the \ominus , Player denies Opponent the right to play, even though the game allows it.

One guiding principle for the notion of strategy is that they should form a category, so there should be a copycat strategy, neutral for composition with respect to other strategies. The presence of an identity for composition is of course key to the application of our setting in denotational semantics, which relies on a categorical formalisation, but we argue that there is a more down-to-earth motivation for it. The *copycat strategy*, to be introduced formally below, acts as an *asynchronous forwarder*. Accordingly, composing with copycat will eliminate overly *synchronous* behaviour from pre-strategies. Examples include the pre-strategy in the previous paragraph not acknowledging Opponent's move, or a pre-strategy playing on a game $\oplus_1 \oplus_2$ with two independent positive events, which plays the moves $\oplus_1 \rightarrow \oplus_2$ in order. As the moves are independent in the game, the ordering $\oplus_1 \rightarrow \oplus_2$ played by the strategy will not be respected by an asynchronous environment – two successive packets sent on the network might arrive in the other order. These intuitions will be revisited formally via examples after the definition of the copycat pre-strategy.

In this section we will define the copycat (pre)strategy, and then characterise the *strategies*: those pre-strategies invariant under their composition with copycat. We provide examples of pre-strategies that do not behave well in presence of latency and give two criteria (*courtesy* and *receptivity*) that are proved necessary and sufficient for a pre-strategy to be a strategy (Theorem 3.20).

3.1. Copycat and its action on strategies. On $A^\perp \parallel A$, each move of A appears twice (with dual polarities). The copycat pre-strategy waits for a negative occurrence to be played and then plays the corresponding positive move. In formal terms, it has the causality $(1 - i, a) \rightarrow (i, a)$ for every positive move (i, a) of $A^\perp \parallel A$. Note that this behaviour corresponds to that of the usual copycat strategy in game semantics.

For instance, on the game $\mathbb{W} = \mathbf{Click}^- \mathbf{Done}^+$ of an interface where Player (the program) can signal it has finished a long computation or Opponent (the user) can click on the screen, the copycat strategy looks like:



Copycat forwards the negative events from one side to the other: acting as the program on the right and as the user on the left. Even if copycat is a pre-strategy from \mathbb{W} to itself, it does not necessarily entail a left-to-right flow of information as can be seen for the event **Click**, rather *from negative to positive*. This general construction yields a pre-strategy playing on $A^\perp \parallel A$ for any game A .

Definition 3.1 (Copycat). Let A be a game. Define \mathfrak{C}_A to be the following event structure:

- *Events*: those of $A^\perp \parallel A$,
- *Causality*: the transitive closure of

$$\leq_{A^\perp \parallel A} \cup \{((1 - i, a), (i, a)) \mid (i, a)^+ \in A^\perp \parallel A\}$$

- *Consistency*: X is consistent in \mathfrak{C}_A iff its down-closure $[X] = \{a \in \mathfrak{C}_A \mid \exists b \in X, a \leq_{\mathfrak{C}_A} b\}$ is consistent in $A^\perp \parallel A$.

This makes an event structure and the identity map is a pre-strategy:

Lemma 3.2. For any game A , \mathfrak{C}_A is an esp (with polarities inherited from $A^\perp \parallel A$), and the identity map written $\mathfrak{C}_A : \mathfrak{C}_A \rightarrow A^\perp \parallel A$ is a pre-strategy, the **copycat pre-strategy**.

Proof. We observe that for $(i, a), (j, a') \in \mathfrak{C}_A$, we have $(i, a) \leq_{\mathfrak{C}_A} (j, a')$ iff:

- Either, $i = j$ and $a \leq_A a'$,
- Or, $i \neq j$, and there is $a \leq_A a'' \leq_A a'$ such that $\text{pol}_{\mathfrak{C}_A}((i, a'')) = -$ and (by necessity) $\text{pol}_{\mathfrak{C}_A}((j, a'')) = +$.

Indeed, this is a transitive relation that contains the generators for $\leq_{\mathfrak{C}_A}$ – dually, two events related by the relation above are related by $\leq_{\mathfrak{C}_A}$. The other axioms of event structures follow easily, and it is trivial that $\mathfrak{C}_A : \mathfrak{C}_A \rightarrow A^\perp \parallel A$ is a map of event structures. \square

Immediate causal links in copycat have a very specific shape:

Lemma 3.3. We have that $(i, a) \rightarrow_{\mathfrak{C}_A} (j, a')$ if and only if one of the two following conditions is met:

- (1) Either $i = j$, $a \rightarrow_A a'$ and either (i, a) is positive in \mathfrak{C}_A or (j, a') is negative in \mathfrak{C}_A .

(2) Or $i \neq j$ and $a = a'$ and $(i, a) \in \mathbb{C}_A$ is negative.

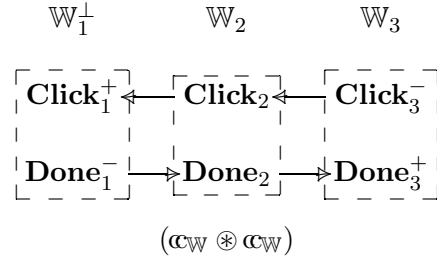
Proof. It is clear that both conditions imply $(i, a) \rightarrow_{\mathbb{C}_A} (j, a')$. Conversely, we know $\leq_{\mathbb{C}_A}$ is generated by $\rightarrow_{A^\perp \parallel A} \cup \{(i, a), (1-i, a) \mid (i, a)^- \in \mathbb{C}_A\}$. This means that $(i, a) \rightarrow (j, a')$ implies either $i \neq j$, $a = a'$ and $(i, a)^- \in \mathbb{C}_A$ (as desired) or $i = j$ and $a \rightarrow_A a'$. In this case, if (i, a) is negative and (j, a') is positive, we have $(i, a) \rightarrow_{\mathbb{C}_A} (1-i, a) <_{\mathbb{C}_A} (1-i, a') \rightarrow_{\mathbb{C}_A} (i, a')$ contradicting $(i, a) \rightarrow_{\mathbb{C}_A} (j, a')$. Hence (i, a) is positive. \square

Copycat acts on pre-strategies on A via composition: $\sigma \mapsto \mathfrak{c}_A \odot \sigma$. This action adds *latency* to pre-strategies: whenever the pre-strategy plays a positive move it has to be forwarded by copycat before being visible. We can now define strategies:

Definition 3.4 (Strategy). A **strategy** on a game A is a pre-strategy $\sigma : S \rightarrow A$ such that $\mathfrak{c}_A \odot \sigma \cong \sigma$.

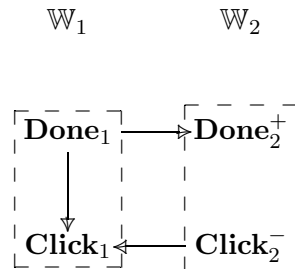
This isomorphism is, in general, not unique: in fact, strategies have in general a non-trivial group of automorphisms. Think, for instance, of the strategy $\sigma : S \rightarrow \mathbb{B}$ where S has just two conflicting events $s \sim s'$, both mapped to \mathfrak{tt} . This strategy σ has two automorphisms: the identity and the swap on S . Likewise, there are two isomorphisms $\mathfrak{c}_{\mathbb{B}} \odot \sigma \cong \sigma$. Despite this, it will follow from our development that if $\mathfrak{c}_A \odot \sigma \cong \sigma$, then there is always a *canonical* such isomorphism that fits in the bicategorical picture of Section 4.

Let us try to understand this definition through examples. Consider first the composition $\mathfrak{c}_{\mathbb{W}} \odot \mathfrak{c}_{\mathbb{W}}$ with $A = \mathbb{W}_1, B = \mathbb{W}_2$ and $C = \mathbb{W}_3$:



Hiding events in \mathbb{W}_2 yields a pre-strategy isomorphic to $\mathfrak{c}_{\mathbb{W}}$. The latency can be observed: immediate causal links of the form $- \rightarrow +$ get delayed in the interaction to $- \rightarrow * \rightarrow +$ where $*$ denotes an invisible event of the interaction. After hiding, the effect disappears here but it is not the case in general. Two situations can appear, calling for two conditions.

3.1.1. *Courtesy.* Assume we have the pre-strategy σ with event structure $\mathbf{Done}^+ \rightarrow \mathbf{Click}^-$ on \mathbb{W} that forces the user to wait for the computation to be over before allowing them to click. Computing the interaction $\mathfrak{c}_{\mathbb{W}} \otimes \sigma$ with $A = 1, B = \mathbb{W}_1$ and $C = \mathbb{W}_2$ yields:



After hiding of $B = \mathbb{W}_1$, $\mathbb{C}_\mathbb{W} \odot \sigma$ has event structure $\mathbf{Click}_2^- \mathbf{Done}_2^+$. There is no causal link anymore because in the interaction the two events are concurrent. Copycat *will* allow the user to Click without waiting for σ 's constraint: there is no way for σ to impose this particular order of moves. In other terms the causal link is not stable under the latency added by copycat.

As a consequence, for a pre-strategy to be invariant under the action of copycat it must not have immediate causal links of the form $+ \rightarrow -$ *that were not already present in the game*. In our setting, playing a move is similar to sending a packet whose sender (Player or Opponent) is given by the polarity. This condition means that unless the protocol (the game) specifies it, there is no way to force Opponent to wait for a Player message before sending their message.

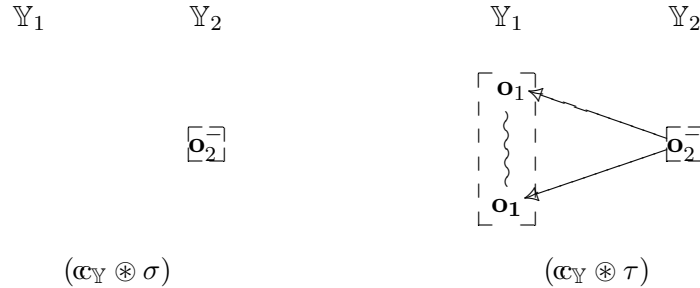
Similar reasoning can be made for immediate causal links $- \rightarrow -$ (one cannot control the order in which Opponent sends out messages) and $+ \rightarrow +$ (latency can change the order in which independent messages arrive).

A pre-strategy respecting these constraints will be called *courteous*²:

Definition 3.5 (Courtesy). A pre-strategy $\sigma : S \rightarrow A$ is **courteous** when for all $s, s' \in S$ such that $s \rightarrow s'$ and $(\text{pol}(s), \text{pol}(s')) \neq (-, +)$, then $\sigma s \rightarrow \sigma s'$.

3.1.2. *Receptivity*. Consider the game $\mathbb{Y} = \mathbf{o}^-$ comprising a single negative event, and the two pre-strategies σ and τ on this game, with respective event structures \emptyset (no moves played by σ) and $\mathbf{o}^- \sim \mathbf{o}^-$ (τ can acknowledge the unique negative event in two different ways, non-deterministically).

Their respective interactions with copycat on \mathbb{Y} give (with $A = 1, B = \mathbb{Y}_1$ and $C = \mathbb{Y}_2$):



After hiding, only \mathbf{o}_2^- is left in both cases. The problem with these pre-strategies is that they either duplicate or ignore a negative event – yet as we have seen, copycat acknowledges available negative moves *first* without depending on the pre-strategy's behaviour. Strategies must therefore have the same behaviour regarding the negative events as copycat: to accept them as soon as they are enabled in the current state of the game, and play them *once*. Such pre-strategies will be called *receptive*:

Definition 3.6 (Receptivity). A pre-strategy $\sigma : S \rightarrow A$ is **receptive** when for each configuration $x \in \mathcal{C}(S)$ such that $\sigma x \xrightarrow{a^-} \text{C}$ there exists a *unique* $s \in S$ (necessarily negative) such that $x \xrightarrow{s} \text{C}$ and $\sigma s = a$.

²This condition was called *innocence* in [RW11]. Courtesy is preferred here to avoid the misleading collision with innocence in the sense of Hyland and Ong [HO00].

For readers familiar with game semantics, it might be helpful to note that in standard games models receptivity is always present in one way or another. It is explicit and named *contingent completeness* in [HO00], but most of the time it is hard-wired in by asking that strategies contain only plays of even length (Opponent extensions being always present, they bring no additional information).

3.2. The characterisation of strategies – overview of the proof. At this point, the main definitions for the framework are in place. The main element which is missing, is the fact that for a pre-strategy $\sigma : S \rightarrow A$, it is equivalent to be a *strategy* (in the sense of Definition 3.4), and to be *receptive* and *courteous* – which was the main result of [RW11]. The rest of this section is devoted to proving this result, stated in Theorem 3.20. In this paper we give a different proof than the one developed in [RW11]. Our new proof is more high-level and modular, and sets up the stage better for extensions of the framework in future papers. The rest of the section is quite technical, and may be skimmed through in a first reading of the paper. We start by giving a high-level overview of the proof.

According to Definition 3.4, $\sigma : S \rightarrow A$ is a strategy if $\mathbb{C}_A \odot \sigma \cong \sigma$. By Lemma 2.17, that means that there is an order-isomorphism

$$\mathcal{C}(S) \cong \mathcal{C}(\mathbb{C}_A \odot S)$$

commuting with the projection to A . In order to characterise the existence of such an isomorphism, we need to study configurations of $\mathbb{C}_A \odot S$ for any pre-strategy $\sigma : S \rightarrow A$. This will be done in several steps.

3.2.1. Decomposing interactions. Taking $z \in \mathcal{C}(\mathbb{C}_A \odot S)$, we have its minimal witness $[z] \in \mathcal{C}(\mathbb{C}_A \otimes S)$. By Lemma 2.12, $[z]$ corresponds to a secured bijection:

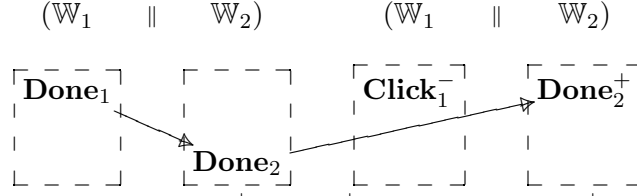
$$\varphi_{[z]} : x \simeq y$$

with $x = x_S \parallel x_A \in \mathcal{C}(S \parallel A)$ and $y = y_{A^\perp} \parallel y_A \in \mathcal{C}(\mathbb{C}_A)$ such that $\sigma x_S = y_{A^\perp}$ and $x_A = y_A$ – in fact, as remarked below Lemma 2.12, by local injectivity, $\varphi_{[z]}$ (and so $[z]$) is determined by such x and y , *i.e.*, by x_S and y_A .

We write $\Psi([z]) = (x_S, y_A) \in \mathcal{C}(S) \times \mathcal{C}(A)$ for this pair, which satisfies that $x_S \in \mathcal{C}(S)$ and $\sigma x_S \parallel y_A \in \mathcal{C}(\mathbb{C}_A)$. Reciprocally (by Lemma 2.12) any such pair induces a configuration of $\mathbb{C}_A \otimes S$ provided the corresponding bijection is secured – but that is always the case, as we will see; so Ψ is an iso. We will also characterise such pairs which, through Ψ , correspond to an interaction whose maximal elements are visible (*i.e.* a minimal witness of a configuration of $\mathbb{C}_A \odot S$). This will yield a complete description of configurations of $\mathbb{C}_A \odot S$ in terms of certain pairs of configurations $(x, y) \in \mathcal{C}(S) \times \mathcal{C}(A)$ (step #1).

For $z \in \mathcal{C}(\mathbb{C}_A \otimes S)$ and $\Psi(z) = (x_S, x_A)$, one may regard x_A as a not completely updated version of σx_S : some negative events of x_A may not have made their way to σx_S , and reciprocally.

Example 3.7. Consider the pre-strategy σ playing on $\mathbb{W}_1 \parallel \mathbb{W}_2$, with event structure $\mathbf{Done}_1^+ \rightarrow \mathbf{Done}_2^+$. The following diagram represents an interaction $z \in \mathcal{C}(\mathbb{C}_A \otimes S)$ of σ with copycat.



Here, we have $\Psi(z) = (\{\mathbf{Done}_1^+, \mathbf{Done}_2^+\}, \{\mathbf{Click}_1^-, \mathbf{Done}_2^+\})$.

In the example above, we observe two phenomena: the event \mathbf{Click}_1^- is played on the right hand side but not forwarded to the left hand side, and the event \mathbf{Done}_1^+ is played on the left hand side but not forwarded to the right hand side. In general, with $\Psi(z) = (x_S, x_A)$, the constraint that $\sigma x_S \parallel x_A \in \mathcal{C}(\mathbb{C}_A)$ means that σx_S has *less negative events* and *more positive events* than x_A , *i.e.*

$$x_A \supseteq^- x_A \cap (\sigma x_S) \subseteq^+ \sigma x_S$$

This relation $x \supseteq^- \subseteq^+ y$ is in fact a partial order on $\mathcal{C}(A)$ called the *Scott order* [Win13b], and written \sqsubseteq_A . It will yield (step #2) a characterisation of configurations of copycat as pairs $(x_S, x_A) \in \mathcal{C}(S) \times \mathcal{C}(A)$ such that $x_A \sqsubseteq_A \sigma x_S$.

To summarise, after steps #1 and #2, we will have achieved an equivalent description of interactions $z \in \mathcal{C}(\mathbb{C}_A \otimes S)$ as the data of $(x_S, x_A) \in \mathcal{C}(S) \times \mathcal{C}(A)$ such that $x_A \sqsubseteq_A \sigma x_S$, *i.e.* as diagrams:

$$\begin{array}{c}
x_S \\
\downarrow \sigma \\
x_A \sqsubseteq_A \sigma x_S
\end{array}$$

whose projection to the game via $\mathbb{C}_A \otimes \sigma : \mathbb{C}_A \otimes S \rightarrow A \parallel A$ is $\sigma x_S \parallel x_A$, where only x_A will be visible after hiding. We now try to produce an isomorphism between configurations of $\mathbb{C}_A \otimes S$ that are minimal witnesses of configurations of $\mathbb{C}_A \odot S$ (those whose maximal events are visible), and configurations of S . We will build transformations of configurations in the two directions.

3.2.2. The isomorphism. Constructing the left-to-right part of the isomorphism $\mathbb{C}_A \odot S \cong S$, we need to associate to any representation of an interaction $(x_S, x_A) \in \mathcal{C}(S) \times \mathcal{C}(A)$ as above, some $x'_S \in \mathcal{C}(S)$ mapping to x_A via σ . Diagrammatically:

$$\begin{array}{ccc}
x_S & \implies & \exists x'_S \quad x_S \\
\downarrow \sigma & & \downarrow \sigma \quad \downarrow \sigma \\
x_A \sqsubseteq_A \sigma x_S & & x_A \sqsubseteq_A \sigma x_S
\end{array}$$

In fact, it will turn out that $x'_S \sqsubseteq_S x_S$, and (for the correspondence to be an iso) that its choice is unique. In other words, we will extract x'_S by proving that strategies are *discrete fibrations*, as in Definition 3.12 (step #3).

We now focus on the right-to-left part of the construction. From $x \in \mathcal{C}(S)$, we need to provide some configuration of $\mathbb{C}_A \odot S$; so we need to provide a witness in $\mathcal{C}(\mathbb{C}_A \otimes S)$. As we have seen, via Ψ we are looking for a pair (x_S, x_A) such that $x_A \sqsubseteq_A \sigma x_S$. Note that x_A is determined by the requirement that $\sigma x = x_A$. From that it seems that the pair (x, x_A) does the trick: we do indeed have $z = \Psi^{-1}(x, x_A) \in \mathcal{C}(\mathbb{C}_A \otimes S)$ – and restricting it to its *visible*

events yields the desired configuration of $\mathbb{C}_A \odot S$. However, it will be useful in proving the isomorphism to have the *minimal* interaction – the minimal witness – corresponding to this configuration of the composition through hiding. The interaction $\Psi^{-1}(x, x_A)$ is not always minimal:

Example 3.8. Consider $\sigma : S \rightarrow \mathbb{W}$ with S comprising only one event s mapped to \mathbf{Click}_1^- . Following the paragraph above, its configuration $\{s\}$ leads to an interaction with copycat corresponding to $(\{s\}, \{\mathbf{Click}_1^-\})$, represented as:

$$\begin{array}{ccc} \mathbb{W} & \parallel & \mathbb{W} \\ \mathbf{Click}_1 & \longleftarrow & \mathbf{Click}_1^- \end{array}$$

Disposing of the left hand side \mathbf{Click}_1 yields a *smaller* interaction witnessing the same configuration of the composition, as it is maximal and not visible.

In fact, for $x \in \mathcal{C}(S)$ there is a *unique* $x^* \subseteq x$ such that $(x^*, \sigma x)$ yields the same configuration of the composition as $(x, \sigma x)$, and such that the maximal events of the represented interaction are all visible. As we will see x^* is obtained from x as above, by removing maximal negative events (step #4). From this uniqueness property and the discrete fibration property, it follows that these constructions are inverses of each other.

3.2.3. *Necessity.* From the above, we know that strategies, as discrete fibrations, compose well with copycat. It remains to show the converse: that strategies which compose well with copycat are discrete fibrations. In other words, we need to show that strategies of the form $\mathbb{C}_A \odot \sigma$ are always discrete fibrations. That will be a direct verification, once we have characterised the Scott order on $\mathbb{C}_A \odot S$ (step #5).

3.3. **Proof of the characterisation of strategies.** Now, we detail and prove all the steps mentioned above.

3.3.1. *Step #1: Composition witnesses as pairs.* We start by showing that there are no possible causal loops in an interaction with copycat, so that such interactions are entirely characterised by matching pairs of configurations. In fact we prove a slight generalisation.

Lemma 3.9 (Deadlock-free lemma). Let $\tau : T \rightarrow A^\perp \parallel B$ be a pre-strategy such that if $t \leq t'$ and both t and t' are sent by τ to the component A^\perp , then $\tau t \leq \tau t'$. Then, given a pre-strategy $\sigma : S \rightarrow A$, and configurations x of S and y of T with $\sigma x \parallel z = \tau y$ for some configuration z of B , the induced bijection $x \parallel z \simeq y$ is secured.

As a consequence, we have an order isomorphism:

$$\mathcal{C}(T \otimes S) \cong \{(x, y) \in \mathcal{C}(S) \times \mathcal{C}(T) \mid \sigma x \parallel z = \tau y \text{ for some } z \text{ in } \mathcal{C}(B)\}$$

Proof. Assume that the bijection is not secured. Without loss of generality, there is a causal loop of the form $(v_1, t_1) \triangleleft \dots \triangleleft (v_{2n}, t_{2n})$ such that $t_{2i} < t_{2i+1}$ and $v_{2i+1} < v_{2i+2}$ and $t_{2n} < t_1$. Note that $v_i \in S \parallel B$ for every i .

Assume that $v_{2i+1} \in B$. Then $v_{2i+2} \in B$ and we have that $\tau(t_{2i+1}) = v_{2i+1} \leq v_{2i+2} = \tau(t_{2i+2})$. Hence by Lemma 2.13, it follows that $t_{2i+1} \leq t_{2i+2}$. If the only two steps of the causal loop were (v_{2i+1}, t_{2i+1}) and (v_{2i}, t_{2i}) , we have a loop in T and a contradiction. Otherwise, we can remove the steps $2i+1$ and $2i+2$ and keep a causal loop. Removing them, if there is a loop of length one remaining, then we have a direct contradiction (*i.e.* $t_1 < t_1$).

Otherwise without loss of generality we can assume $v_i \in S$ for every i . In this case, by hypothesis on τ we have that $t_{2i} < t_{2i+1}$ implies that $\sigma v_{2i} = \tau t_{2i} < \tau t_{2i+1} = \sigma v_{2i+1}$. By Lemma 2.13 again, it follows that $v_1 < \dots < v_1$ – a contradiction.

This establishes that the bijection induced by any pair of synchronized configurations (w, y) is secured and thus is a configuration of the interaction. We conclude with the sequence of order-isos:

$$\begin{aligned} \mathcal{C}(T \otimes S) &\cong \{\varphi : w \simeq y \text{ secured} \mid \\ &\quad w \in \mathcal{C}(S \parallel B), y \in \mathcal{C}(T) \text{ such that } \tau y = (\sigma \parallel B) w\} \\ &\cong \{\varphi : w \simeq y \mid w \in \mathcal{C}(S \parallel B), y \in \mathcal{C}(T) \text{ such that } \tau y = (\sigma \parallel B) w\} \\ &\cong \{(x \parallel z, y) \in \mathcal{C}(S \parallel B) \times \mathcal{C}(T) \mid \sigma x \parallel z = \tau y\} \\ &\cong \{(x, y) \in \mathcal{C}(S) \times \mathcal{C}(T) \mid \sigma x \parallel z = \tau y \text{ for some } z \text{ in } \mathcal{C}(B)\} \end{aligned}$$

□

Let $\sigma : S \rightarrow A$ be pre-strategy. The previous lemma, instantiated with $\tau = \mathbb{C}_A$, gives an order-isomorphism:

$$\begin{aligned} \Psi_\sigma : \mathcal{C}(\mathbb{C}_A \otimes S) &\cong \{(x, y_1 \parallel y_2) \in \mathcal{C}(S) \times \mathcal{C}(\mathbb{C}_A) \mid \sigma x = y_1\} \\ &\cong \{(x, y) \in \mathcal{C}(S) \times \mathcal{C}(A) \mid \sigma x \parallel y \in \mathcal{C}(\mathbb{C}_A)\} \end{aligned}$$

Every such pair represents an interaction, which gives through hiding a configuration of $\mathbb{C}_A \odot S$. However, many interactions correspond to the same configuration of the composition. In fact, as we have seen in Section 2.3, configurations of $\mathbb{C}_A \odot S$ bijectively correspond to interactions in $\mathbb{C}_A \otimes S$ whose maximal events are visible. We now characterise them.

Lemma 3.10. Let $\varphi : x \parallel y \simeq \sigma x \parallel y$ be a secured bijection corresponding to a configuration of $\mathbb{C}_A \otimes S$. The following are equivalent:

- (i) All maximal events of φ are visible
- (ii) Every maximal event s of x is positive and $\sigma s \in y$.

Moreover, in this case, if σ is courteous, we have $\sigma x \subseteq^- y$.

Proof. (i) \Rightarrow (ii). Let $s \in x$ be a maximal event. The event $c = ((0, s), (0, \sigma s))$ is not visible in φ . Hence it is not maximal: there exists $c' \in \varphi$ such that $c \rightarrow_\varphi c'$. By Lemma 2.10, there are two cases:

- Either $\pi_1 c \rightarrow_{S \parallel A} \pi_1 c'$, *i.e.* $c' = ((0, s'), (0, \sigma s'))$ and $s \rightarrow_x s'$: this is absurd as s is maximal in x .
- Or $\pi_2 c \rightarrow_{\mathbb{C}_A} \pi_2 c'$: by Lemma 3.3, there are two possibilities. The first one is that $c' = ((0, s'), (0, \sigma s'))$: absurd, as it would entail $\sigma s \rightarrow \sigma s'$ and $s < s'$ by Lemma 2.13 contradicting maximality. The second one is that $c' = ((1, \sigma s), (1, \sigma s))$.

This means that $(1, \sigma s)$ is positive in \mathbb{C}_A , *i.e.* s is positive, and moreover $(1, \sigma s) \in \sigma x \parallel y$ so $\sigma s \in y$.

(ii) \Rightarrow (i). Let c be a maximal event of φ and assume it is not visible. It is then of the form $c = ((0, s), (0, \sigma s))$. If $s \rightarrow_x s'$ then $c <_\varphi ((0, s'), (0, \sigma s'))$ which is absurd so s must be maximal in x . By assumption s is positive and $\sigma s \in y$. Then we have $(0, \sigma s) \rightarrow_{\mathbb{C}_A} (1, \sigma s)$ so $c <_\varphi ((1, \sigma s), (1, \sigma s))$ which contradicts the maximality of c .

Finally, assume σ is courteous. We prove that maximal events of σx are included in y . Take $\sigma s \in \sigma x$ a maximal event. If s is negative then $(0, \sigma s)$ is positive in $A^\perp \parallel A$.

Therefore we have $(1, \sigma s) \leq_{\mathbb{C}_A} (0, \sigma s)$. Since $\sigma x \parallel y \in \mathcal{C}(\mathbb{C}_A)$, we are done. Otherwise, if s is positive it has to be maximal in x : indeed if we had $s^+ \rightarrow_x s'$, by courtesy $\sigma s \rightarrow_{\sigma x} \sigma s'$ would follow contradicting the maximality of σs . Then we can conclude by assumption: $\sigma s \in y$ as desired. \square

Summarizing step #1, we now know that configurations of $\mathbb{C}_A \odot S$ correspond, in an order-preserving and order-reflecting way, to pairs of configurations $(x, y) \in \mathcal{C}(S) \times \mathcal{C}(A)$, such that $\sigma x \parallel y \in \mathcal{C}(\mathbb{C}_A)$, and such that the maximal events of x are positive and also appear in y .

Now, we study the requirement that $\sigma x \parallel y \in \mathcal{C}(\mathbb{C}_A)$.

3.3.2. Step #2: The Scott order. As observed before, for $x, y \in \mathcal{C}(A)$, $y \parallel x \in \mathcal{C}(\mathbb{C}_A)$ whenever y has more positive events and less negative events than x . More precisely:

Lemma 3.11 (Scott order). Let $x, y \in \mathcal{C}(A)$. The following are equivalent:

- (i) $y \parallel x \in \mathcal{C}(\mathbb{C}_A)$
- (ii) $x \supseteq^- (x \cap y) \subseteq^+ y$ (where $x \subseteq^+ y$ means that $x \subseteq y$ and $\text{pol}(y \setminus x) \subseteq \{+\}$ and similarly for $x \supseteq^- y$)
- (iii) there exists $z \in \mathcal{C}(A)$ such that $x \supseteq^- z \subseteq^+ y$.

In this case we write $x \sqsubseteq_A y$: this is an order called **the Scott order** of A .

Proof. (i) \Rightarrow (ii). We show $x \cap y \subseteq^+ y$; the other inclusion is similar. Let $a^- \in y$, we must show it is in x . Since $(0, a) \in A^\perp \parallel A$ is positive, we have $(1, a) <_{\mathbb{C}_A} (0, a)$. The down-closure of $y \parallel x$ implies that $(1, a) \in y \parallel x$ as $a \in y$. This exactly means that $a \in x$ as desired.

(ii) \Rightarrow (iii). clear.

(iii) \Rightarrow (i). Assume we have $x \supseteq^- z \subseteq^+ y$. The set $y \parallel x$ is clearly consistent so we need only prove it is down-closed. Since x and y are already down-closed in A , we need only to check for the additional immediate causal links. Assume we have $(1, a^+) \in y \parallel x$ (so $a \in x$). By hypothesis we have $a \in z$ because it is positive. Since $z \subseteq y$ we deduce $a \in y$ that is $(0, a) \in y \parallel x$ as desired. The case $(0, a^-) \in y \parallel x$ is similar.

It is an order. It is clearly reflexive. If $x \supseteq^- (x \cap y) \subseteq^+ y$ and $y \supseteq^- (x \cap y) \subseteq^+ x$, it follows that $x \setminus x \cap y$ has to be empty thus $x = x \cap y = y$.

For transitivity assume $x \supseteq^- (x \cap y) \subseteq^+ y \supseteq^- (y \cap z) \subseteq^+ z$. Then if $a \in x \setminus z$, there are two cases. If $a \in y$, then since $a \notin y \cap z$, from $y \cap z \subseteq^- y$ we know that a is negative. If $a \notin y$, then by $x \cap y \subseteq^- x$ it must be negative. Thus $x \supseteq^- (x \cap z)$ as desired – the other inclusion is similar. \square

If $x \sqsubseteq_A y$ then intuitively y has more output for less input. This is analogous to, and in special cases coincides with, the order on functions in domain theory; hence the name “Scott order”. In summary, configurations of $\mathbb{C}_A \otimes S$ correspond to pairs $(x, y) \in \mathcal{C}(S) \times \mathcal{C}(A)$ with $y \sqsubseteq_A \sigma x$.

3.3.3. Step #3: Discrete fibrations. Since configurations of $\mathbb{C}_A \otimes S$ can be elegantly expressed using the Scott order, it will be key to our proof that strategies satisfy a *discrete fibration property* with respect to it. We first recall:

Definition 3.12 (Discrete fibration). Let (X, \leq_X) and (Y, \leq_Y) be orders and $f : X \rightarrow Y$ be a monotonic map. It is a **discrete fibration** when for all $x \in X, y \in Y$ such that $y \leq_Y fx$ there exists a unique $x' \leq_X x \in X$ such that $fx' = y$.

Now, we prove the following characterisation of courtesy and receptivity.

Lemma 3.13. Let $\sigma : S \rightarrow A$ be a pre-strategy. The following are equivalent:

- (i) σ is courteous and receptive,
- (ii) $\sigma : (\mathcal{C}(S), \supseteq^-) \rightarrow (\mathcal{C}(A), \supseteq^-)$ and $\sigma : (\mathcal{C}(S), \sqsubseteq^+) \rightarrow (\mathcal{C}(A), \sqsubseteq^+)$ are discrete fibrations,
- (iii) $\sigma : (\mathcal{C}(S), \sqsubseteq_S) \rightarrow (\mathcal{C}(A), \sqsubseteq_A)$ is a discrete fibration.

Proof. • (iii) \Rightarrow (ii) is straightforward.

- (ii) \Rightarrow (i): *Courtesy.* If $s_1^+ \rightarrow s_2$ in S , then by using the discrete fibration property for \sqsubseteq^+ we prove $\sigma s_1 \leq \sigma s_2$ (hence $\sigma s_1 \rightarrow \sigma s_2$ by Lemma 2.13). Indeed if it is not the case, then σs_1 and σs_2 are concurrent in A – otherwise we would have $\sigma s_2 \leq \sigma s_1$, so $s_2 \leq s_1$ by Lemma 2.13, absurd.

Hence $\sigma[s_2] \setminus \{\sigma s_1\}$ is a configuration of A that positively extends to $\sigma[s_2]$. Thus $[s_2]$ should be the positive extension of a configuration x whose image in the game is $\sigma[s_2] \setminus \{\sigma s_1\}$. By local injectivity, $\sigma s_1 \neq \sigma s_2$, therefore $\sigma s_2 \in \sigma[s_2] \setminus \{\sigma s_1\}$. By local injectivity again, this implies that $s_2 \in x$, so $s_1 \in x$ by down-closure, so $\sigma s_1 \in \sigma[s_2] \setminus \{\sigma s_1\}$, absurd.

If $s_1 \rightarrow s_2^-$, the only case not already covered by the above is that of $s_1^- \rightarrow s_2^-$. Assume σs_1 and σs_2 are concurrent in A . Set $x = [s_2] \setminus \{s_1, s_2\} \in \mathcal{C}(S)$. We have $\sigma x \sqsubseteq^- \sigma x \cup \{\sigma s_2\}$, so by existence of the discrete fibration property there is $x \sqsubseteq x \cup \{s'_2\} \in \mathcal{C}(S)$ and $\sigma s'_2 = \sigma s_2$. But likewise, $\sigma(x \cup \{s'_2\})$ extends in A with σs_1 , so by existence of the discrete fibration property there is s'_1 such that $\sigma s'_1 = \sigma s_1$ and $x \cup \{s'_1, s'_2\} \in \mathcal{C}(S)$. But then by uniqueness of the discrete fibration property we have $x \cup \{s_1, s_2\} = x \cup \{s'_1, s'_2\}$ so by local injectivity $s_1 = s'_1$ and $s_2 = s'_2$, contradicting $s_1 \rightarrow s_2$ since $x \cup \{s'_2\} \in \mathcal{C}(S)$.

Receptivity. This is just an instance of the fibration property for \supseteq^- for atomic extensions.

- (i) \Rightarrow (iii): Let $x \in \mathcal{C}(S)$ and $y \in \mathcal{C}(A)$ such that $y \sqsubseteq \sigma x$.

Uniqueness. We prove by induction on the cardinal of $y \in \mathcal{C}(A)$, that for all $x_1, x_2 \in \mathcal{C}(S)$, if $x_i \sqsubseteq x$ and $\sigma x_i = y$, then $x_1 = x_2$. Assume the result for all $y' \in \mathcal{C}(A)$ strictly smaller than a fixed $y \in \mathcal{C}(A)$.

First, we prove that x_1 and x_2 have the same positive events. Indeed if $s_1 \in x_1$ is positive, then by $\sigma x_1 = y = \sigma x_2$ there is a (unique) $s_2 \in x_2$ such that $\sigma s_1 = \sigma s_2$. Since $x_i \sqsubseteq x$, s_1 and s_2 are in x , and by local injectivity implies $s_1 = s_2$.

If all maximal events of x_1 and x_2 are positive, we are done by down-closure. Otherwise one of them has a negative maximal event, say *wlog.* $s_1 \in x_1$. Since $\sigma x_1 = \sigma x_2$ there is a unique $s_2 \in x_2$ such that $\sigma s_1 = \sigma s_2$.

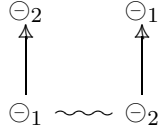
If there exists $s'_2 \in x_2$ with $s_2 \rightarrow s'_2$, since σs_2 is maximal in $\sigma x_1 = \sigma x_2$ (from Lemma 2.13, σ reflects causality), by courtesy we must have s'_2 positive, and hence $s'_2 \in x_1$. It follows that s_1, s_2 are consistent (both in x_1). Hence $s_1 = s_2$, and $s_1 \rightarrow s'_2 \in x_1$, which is absurd. Therefore s_2 is maximal in x_2 .

This entails that $x_1 \setminus \{s_1\}$ and $x_2 \setminus \{s_2\}$ are configurations of S to which we can apply the induction hypothesis for the smaller $y' := y \setminus \{\sigma s_1\}$: the configurations

$x_1 \setminus \{s_1\}$ and $x_2 \setminus \{s_2\}$ must be equal. Since $\sigma s_1 = \sigma s_2$ is a negative extension of $\sigma x_1 \setminus \{\sigma s_1\}$, by receptivity it follows that $s_1 = s_2$.

Existence. By induction on \sqsubset , the irreflexive version of \sqsubseteq (by splitting it into atomic extensions). If $y \xrightarrow{a^+} \sigma x$, write s for the preimage of a in x . If s is not maximal in x , it means that there exists $s \rightarrow s'$ in x . By courtesy since s is positive, we have $\sigma s \rightarrow \sigma s'$ in σx . Hence a is not maximal in σx which is absurd. If $\sigma x \xrightarrow{a^-} y$, it is a consequence of receptivity. \square

Note that for a pre-strategy $\sigma : S \rightarrow A$ it is not equivalent to be receptive and to be a discrete fibration $(\mathcal{C}(S), \supseteq^-) \rightarrow (\mathcal{C}(A), \supseteq^-)$, as demonstrated by the following pre-strategy on the game $A = \ominus_1 \ominus_2$:



This pre-strategy is receptive but not a discrete fibration for \supseteq^- . Indeed, for $x = \emptyset$, $y = \{\ominus_1, \ominus_2\}$ there are two possible matching extensions $x \sqsubseteq x'$. This pre-strategy fails courtesy – the equivalence only holds on courteous pre-strategies.

Putting together the description of configurations of the interaction with copycat of Lemma 3.9, the characterisation of configurations of copycat in Lemma 3.11 and the discrete fibration property above, we get the first direction of the isomorphism between $\mathcal{C}(S)$ and $\mathcal{C}(\mathbb{C}_A \odot S)$.

Proposition 3.14. Let $\sigma : S \rightarrow A$ be receptive and courteous. The discrete fibration property yields a function:

$$L_\sigma : \mathcal{C}(\mathbb{C}_A \odot S) \rightarrow \mathcal{C}(S)$$

commuting with the projection to A (i.e. for all $x \in \mathcal{C}(\mathbb{C}_A \odot S)$, $\sigma(L_\sigma x) = (\mathbb{C}_A \odot \sigma)x$).

Proof. Take $x \in \mathcal{C}(\mathbb{C}_A \odot S)$, yielding $[x]_{\mathbb{C}_A \otimes S} \in \mathcal{C}(\mathbb{C}_A \otimes S)$. In turn, we get

$$\Psi([x]_{\mathbb{C}_A \otimes S}) = (y_S, y_A) \in \mathcal{C}(S) \times \mathcal{C}(A)$$

such that $y_A \sqsubseteq_A \sigma y_S$. But then, by the discrete fibration property, there is a unique $z_S \sqsubseteq_S y_S \in \mathcal{C}(S)$ such that $\sigma z_S = y_A$; and we set $L_\sigma(x) = z_S$. \square

We still have to prove that L_σ preserves inclusion. To establish that, we need first to characterise how the order on $\mathcal{C}(\mathbb{C}_A \odot S)$, and in particular the covering relation, is reflected on interaction witnesses through Ψ .

Lemma 3.15. Let $x^1, x^2 \in \mathcal{C}(\mathbb{C}_A \odot S)$ and let $(y_S^1, y_A^1), (y_S^2, y_A^2)$ be the representations of minimal witnesses (e.g., $(y_S^1, y_A^1) = \Psi([x^1]_{\mathbb{C}_A \otimes S})$). The following are equivalent:

$$\begin{aligned} x^1 \xrightarrow{+} \sqsubset x^2 \text{ in } \mathbb{C}_A \odot S &\Leftrightarrow y_S^1 \sqsubseteq y_S^2 \ \& \ y_A^1 \xrightarrow{+} \sqsubset y_A^2 \\ x^1 \xrightarrow{-} \sqsubset x^2 \text{ in } \mathbb{C}_A \odot S &\Leftrightarrow y_S^1 = y_S^2 \ \& \ y_A^1 \xrightarrow{-} \sqsubset y_A^2 \end{aligned}$$

Proof. Positive extension. Assume $x^1 \overset{+}{\dashv} x^2$. Then $(\mathbb{C}_A \odot \sigma) x^1 \overset{+}{\dashv} (\mathbb{C}_A \odot \sigma) x^2$ implying $y_A^1 \dashv y_A^2$. Moreover, we have $[x^1]_{\mathbb{C}_A \otimes S} \subseteq [x^2]_{\mathbb{C}_A \otimes S}$ implying $y_S^1 \subseteq y_S^2$.

Conversely, we have $y_A^1 \sqsubseteq_A y_A^2 \sqsubseteq_A \sigma y_S^2$ by hypothesis. Hence $(y_S^2, y_A^1) \in \Psi(\mathbb{C}_A \otimes S)$. Writing \sqsubseteq^0 for extension by invisible events in $\mathbb{C}_A \otimes S$, we have:

$$[x^1]_{\mathbb{C}_A \otimes S} = \Psi^{-1}(y_S^1, y_A^1) \sqsubseteq^0 \Psi^{-1}(y_S^2, y_A^1) \overset{+}{\dashv} \Psi^{-1}(y_S^2, y_A^2) = [x^2]_{\mathbb{C}_A \otimes S}$$

Hence $x^1 \overset{+}{\dashv} x^2$ as desired.

Negative extension. If $x^1 \overset{-}{\dashv} x^2$, then we have $y_A^1 \overset{-}{\dashv} y_A^2$ and $y_S^1 \subseteq y_S^2$ by the same argument as in the previous equivalence.

Assume there were a $s \in y_S^2 \setminus y_S^1$. Without loss of generality s can be assumed maximal in y_S^2 . By Lemma 3.10, s is positive and $\sigma s \in y_A^2$. If we had $\sigma s \in y_A^1$, then we would have $\sigma s \in \sigma y_S^1$ as well as $y_A^1 \sqsubseteq_A \sigma y_S^1$; so $s \in y_S^1$ (by local injectivity), absurd. So, $s \in y_A^2 \setminus y_A^1$, contradicting its positivity. Therefore, $y_S^1 = y_S^2$ as desired.

Conversely, if $y_S^1 = y_S^2$ and $y_A^1 \overset{-}{\dashv} y_A^2$ then we have this extension in $\mathbb{C}_A \otimes S$:

$$[x^1]_{\mathbb{C}_A \otimes S} = \Psi^{-1}(y_S^1, y_A^1) \overset{-}{\dashv} \Psi^{-1}(y_S^1, y_A^2) = [x^2]_{\mathbb{C}_A \otimes S}$$

yielding $x^1 \overset{-}{\dashv} x^2$ in $\mathbb{C}_A \odot S$, since the event we added is visible. \square

From that we easily get:

Proposition 3.16. Let, $\sigma : S \rightarrow A$ be receptive and courteous, then the function $L_\sigma : \mathcal{C}(\mathbb{C}_A \odot S) \rightarrow \mathcal{C}(S)$ is monotonic.

Proof. We prove it for coverings. If $x^1, x^2 \in \mathcal{C}(S)$ are such that $x^1 \dashv x^2$, we write $(y_S^1, y_A^1) = \Psi([x^1]_{\mathbb{C}_A \otimes S})$ and $(y_S^2, y_A^2) = \Psi([x^2]_{\mathbb{C}_A \otimes S})$ for the representations of their minimal interaction witnesses.

We distinguish two cases, depending on the polarity of the extension. If $x^1 \overset{-}{\dashv} x^2$, then by the lemma above $y_S^1 = y_S^2$ and $y_A^1 \overset{-}{\dashv} y_A^2$. It immediately follows that $L_\sigma(x^1) \overset{-}{\dashv} L_\sigma(x^2)$ by uniqueness of the discrete fibration property. If $x^1 \overset{+}{\dashv} x^2$, then $y_S^1 \subseteq y_S^2$ and $y_A^1 \overset{+}{\dashv} y_A^2$. But then by Lemma 3.10 we actually have $\sigma y_S^1 \subseteq^- y_A^1$ and $\sigma y_S^2 \subseteq^- y_A^2$, from which it follows that $y_S^1 \overset{+}{\dashv} y_S^2$ as well. And then again, $L_\sigma(x^1) \overset{+}{\dashv} L_\sigma(x^2)$ follows directly from uniqueness of the discrete fibration property. \square

3.3.4. Step #4: Reconstructing minimal interactions. Reciprocally, from $x \in \mathcal{C}(S)$, we have seen that the pair $(x, \sigma x)$ represents a configuration in $\mathbb{C}_A \otimes S$ that gives us a configuration of the composition through hiding. But it might not be the minimal witness, *i.e.* it might not satisfy the conditions of Lemma 3.10.

In order to prove the desired isomorphism, we need to extract from x a x^* such that $(x^*, \sigma x)$ satisfies these conditions. The configuration x^* is obtained by stripping all the maximal negative events away from x , as detailed now.

Lemma 3.17. Let $x \in \mathcal{C}(S)$. There is a unique $x^* \subseteq x \in \mathcal{C}(S)$ such that $\Psi^{-1}(x^*, \sigma x) \in \mathcal{C}(\mathbb{C}_A \otimes S)$ and all maximal events of $\Psi^{-1}(x^*, \sigma x)$ are visible. Restricting to visible events,

this yields a monotonic function

$$\begin{aligned} R_\sigma &: \mathcal{C}(S) \rightarrow \mathcal{C}(\mathbb{C}_A \odot S) \\ x &\mapsto (\Psi^{-1}(x^*, \sigma x)) \cap (\mathbb{C}_A \odot S) \end{aligned}$$

Proof. Uniqueness. Assume we have two x'_1 and x'_2 in $\mathcal{C}(S)$ satisfying the hypotheses. The configurations $\Psi^{-1}(x'_1, \sigma x) \in \mathcal{C}(\mathbb{C}_A \otimes S)$ and $\Psi^{-1}(x'_2, \sigma x) \in \mathcal{C}(\mathbb{C}_A \otimes S)$ correspond to secured bijections:

$$x'_1 \parallel \sigma x \stackrel{\varphi_1}{\simeq} \sigma x'_1 \parallel \sigma x \quad x'_2 \parallel \sigma x \stackrel{\varphi_2}{\simeq} \sigma x'_2 \parallel \sigma x$$

whose maximal events are visible, and $\sigma x \sqsubseteq \sigma x'_1, \sigma x \sqsubseteq \sigma x'_2$.

By Lemma 3.10, the maximal events of x'_1 and x'_2 are positive. Moreover, we have $\sigma x'_1 \subseteq^- \sigma x$. Indeed, we already know that $\sigma x'_1 \supseteq^+ \cdot \subseteq^- \sigma x$, and for $a^+ \in \sigma x$, we have $(0, a) \leq (1, a) \in \mathbb{C}_A$. So, there is $((0, s), (0, a)) \in \varphi_1$. Therefore, $a = \sigma s \in \sigma x'_1$. With these two remarks, it is elementary to check (using $x'_1 \subseteq x$ and local injectivity) that $x'_1 = [x^+]$, where x^+ denotes the set of positive events of x – the same reasoning holds for x'_2 , and hence $x'_1 = x'_2$.

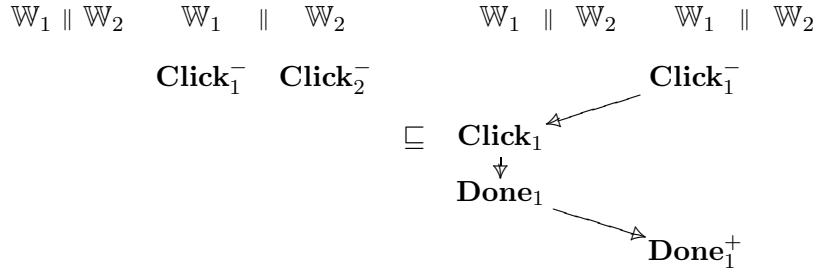
Existence. Write $x^* = [x^+]_S$. The set $x \setminus x^*$ contains all the negative events of x without any positive event above them, thus we have $x^* \subseteq^- x$. Thus $\sigma x \sqsubseteq \sigma x^*$, therefore $\Psi^{-1}(x^*, \sigma x) \in \mathcal{C}(\mathbb{C}_A \otimes S)$. Maximal events are visible because x^* and σx satisfy the condition (ii) of Lemma 3.10.

From the definition of $(-)^*$ above, the monotonicity of R_σ is clear. \square

3.3.5. Step #5: Characterising the Scott order on $\mathcal{C}(\mathbb{C}_A \odot S)$. Using the above, we can prove that indeed receptivity and courtesy are sufficient to be preserved by composition with copycat (proof forthcoming in Theorem 3.20). For necessity, we will prove that strategies obtained by composition with copycat are automatically discrete fibrations. In order to do that, we first need to study the Scott order on $\mathcal{C}(\mathbb{C}_A \odot S)$ (we write V for the set of visible events of $\mathbb{C}_A \otimes S$, that is, the events of $\mathbb{C}_A \odot S$).

As we have seen, configurations of $\mathbb{C}_A \odot S$ correspond to certain pairs $\Psi(z) = (x, y) \in \mathcal{C}(S) \times \mathcal{C}(A)$ where the maximal events of x are positive. Progressing in $\sqsubseteq_{\mathbb{C}_A \odot S}$ means removing some (maximal) negative events from y , and adding some positives to it. The first part is easy, as these events had not been propagated to x yet. However, adding some positives in y might require to replay them first in x , along with their negative dependencies. For instance:

Example 3.18. Consider $A = \mathbb{W}_1 \parallel \mathbb{W}_2$ and σ playing on A , with event structure $\mathbf{Click}_1^- \rightarrow \mathbf{Done}_1^+$ and concurrent \mathbf{Click}_2^- . The two interactions below are minimal witnesses of (respectively) $x_1, x_2 \in \mathcal{C}(\mathbb{C}_A \odot S)$, with $x_1 \sqsubseteq_{\mathbb{C}_A \odot S} x_2$:



We observe that although the visible part progresses *w.r.t.* the Scott order, the invisible part only gains events, and potentially of both polarities: it progresses *w.r.t.* plain inclusion.

Formally, we prove the following lemma.

Lemma 3.19. Let $z, z' \in \mathcal{C}(\mathbb{C}_A \odot S)$ and let $(x, y), (x', y')$ be the respective representations of their minimal witnesses via Ψ . The following are equivalent:

- (1) $z \sqsubseteq_{\mathbb{C}_A \odot S} z'$
- (2) $y \sqsubseteq_A y'$ and $x \subseteq x'$

Proof. Immediate consequence of Lemma 3.15. □

3.3.6. *Step #6: Wrapping up.* Having introduced all the tools and lemmas needed for our proof, we now prove the main theorem.

Theorem 3.20. Let $\sigma : S \rightarrow A$ be a pre-strategy. The following are equivalent:

- (i) σ is a strategy
- (ii) $\sigma : (\mathcal{C}(S), \supseteq^-) \rightarrow (\mathcal{C}(A), \supseteq^-)$ and $\sigma : (\mathcal{C}(S), \subseteq^+) \rightarrow (\mathcal{C}(A), \subseteq^+)$ are discrete fibrations,
- (iii) the map $\sigma : (\mathcal{C}(S), \sqsubseteq_S) \rightarrow (\mathcal{C}(A), \sqsubseteq_A)$ is a discrete fibration
- (iv) σ is courteous and receptive

Proof. The equivalence between (ii), (iii), (iv) is proved by Lemma 3.13.

- (i) \Rightarrow (iii): Let $f : \sigma \cong \mathbb{C}_A \odot \sigma$ be an isomorphism of strategies. Let $(x, y) \in \mathcal{C}(S) \times \mathcal{C}(A)$ with $y \sqsubseteq \sigma x$. Write $\Psi([f(x)]_{\mathbb{C}_A \otimes S}) = (w, \sigma x) \in \Psi(\mathbb{C}_A \otimes S)$ with $w \in \mathcal{C}(S)$ and $\sigma x \sqsubseteq \sigma w$.

Existence. Consider $x_0 = [\{s \in w \mid \sigma s \in y\}]^*$ (Lemma 3.17). By definition the maximal events of $\Psi^{-1}(x_0, y)$ are all visible. Hence (x_0, y) corresponds to a configuration $z \in \mathcal{C}(\mathbb{C}_A \odot S)$. Applying f^{-1} we get a configuration $x' \in \mathcal{C}(S)$ whose image by σ is y . Since $y \sqsubseteq \sigma x$ and $x_0 \subseteq w$, we have by Lemma 3.19, $z \sqsubseteq f(x)$. Hence $x' \sqsubseteq x$ (f^{-1} preserves the Scott order).

Uniqueness. Assume we have two x'_1 and x'_2 satisfying $x'_i \sqsubseteq x$ and $\sigma x'_i = y$. We have $f(x'_1) = V \cap (\Psi(x'_1, y))$ and $f(x'_2) = V \cap (\Psi(x'_2, y))$ for some configurations x''_1 and x''_2 . Applying Lemma 3.17 we get $x''_1 = x''_2$ which yields $f(x'_1) = f(x'_2)$ and then $x'_1 = x'_2$ by injectivity of f .

- (iii) \Rightarrow (i): We have constructed two inclusion-preserving maps $L_\sigma : \mathcal{C}(\mathbb{C}_A \odot S) \rightarrow \mathcal{C}(S)$ and $R_\sigma : \mathcal{C}(S) \rightarrow \mathcal{C}(\mathbb{C}_A \odot S)$. By construction, they are inverses – $L_\sigma \circ R_\sigma = \text{id}_{\mathcal{C}(S)}$ by uniqueness of the discrete fibration property, and $R_\sigma \circ L_\sigma = \text{id}_{\mathcal{C}(\mathbb{C}_A \odot S)}$ by uniqueness of the $(-)^*$ operation in Lemma 3.17. By Lemma 2.17, this yields the desired isomorphism between $\mathbb{C}_A \odot \sigma$ and σ . □

4. THE BICATEGORY OF CONCURRENT GAMES

We have developed a notion of concurrent strategies, and characterised those which behave well in an asynchronous, distributed world. For this to serve as a basis for the compositional semantics of concurrent processes or programs, it is of paramount importance to study the categorical structure of strategies, *i.e.* the algebraic laws satisfied by composition.

Usually – as described first by Joyal on Conway games [Joy77] – composition of strategies yields a category having games as objects, strategies as morphisms and copycat strategies as identities. Here however, we cannot use equality to compare strategies. Indeed, take $\sigma : S \rightarrow A$ and $\sigma' : S' \rightarrow A$ two strategies on A . As we have observed in Section 2, comparing them requires us first to relate S and S' , which we do via a map $f : S \rightarrow S'$ making the obvious triangle commute. This map is in general not unique: we saw below Definition 3.4 a strategy with two automorphisms.

For many purposes, the exact identity of an isomorphism relating two strategies is irrelevant, and in these cases we can (and later will) quotient to a category. This quotient, and the investigation of its further structure, will be carried out in Section 5. But the un-quotiented structure also matters – when working with our games, one is often led to reason on representatives rather than isomorphism classes (for instance when computing infinite strategies as limits of ω -chains of finite strategies). Similarly, further developments in this framework (beyond this paper) rely on properties of composition that the quotiented category is too rough to convey. So we first investigate the composition operation without quotienting, and show how the specific isomorphisms between strategies fit in the categorical picture. This is the purpose of this section, where we will establish that games, strategies and maps between them form a bicategory. We will first review the definition of a bicategory. Then, we will turn to the construction of our concrete bicategory of concurrent games.

4.1. Bicategories. First, recall that a **bicategory** \mathcal{C} consists of the following *basic data* (with notations inspired from our concrete bicategory of concurrent games):

- A set of *objects*, or 0-cells (we use A, B, C, \dots to range over objects).
- For any two objects A, B , a category $\mathcal{C}(A, B)$. Its objects are the *morphisms* or 1-cells of \mathcal{C} (we use σ, τ, \dots to range over morphisms, and write *e.g.* $\sigma : A \multimap B$), and its morphisms are the *2-cells* of \mathcal{C} (we range f, g, \dots to range over 2-cells, and write *e.g.* $f : \sigma \Rightarrow \tau$).
- For each object A , a distinguished morphism $\mathfrak{c}_A : A \multimap A$, called the *identity*.
- For each objects A, B, C , a functor:

$$\odot : \mathcal{C}(B, C) \times \mathcal{C}(A, B) \rightarrow \mathcal{C}(A, C)$$

The functor \odot gives the *composition* $\tau \odot \sigma : A \multimap C$ of 1-cells $\sigma : A \multimap B$ and $\tau : B \multimap C$, but its functorial action also allows us to transport 2-cells alongside compositions. For instance, if $f : \sigma \Rightarrow \sigma'$, then $\tau \odot f : \tau \odot \sigma \Rightarrow \tau \odot \sigma'$ (A consequence of that is that isomorphism of 1-cells is a congruence, *i.e.* is preserved under composition).

But that is not all. In a bicategory, the associativity of \odot and neutrality of \mathfrak{c}_A do not in general hold in the strict sense (or we would have a 2-category), but only up to *coherent isomorphisms*. This means that we have the following *isomorphisms*:

- For any $\sigma : A \multimap B, \tau : B \multimap C, \rho : C \multimap D$, an isomorphism (the *associator*):

$$\alpha_{\sigma, \tau, \rho} : (\rho \odot \tau) \odot \sigma \Rightarrow \rho \odot (\tau \odot \sigma)$$

natural in σ, ρ and τ .

- For any $\sigma : A \multimap B$, two isomorphisms (the *unitors*):

$$\rho_\sigma : \sigma \odot \mathfrak{c}_A \Rightarrow \sigma \qquad \lambda_\sigma : \mathfrak{c}_B \odot \sigma \Rightarrow \sigma$$

natural in σ .

Finally, these data need to satisfy some coherence conditions: the associators are subject to Mac Lane's *pentagon identity* whereas the unitors must satisfy the *triangle identity* (those are the same as in a monoidal category). We do not recall them now, but we will state them in the course of the construction of our concrete bicategory.

We now go on to construct our concrete bicategory. We gave the definition of a bicategory in two steps: first what we called the *basic data*, and then natural isomorphisms for associativity and unities, subject to coherence conditions. Our construction of the concrete bicategory will follow the same lines.

4.2. Basic data of the bicategory CG. As expected, the *objects* of CG are the games, and the *morphisms* from A to B are the strategies $\sigma : S \rightarrow A^\perp \parallel B$. As in the definition above we will occasionally write $\sigma : A \dashrightarrow B$, keeping the S anonymous.

For $\sigma : S \rightarrow A^\perp \parallel B$, $\tau : T \rightarrow B^\perp \parallel C$, a 2-cell $f : \sigma \Rightarrow \tau$ is a map of esps $f : S \rightarrow T$ making the following triangle commute:

$$\begin{array}{ccc} S & \xrightarrow{f} & T \\ \sigma \searrow & & \swarrow \tau \\ & A^\perp \parallel B & \end{array}$$

Such 2-cells can be composed as functions, and for $\sigma : S \rightarrow A^\perp \parallel B$ the identity $\text{id}_S : S \rightarrow S$ is a valid 2-cell $\text{id}_\sigma : \sigma \Rightarrow \sigma$. Therefore, for any two games A and B we get a category $\text{CG}(A, B)$ as required by the definition.

4.2.1. Functorial composition. Now, we need a functor:

$$\odot : \text{CG}(B, C) \times \text{CG}(A, B) \rightarrow \text{CG}(A, C)$$

For $\tau : B \dashrightarrow C$ and $\sigma : A \dashrightarrow B$, its action is the composition $\tau \odot \sigma$ as in Section 2. This operation was defined on pre-strategies rather than strategies, so we note in passing:

Proposition 4.1. For $\sigma : S \rightarrow A^\perp \parallel B$ and $\tau : T \rightarrow B^\perp \parallel C$ strategies, $\tau \odot \sigma$ is a strategy.

Proof. We use the second formulation of the definition of strategies, as in Theorem 3.20.

Negative fibration. Take $x \in \mathcal{C}(T \odot S)$ such that $(\tau \odot \sigma)(x) \subseteq^- x'_A \parallel x'_C$ for some $x'_A \parallel x'_C \in \mathcal{C}(A^\perp \parallel C)$. By definition, its down-closure in $T \otimes S$ is a configuration $y = [x] \in \mathcal{C}(T \otimes S)$, whose maximal elements are visible. By Lemma 2.12, this configuration is represented by (the graph of) a secured bijection $\varphi \in \mathcal{B}_{\sigma \parallel C, A \parallel \tau}^{\text{sec}}$. We write:

$$y_S \parallel y_C \stackrel{\varphi}{\simeq} y_A \parallel y_T$$

with $\sigma y_S = y_A \parallel y_B$ and $\tau y_T = y_B \parallel y_C$. By hypothesis we have $y_A \parallel y_B \subseteq^- y'_A \parallel y_B$, and $y_B \parallel y_C \subseteq^- y_B \parallel y'_C$ for some $y'_A \in \mathcal{C}(A^\perp)$ and $y'_C \in \mathcal{C}(C)$. Since σ and τ are strategies, there are unique $y_S \subseteq y'_S \in \mathcal{C}(S)$ and $y_T \subseteq y'_T \in \mathcal{C}(T)$ such that $\sigma y'_S = y'_A \parallel y_B$ and $\tau y'_T = y_B \parallel y'_C$. The induced extension of φ

$$y'_S \parallel y'_C \stackrel{\varphi'}{\simeq} y'_A \parallel y'_T$$

is secured: the added events only map to A and C , so there is no interaction (hence potential deadlock) between σ and τ going on. Moreover, φ' represents a configuration

$y \subseteq y' \in \mathcal{C}(T \otimes S)$, which maps to $x'_A \parallel x_B \parallel x'_C$. By projection we get the required extension of x . Uniqueness follows directly from uniqueness for y'_S and y'_T .

Positive fibration. Similar reasoning. \square

So composition, despite being defined on pre-strategies rather than strategies, preserves courtesy and receptivity – it is well-defined on 1-cells of our bicategory. We now need to prove that it is well-defined on 2-cells as well. In fact, we will show that it is well-defined on morphisms between arbitrary pre-strategies, not only those that are receptive and courteous. Until Section 4.4 (where we study compositions with copycat), the development will use neither receptivity nor courtesy.

Let $\sigma : S \rightarrow A^\perp \parallel B$, $\sigma' : S' \rightarrow A^\perp \parallel B$ and $\tau : T \rightarrow B^\perp \parallel C$ be pre-strategies, and $f : S \rightarrow S'$ be a morphism from σ to σ' . We proved in Lemma 2.14 that the interaction $T \otimes S$ was the pullback of $\sigma \parallel C$ and $A \parallel \tau$. By the corresponding universal property, it follows that there is a unique map $f \otimes T : S \otimes T \rightarrow S' \otimes T$ making the required diagrams commute. In particular, this remark establishes that the interaction operation $- \otimes -$ is functorial in morphisms between pre-strategies. In order for \odot to inherit this, it is convenient to use that \otimes and \odot are related by a universal property involving *partial maps*:

Definition 4.2. A **partial map** of es(p)s $f : E \rightarrow F$ is a partial function, such that for all $x \in \mathcal{C}(E)$ we have $fx \in \mathcal{C}(F)$, and such that for all $e_1, e_2 \in x \in \mathcal{C}(E)$, if $fe_1 = fe_2$ (with both defined), then $e_1 = e_2$.

A key example of a partial map in our setting, is the *hiding map*: given an es(p) E and $V \subseteq E$, there is a partial map:

$$\mathfrak{h} : E \rightarrow E \downarrow V$$

acting as the identity on V and undefined otherwise. So in particular, for pre-strategies $\sigma : S \rightarrow A^\perp \parallel B$ and $\tau : T \rightarrow B^\perp \parallel C$, there is a partial map:

$$\mathfrak{h} : T \otimes S \rightarrow T \odot S.$$

Projection and hiding provide a partial-total factorization system, which obeys:

Lemma 4.3. Let $f : E \rightarrow F$ be a partial map of es(p)s, and V be the subset of events of E on which f is defined. Then, f factors as $(f \upharpoonright V) \circ \mathfrak{h}$ (where $f \upharpoonright V : E \downarrow V \rightarrow F$ is total). Moreover, for any other factorisation $f = g_2 \circ g_1$ with $g_1 : E \rightarrow X$ and $g_2 : X \rightarrow F$, there is a unique total $h : E \downarrow V \rightarrow X$ such that $h \circ \mathfrak{h} = g_1$ and $g_2 \circ h = f \upharpoonright V$, as pictured in the diagram below:

$$\begin{array}{ccccc}
 & E & & & \\
 & \downarrow \mathfrak{h} & \searrow g_1 & & \\
 & E \downarrow V & \xrightarrow{\dots} h & \rightarrow & X \\
 & \downarrow f \upharpoonright V & & \swarrow g_2 & \\
 & F & & &
 \end{array}$$

f is indicated by a curved arrow from E to F .

We say that $\mathfrak{h} : E \rightarrow E \downarrow V$ has the **partial-total universal property**.

Proof. Direct verification. \square

From that, it is easy to construct the functorial action of \odot . Take σ, σ', τ and f as above. As explained, we obtain $T \otimes f : T \otimes S \rightarrow T \otimes S'$ by the universal property of the interaction pullback.

But by Lemma 4.3, the two maps $\mathfrak{h}_{\sigma, \tau} : T \otimes S \rightarrow T \odot S$ and $\mathfrak{h}_{\sigma', \tau} : T \otimes S' \rightarrow T \odot S'$ have the partial-total universal property. Using it, we get a unique map $T \odot f : T \odot S \rightarrow T \odot S'$ matching $T \otimes f$ up to hiding. It is straightforward from the universal properties that this operation is functorial, that its symmetric counterparts $g \otimes S$ and $g \odot S$ are as well and that they satisfy the interchange laws, yielding the required bifunctor.

In fact we note in passing that \odot preserves more general notions of morphisms of pre-strategies, that do *not* leave the game invariant:

Lemma 4.4. Consider two commuting diagrams between pre-strategies (using the obvious functorial action of $(-)^{\perp}$ and $- \parallel -$ in \mathcal{EP}):

$$\begin{array}{ccc} S_1 & \xrightarrow{f} & S_2 \\ \sigma_1 \downarrow & & \downarrow \sigma_2 \\ A_1^{\perp} \parallel B_1 & \xrightarrow{h_1^{\perp} \parallel h_2} & A_2^{\perp} \parallel B_2 \end{array} \quad \begin{array}{ccc} T_1 & \xrightarrow{g} & T_2 \\ \tau_1 \downarrow & & \downarrow \tau_2 \\ B_1^{\perp} \parallel C_1 & \xrightarrow{h_2^{\perp} \parallel h_3} & B_2^{\perp} \parallel C_2 \end{array}$$

Then, the following diagram commutes.

$$\begin{array}{ccc} T_1 \odot S_1 & \xrightarrow{g \odot f} & T_2 \odot S_2 \\ \tau_1 \odot \sigma_1 \downarrow & & \downarrow \tau_2 \odot \sigma_2 \\ A_1^{\perp} \parallel C_1 & \xrightarrow{h_1^{\perp} \parallel h_3} & A_2^{\perp} \parallel C_2 \end{array}$$

Proof. For interactions first, the map $g \otimes f : T_1 \otimes S_1 \rightarrow T_2 \otimes S_2$ is defined from the universal property of the pullback for $T_2 \otimes S_2$, using the two commuting diagrams in the hypothesis. It follows by definition that the diagram

$$\begin{array}{ccc} T_1 \otimes S_1 & \xrightarrow{g \otimes f} & T_2 \otimes S_2 \\ \tau_1 \otimes \sigma_1 \downarrow & & \downarrow \tau_2 \otimes \sigma_2 \\ A_1 \parallel B_1 \parallel C_1 & \xrightarrow{h_1 \parallel h_2 \parallel h_3} & A_2 \parallel B_2 \parallel C_2 \end{array}$$

commutes. The map $g \odot f : T_1 \odot S_1 \rightarrow T_2 \odot S_2$ and the required diagram commutation follow from the partial-total universal property. \square

4.3. Associators. We now define the *associator*, i.e. for every three strategies $\sigma : A \rightarrow B$, $\tau : B \rightarrow C$, $\rho : C \rightarrow D$, an isomorphism

$$\alpha_{\sigma, \tau, \rho} : (\rho \odot \tau) \odot \sigma \Rightarrow \rho \odot (\tau \odot \sigma)$$

natural in σ, τ, ρ , and subject to *Mac Lane's pentagon* (detailed in the development below). We will start with the definition of the associator.

4.3.1. *Associativity for interaction.* For the rest of this subsection we only consider polarity-agnostic operations, so we will ignore polarity from now on.

Consider $\sigma : S \rightarrow A \parallel B$, $\tau : T \rightarrow B \parallel C$, and $\rho : U \rightarrow C \parallel D$. The composition $\rho \odot \tau : U \odot T \rightarrow B \parallel D$ is obtained by restriction from the mediating map $\rho \otimes \tau : U \otimes T \rightarrow B \parallel C \parallel D$ of the interaction pullback. In turn, we can form $(\rho \otimes \tau) \otimes \sigma : (U \otimes T) \otimes S \rightarrow A \parallel B \parallel C \parallel D$ as (the mediating map of) the pullback of $\sigma \parallel C \parallel D$ and $A \parallel (\rho \otimes \tau)$. From that (using that pullbacks are stable under parallel composition) it appears that $(\rho \otimes \tau) \otimes \sigma$ is (the mediating map of) a ternary pullback of $\sigma \parallel C \parallel D$, $A \parallel \tau \parallel D$ and $A \parallel B \parallel \rho$. But a similar reasoning holds for $\rho \otimes (\tau \otimes \sigma)$, so by the universal property of pullbacks, there is a unique map $a_{\sigma, \tau, \rho}$, necessarily an isomorphism, making the projections to $\sigma \parallel C \parallel D$, $A \parallel \tau \parallel D$ and $A \parallel B \parallel \rho$ commute:

$$\begin{array}{ccc} (U \otimes T) \otimes S & \xrightarrow{a_{\sigma, \tau, \rho}} & U \otimes (T \otimes S) \\ & \searrow_{(\rho \otimes \tau) \otimes \sigma} & \swarrow_{\rho \otimes (\tau \otimes \sigma)} \\ & A \parallel B \parallel C \parallel D & \end{array}$$

Given another $\delta : V \rightarrow D \parallel E$, all bracketings of the quaternary interaction between $\sigma, \tau, \rho, \delta$ can be obtained via pullbacks of $\sigma \parallel C \parallel D \parallel E$, $A \parallel \tau \parallel D \parallel E$, $A \parallel B \parallel \rho \parallel E$ and $A \parallel B \parallel C \parallel \delta$ taken in different orders. It follows from an easy diagram chase that *Mac Lane's pentagon* commutes at the level of interactions:

$$\begin{array}{ccc} & ((V \otimes U) \otimes T) \otimes S & \\ & \swarrow_{a_{\tau, \rho, \delta} \otimes S} & \searrow_{a_{\sigma, \tau, \rho \otimes \delta}} \\ (V \otimes (U \otimes T)) \otimes S & & (V \otimes U) \otimes (T \otimes S) \\ \downarrow_{a_{\sigma, \rho \otimes \tau, \delta}} & & \swarrow_{a_{\tau \otimes \sigma, \rho, \delta}} \\ V \otimes ((U \otimes T) \otimes S) & & \\ & \searrow_{V \otimes a_{\sigma, \tau, \rho}} & \\ & V \otimes (U \otimes (T \otimes S)) & \end{array}$$

To conclude associativity, we need to show how to reproduce the same reasoning on composition, or more adequately deduce it from that on interactions.

4.3.2. *Partial-total factorization and hiding witnesses.* In order to deduce associators on composition and their coherence from those on interactions, we generalize the partial-total universal property of Lemma 4.3 to n -ary interactions and compositions. For instance, we need to prove that the hiding map (to be defined precisely):

$$\mathfrak{h} : (U \otimes T) \otimes S \rightarrow (U \odot T) \odot S$$

has the partial-total universal property. It is rather inconvenient to prove it directly – instead, we prove an auxiliary property that is easier to combine.

Definition 4.5. Let $f : E \rightarrow F$ be a partial map. A **hiding witness** for f is a monotonic function:

$$\text{wit}_f : \mathcal{C}(F) \rightarrow \mathcal{C}(E)$$

such that for all $x \in \mathcal{C}(E)$, $\text{wit}_f \circ f(x) \subseteq x$ and for all $x \in \mathcal{C}(F)$, $f \circ \text{wit}_f(x) = x$.

The hiding witness assigns, to any $x \in \mathcal{C}(F)$, a canonical witness $\text{wit}_f(x) \in \mathcal{C}(E)$, that projects back to x through f . The hiding witnesses give a configuration-based version of projection – or of the partial-total factorization, as established by the lemma below.

Proposition 4.6. Let $f : E \rightarrow F$ be a partial map. Then, the three following propositions are equivalent:

- (i) There exists an isomorphism $\varphi : E \downarrow V \cong F$ such that $\varphi \circ \mathfrak{h} = f$ (where V is the domain of definition of f – note that φ is necessarily f restricted to V),
- (ii) f has the partial-total universal property,
- (iii) f has a hiding witness.

We call **hiding maps** any partial maps satisfying those properties. Note that by (i) it follows that in any hiding map f is partial rigid, *i.e.* for any $e_1 \leq e_2$, if $f(e_1), f(e_2)$ defined then $f(e_1) \leq f(e_2)$.

Proof. (i) \Leftrightarrow (ii). From left to right, we transport through φ the partial-total universal property of Lemma 4.3. From right to left, we use the fact that both $\mathfrak{h} : E \rightarrow E \downarrow V$ and $f : E \rightarrow F$ have the partial-total universal property, yielding the desired isomorphism.

(ii) \Rightarrow (iii). W.l.o.g., we prove it for $\mathfrak{h} : E \rightarrow E \downarrow V$. For $x \in \mathcal{C}(E \downarrow V)$, define $\text{wit}(x) = [x] \in \mathcal{C}(E)$. Clearly, $\mathfrak{h}(\text{wit}(x)) = [x] \cap V = x$ and $\text{wit}(\mathfrak{h}(x)) = [x \cap V] \subseteq x$ as required, and it preserves union by definition.

(iii) \Rightarrow (i). We construct the isomorphism on configurations:

$$\begin{aligned} p & : \mathcal{C}(E \downarrow V) &\rightarrow & \mathcal{C}(F) \\ & x &\mapsto & f([x]) \\ \\ q & : \mathcal{C}(F) &\rightarrow & \mathcal{C}(E \downarrow V) \\ & y &\mapsto & \text{wit}(y) \cap V \end{aligned}$$

It is clear by definition that these maps are monotonic, we need to prove that they are inverses of each other. For one direction, for all $y \in \mathcal{C}(F)$, since $\text{wit}(y) \in \mathcal{C}(E)$ it is down-closed in E and thus can only differ from $[\text{wit}(y) \cap V] \in \mathcal{C}(E)$ with events not in V , so $f([\text{wit}(y) \cap V]) = f(\text{wit}(y)) = y$, *i.e.* $p \circ q(y) = y$.

For the other direction, we note first that if $x \in \mathcal{C}(E)$ has all its maximal events in V , then $\text{wit}(f(x)) = x$. Indeed, we have $\text{wit}(f(x)) \subseteq x$ by hypothesis. But both sides map to $f(x)$ via f , inducing by local injectivity bijections $\text{wit}(f(x)) \cap V \simeq f(x)$ and $x \cap V \simeq f(x)$. It follows that $\text{wit}(f(x)) \cap V = x \cap V$. But $x = [x \cap V]$ since its maximal elements are visible. Putting everything together:

$$x = [x \cap V] = [\text{wit}(f(x)) \cap V] \subseteq \text{wit}(f(x)) \subseteq x$$

So $x = \text{wit}(f(x))$. Turning back to our main proof, we need to show that $q \circ p(x) = x$ for $x \in \mathcal{C}(E \downarrow V)$, *i.e.* that $\text{wit}(f([x])) \cap V = x$. But by definition, $[x]$ has its maximal events in V , so $\text{wit}(f([x])) = [x]$. So we are left to prove that $[x] \cap V = x$, which is clear.

So we have constructed an order-isomorphism between the domains of configurations of $E \downarrow V$ and F , which yields an isomorphism by Lemma 2.17. Finally, the required equality is obvious by Lemma 2.15. \square

4.3.3. *Associators for composition.* The third formulation of hiding maps enables us to combine them in several ways. Firstly, they are stable under composition:

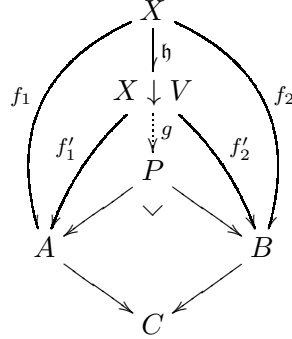
Lemma 4.7. Let $\mathfrak{h} : E_1 \rightarrow E_2$ and $\mathfrak{h}' : E_2 \rightarrow E_3$ be hiding maps, then $\mathfrak{h}' \circ \mathfrak{h} : E_1 \rightarrow E_3$ is a hiding map as well.

Proof. Obvious, by composing the hiding witnesses. \square

We can also combine hiding maps “horizontally”, using the universal property of the interaction. For that though, we need first to prove that this universal property applies to partial maps.

Lemma 4.8. A pullback square in \mathcal{E} is also a pullback square in the category \mathcal{E}_\perp having event structures as objects, and *partial maps* as morphisms.

Proof. The proof is summarized in the following diagram:



Take f_1, f_2 partial maps such that the outer square commutes. Necessarily, f_1 and f_2 are defined on the same subset of events of X ; call it V . By Lemma 4.3, $\mathfrak{h} : X \rightarrow X \downarrow V$ satisfies the partial-total universal property. By the universal property of the pullback in \mathcal{E} , there exists a unique $g : X \downarrow V \rightarrow P$ making the triangle commutes, yielding a factorization $g \circ \mathfrak{h} : X \rightarrow P$. Uniqueness follows directly from the uniqueness of the pullback and of the partial-total universal property. \square

Therefore, we can use the universal property of the interaction pullback to manipulate and compose hiding maps. This allows us to state and prove the lemma below, which plays a similar role to the *zipping lemma* used in proving associativity of composition in sequential games – hence the name.

Lemma 4.9 (Zipping lemma). Let $\mathfrak{h} : S \rightarrow S'$ be a hiding map making the following diagram commute:

$$\begin{array}{ccc} S & \xrightarrow{\mathfrak{h}} & S' \\ \sigma \downarrow & & \downarrow \sigma' \\ A \parallel B \parallel C & \xrightarrow{A \parallel \perp \parallel C} & A \parallel C \end{array}$$

Then, for $\rho : U \rightarrow C \parallel D$, the morphism $U \otimes \mathfrak{h} : U \otimes S \rightarrow U \otimes S'$ defined using the universal property of $U \otimes S'$ via Lemma 4.8 is a hiding map.

Proof. We show that $U \otimes \mathfrak{h}$ has a hiding witness. A configuration of $U \otimes S'$ corresponds to configurations $x_{S'} \parallel x_D$ and $x_A \parallel x_U$ of the event structures as annotated, such that:

$$\begin{aligned}\sigma' x_{S'} &= x_A \parallel x_C \\ \rho x_U &= x_C \parallel x_D\end{aligned}$$

and such that the induced bijection between $x_{S'} \parallel x_D$ and $x_A \parallel x_U$ is secured.

From that, we consider $\text{wit}_{\mathfrak{h}}(x_{S'}) \parallel x_D$ and $x_A \parallel x_B \parallel x_U$, where x_B is obtained by $\sigma(\text{wit}_{\mathfrak{h}}(x_{S'})) = x_A \parallel x_B \parallel x_C$. By construction we have $(\sigma \parallel D)(\text{wit}_{\mathfrak{h}}(x_{S'}) \parallel x_D) = (A \parallel B \parallel \rho)(x_A \parallel x_B \parallel x_U)$. The induced bijection is secured: a causal loop in it could not stay in (events projected to) B , as the causality on the corresponding pairs is entirely determined by S . So, using that \mathfrak{h} is partial rigid by Proposition 4.6 it would induce a causal loop in the original bijection, that was supposed secured. All the additional properties to check follow by construction. \square

At this point, we can define the associator. Recall that for $\sigma : S \rightarrow A \parallel B$, $\tau : T \rightarrow B \parallel C$ and $\rho : U \rightarrow C \parallel D$ we have the associator at the level of interactions:

$$a_{\sigma,\tau,\rho} : (U \otimes T) \otimes S \rightarrow U \otimes (T \otimes S)$$

By using the two lemmas above, we have two hiding maps:

$$\begin{aligned}\mathfrak{h}_{\sigma,(\tau,\rho)} &= (U \otimes T) \otimes S \xrightarrow{\mathfrak{h}_{\tau,\rho} \otimes S} (U \odot T) \otimes S \xrightarrow{\mathfrak{h}_{\sigma,\rho \odot \tau}} (U \odot T) \odot S \\ \mathfrak{h}_{(\sigma,\tau),\rho} &= U \otimes (T \otimes S) \xrightarrow{U \otimes \mathfrak{h}_{\sigma,\tau}} U \otimes (T \odot S) \xrightarrow{\mathfrak{h}_{\tau \odot \sigma, \rho}} U \odot (T \odot S)\end{aligned}$$

From the definitions, it is easy to check that the following outer diagram commutes:

$$\begin{array}{ccc} U \otimes (T \otimes S) & \xrightarrow{a_{\sigma,\tau,\rho}} & (U \otimes T) \otimes S \\ \mathfrak{h}_{\sigma,(\tau,\rho)} \downarrow & & \downarrow \mathfrak{h}_{(\sigma,\tau),\rho} \\ U \odot (T \odot S) & \xrightarrow{\alpha_{\sigma,\tau,\rho}} & (U \odot T) \odot S \\ & \searrow \rho \odot (\tau \odot \sigma) & \swarrow (\rho \odot \tau) \odot \sigma \\ & & A \parallel D \end{array}$$

So by the partial-total universal properties of $\mathfrak{h}_{(\sigma,\tau),\rho}$ and $\mathfrak{h}_{\sigma,(\tau,\rho)}$, $a_{\sigma,\tau,\rho}$ induces a unique isomorphism $\alpha_{\sigma,\tau,\rho} : (U \odot T) \odot S \rightarrow U \odot (T \odot S)$ making the two sub-diagrams commute.

4.3.4. Naturality and coherence. To conclude the associativity part of the bicategory construction, we need to check that these isomorphisms are natural in σ, τ, ρ and satisfy Mac Lane's pentagon. In both cases, the proof consists in verifying it first for interactions (as we already did earlier from the pentagon), and deducing it for composition by checking that the maps involved in the diagram for composition are canonically related to those for interaction, as above. We skip the details, that can be recovered easily.

4.4. **Unitors.** The last ingredients of our bicategory are the two unitors. For any strategy $\sigma : S \rightarrow A^\perp \parallel B$, those are the two isomorphisms for cancellation of copycat:

$$\begin{aligned}\rho_\sigma &= S \odot \mathbb{C}_A \rightarrow S \\ \lambda_\sigma &= \mathbb{C}_B \odot S \rightarrow S\end{aligned}$$

We start by defining λ_σ (and ρ_σ): their definition is not strictly speaking covered by the result of Theorem 3.20 which only dealt with closed compositions of a strategy $\sigma : S \rightarrow A$ with \mathbb{C}_A . However the construction is very similar and will only be roughly sketched here.

Lemma 4.10. Let $\sigma : S \rightarrow A^\perp \parallel B$. Then, there are order-isomorphisms:

$$\begin{aligned}\Psi_r &: \mathcal{C}(S \otimes \mathbb{C}_A) \cong \{(x_A^l, x_S) \in \mathcal{C}(A) \times \mathcal{C}(S) \mid \sigma x_S = x_A^r \parallel x_B \ \& \ x_A^l \sqsupseteq_A x_A^r\} \\ \Psi_l &: \mathcal{C}(\mathbb{C}_B \otimes S) \cong \{(x_S, x_B^r) \in \mathcal{C}(S) \times \mathcal{C}(B) \mid \sigma x_S = x_A \parallel x_B^l \ \& \ x_B^r \sqsubseteq_B x_B^l\}\end{aligned}$$

where the right hand side sets are ordered by componentwise inclusion.

Proof. Straightforward adaptation of Lemma 3.9. \square

At this point, it is also worth mentioning that it follows from courtesy of σ that in a situation like in the lemma above, we actually have $x_B^l \sqsubseteq^- x_B^r$. No positive events can be added by going from x_B^r to x_B^l , as using courtesy one can show that those could not be below a visible events. That fact is not used in our development, so we skip the detailed proof.

We jump to the definition of the unitors:

Lemma 4.11. For any $\sigma : S \rightarrow A^\perp \parallel B$, there are isomorphisms of strategies:

$$\rho_\sigma : S \odot \mathbb{C}_A \rightarrow S \quad \lambda_\sigma : \mathbb{C}_B \odot S \rightarrow S$$

which respectively,

- To any $x \in \mathcal{C}(S \odot \mathbb{C}_A)$ with unique witness $[x] = \Psi_l^{-1}(x_A^l, x_S) \in \mathcal{C}(\mathbb{C}_A \otimes S)$ with $\sigma x_S = x_A^r \parallel x_B$ and $x_A^l \sqsubseteq_{A^\perp} x_A^r$, ρ_σ associates the unique $x'_S \sqsubseteq x_S$ such that $\sigma x'_S = x_A^l \parallel x_B$ given by the discrete fibration property of σ .
- To any $x \in \mathcal{C}(\mathbb{C}_B \odot S)$ with unique witness $[x] = \Psi_r^{-1}(x_S, x_B^r) \in \mathcal{C}(S \otimes \mathbb{C}_B)$ with $\sigma x_S = x_A \parallel x_B^l$ and $x_B^r \sqsubseteq_B x_B^l$, λ_σ associates the unique $x'_S \sqsubseteq x_S$ such that $\sigma x'_S = x_A \parallel x_B^r$.

Proof. Straightforward adaptation of (iii) \Rightarrow (i) in the proof of Theorem 3.20. \square

We now show that the unitors $\lambda_\sigma, \rho_\sigma$ are natural in σ . In fact, it will be helpful later on to prove here a slightly more general property: that the unitors acts naturally with respect to generalized morphisms between strategies, that change the base game as well. In order to state it, first note that the construction $A \mapsto \mathbb{C}_A$ on eps can be easily extended into a functor:

$$\mathbb{C} : \mathcal{EP} \rightarrow \mathcal{EP}$$

Indeed, for $f : A \rightarrow B$ a map of eps, we have $f^\perp \parallel f : A^\perp \parallel A \rightarrow B^\perp \parallel B$ (using the obvious functorial action of $(-)^{\perp}$ and \parallel on \mathcal{EP}). But $A^\perp \parallel A$ and $B^\perp \parallel B$ are respectively the sets of events of \mathbb{C}_A and \mathbb{C}_B ; and it is a simple verification that we do have $\mathbb{C}_f = f^\perp \parallel f : \mathbb{C}_A \rightarrow \mathbb{C}_B$. Functoriality of the construction is clear. Using that, we state and prove the following:

Lemma 4.12. Let $\sigma_1 : S_1 \rightarrow A_1^\perp \parallel B_1$, $\sigma_2 : S_2 \rightarrow A_2^\perp \parallel B_2$, and $f : S_1 \rightarrow S_2$, $h : A_1 \rightarrow A_2$, $h' : B_1 \rightarrow B_2$ such that the following diagram commutes:

$$\begin{array}{ccc} S_1 & \xrightarrow{f} & S_2 \\ \sigma_1 \downarrow & & \downarrow \sigma_2 \\ A_1^\perp \parallel B_1 & \xrightarrow{h^\perp \parallel h'} & A_2^\perp \parallel B_2 \end{array}$$

Then, the following two diagrams commute as well:

$$\begin{array}{ccc} & & A_1^\perp \parallel B_1 \\ & \nearrow \alpha_{B_1} \odot \sigma_1 & \downarrow \sigma_1 \\ \mathbb{C}_{B_1} \odot S_1 & \xrightarrow{\lambda_{\sigma_1}} & S_1 \\ \downarrow \mathbb{C}_{h'} \odot f & & \downarrow f \\ \mathbb{C}_{B_2} \odot S_2 & \xrightarrow{\lambda_{\sigma_2}} & S_2 \\ & \nearrow \alpha_{B_2} \odot \sigma_2 & \downarrow \sigma_2 \\ & & A_2^\perp \parallel B_2 \end{array} \quad \begin{array}{ccc} & & A_1^\perp \parallel B_1 \\ & \nearrow \sigma_1 \odot \alpha_{A_1} & \downarrow \sigma_1 \\ S_1 \odot \mathbb{C}_{A_1} & \xrightarrow{\rho_{\sigma_1}} & S_1 \\ \downarrow f \odot \mathbb{C}_h & & \downarrow f \\ S_2 \odot \mathbb{C}_{A_2} & \xrightarrow{\rho_{\sigma_2}} & S_2 \\ & \nearrow \sigma_2 \odot \alpha_{A_2} & \downarrow \sigma_2 \\ & & A_2^\perp \parallel B_2 \end{array}$$

In particular (when h, h' are identities), λ_σ and ρ_σ are natural in σ .

Proof. Let us focus on the left hand side diagram, the other is symmetric. Of all the faces of the diagram, the right hand side one is by hypothesis, the upper and lower are by definition of unitors in Lemma 4.11, and the left hand side one is by Lemma 4.4. It remains to prove that the front face commutes.

Let $x \in \mathcal{C}(\mathbb{C}_{B_1} \odot S_1)$, with unique witness $[x] = \Psi_r(x_{S_1}, x_{B_1}^r)$, with $\sigma_1 x_{S_1} = x_{A_1} \parallel x_{B_1}^l$ and $x_{B_1}^r \sqsubseteq x_{B_1}^l$. The left unitor λ_{σ_1} sends x to the unique $x'_{S_1} \sqsubseteq x_{S_1}$ such that $\sigma x'_{S_1} = x_{A_1} \parallel x_{B_1}^r$, whereas $\mathbb{C}_{h'} \odot f$ by definition sends it to $(\mathbb{C}_{h'} \odot f)(x)$ with unique witness $\Psi_r(f(x_{S_1}), h'(x_{B_1}^r))$. But then, $f(x'_{S_1}) \sqsubseteq f(x_{S_1})$ is such that $\sigma_2(f(x'_{S_1})) = h(x_{A_1}) \parallel h'(x_{B_1}^r)$, and the unique such (by uniqueness of the discrete fibration property). Therefore, $\lambda_{\sigma_2}((\mathbb{C}_{h'} \odot f)(x)) = f(x'_{S_1})$. \square

And finally, using the description of their action we verify the coherence law for unitors.

Lemma 4.13. For $\sigma : S \rightarrow A^\perp \parallel B$ and $\tau : T \rightarrow B^\perp \parallel C$, the following diagram commutes.

$$\begin{array}{ccc} (T \odot \mathbb{C}_B) \odot S & \xrightarrow{\alpha_{\sigma, \mathbb{C}_B, \tau}} & T \odot (\mathbb{C}_B \odot S) \\ \downarrow \rho_{\tau \odot S} & & \downarrow T \odot \lambda_\sigma \\ & T \odot S & \end{array}$$

Proof. Let $x \in \mathcal{C}((T \odot \mathbb{C}_B) \odot S)$. Necessarily, it has a witness $\text{wit}(x) \in \mathcal{C}((T \otimes \mathbb{C}_B) \otimes S)$. By characterisation of pullbacks, it corresponds to three configurations $x_S \parallel x_B^r \parallel x_C$, $x_A \parallel x_B^l \parallel x_B^r \parallel x_C$, and $x_A \parallel x_B^l \parallel x_T$ such that $\sigma x_S = x_A \parallel x_B^l$, $x_B^r \sqsubseteq x_B^l$ (regarded as configurations of B), and $\tau x_T = x_B^r \parallel x_C$. Moreover, the induced order on triples is secured, and its maximal elements are visible. But this implies that actually $x_B^l = x_B^r$ – it is easy to show that if (non-visible) $b \in x_B^l$ is not in x_B^r , then it cannot be below a visible event. From

that it follows that both paths alongside the triangle above map x to (the configuration of $T \odot S$ represented by) $x_S \parallel x_C$ and $x_A \parallel x_T$. \square

We have finished the proof that CG is a bicategory.

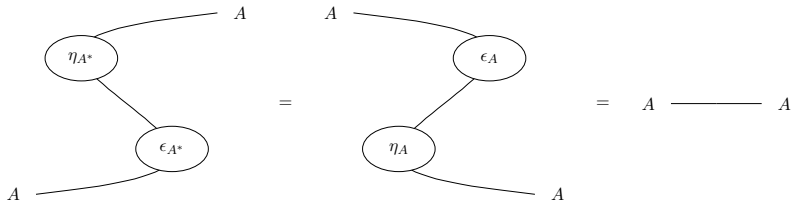
5. A COMPACT-CLOSED (BI)CATEGORY

In this section, we show that similarly to Joyal’s category of Conway games, our bicategory of concurrent games has a compact closed structure, a structure that is central in the applications of our framework to game semantics of programming languages.

Recall that a compact closed category is a symmetric monoidal category, where each object A has a *dual* A^* , which is related to A via two morphisms:

$$\eta_A : 1 \multimap A^* \otimes A \qquad \epsilon_A : A \otimes A^* \multimap 1$$

where 1 is the unit of the tensor (in our concrete case it is the empty game). These morphisms have to obey two laws that are best represented in the language of string diagrams:



Compact closed categories play an important role in the background in semantics: the equations of compact closed categories are mirrored, *e.g.* in the reduction rules of proof nets and in the adjunction laws (β and η -conversion) of cartesian closed or symmetric closed monoidal categories. In fact, any compact closed category is symmetric closed monoidal (more precisely, $*$ -autonomous, and a model of MLL [Gir87]): setting $A \multimap B = A^* \otimes B$, we have the adjunction $A \otimes - \dashv A \multimap -$. In short, compact closed categories form the backbone of an equational presentation of the dynamics of linear higher-order computation.

But unlike Conway games, CG is a *bicategory*. In fact, we believe that it gives an example of a *compact closed bicategory*, as defined by Kelly [Kel72] and detailed by Stay [Sta13]. However, the precise definition of a compact closed bicategory is rather intimidating. It might be possible to deduce the bicategorical compact closed structure of CG from that of the bicategory of profunctors [Sta13], but we do not do so here. Although in subsequent work we occasionally rely on the algebraic laws for composition pre-quotient, the literature and body of work that we need to connect to when setting up our game semantics (for instance, models of linear logic [Mel09]) does not exploit this bicategorical structure. So, we only check that the *quotiented category* is compact closed.

By abuse of notations, from now on we will use the same notation CG for the quotiented category instead of the bicategory. Regarded as a category, CG has esps as objects, and as morphisms strategies $\sigma : S \rightarrow A^\perp \parallel B$ up to isomorphism. In the rest of this section, we check the components of a compact closed category.

5.1. **The bifunctor.** First, we define a bifunctor $\otimes : \mathbf{CG}^2 \rightarrow \mathbf{CG}$. On objects, $A \otimes B$ is simply defined as $A \parallel B$. On morphisms, for $\sigma_1 : S_1 \rightarrow A_1^\perp \parallel B_1$ and $\sigma_2 : S_2 \rightarrow A_2^\perp \parallel B_2$, we define

$$\sigma_1 \otimes \sigma_2 = S_1 \parallel S_2 \xrightarrow{\sigma_1 \parallel \sigma_2} (A_1^\perp \parallel B_1) \parallel (A_2^\perp \parallel B_2) \xrightarrow{\gamma_{A_1^\perp, B_1, A_2^\perp, B_2}} (A_1 \parallel A_2)^\perp \parallel (B_1 \parallel B_2)$$

where $\gamma_{A,B,C,D} : (A \parallel B) \parallel (C \parallel D) \rightarrow (A \parallel C) \parallel (B \parallel D)$ is the obvious isomorphism of esps. We show that this operation is a bifunctor. Firstly, it preserves the identity.

Proposition 5.1. For any esp A , we have

$$\mathbb{C}_{A \otimes B} \cong \mathbb{C}_A \otimes \mathbb{C}_B$$

Proof. We have the isomorphism

$$\gamma_{A^\perp, B^\perp, A, B} : (A^\perp \parallel B^\perp) \parallel (A \parallel B) \rightarrow (A^\perp \parallel A) \parallel (B^\perp \parallel B)$$

which can also be typed as $\gamma_{A^\perp, B^\perp, A, B} : \mathbb{C}_{A \otimes B} \rightarrow \mathbb{C}_A \parallel \mathbb{C}_B$, which obviously commutes with the projections to the game. \square

Secondly, it preserves composition.

Proposition 5.2. Let:

$$\begin{array}{ll} \sigma_1 : S_1 \rightarrow A_1^\perp \parallel B_1 & \tau_1 : T_1 \rightarrow B_1^\perp \parallel C_1 \\ \sigma_2 : S_2 \rightarrow A_2^\perp \parallel B_2 & \tau_2 : T_2 \rightarrow B_2^\perp \parallel C_2 \end{array}$$

Then,

$$(\tau_1 \circ \sigma_1) \otimes (\tau_2 \circ \sigma_2) \cong (\tau_1 \otimes \tau_2) \circ (\sigma_1 \otimes \sigma_2)$$

Proof. We start by proving it for interactions. As the parallel composition of pullback squares is a pullback square, we have two pullbacks related by isomorphisms:

$$\begin{array}{ccc} & (T_1 \otimes S_1) \parallel (T_2 \otimes S_2) & \\ & \downarrow \vee & \\ (S_1 \parallel C_1) \parallel (S_2 \parallel C_2) & & (A_1 \parallel T_1) \parallel (A_2 \parallel T_2) \\ \swarrow (\sigma \parallel C_1) \parallel (\sigma_2 \parallel C_2) & & \swarrow (A_1 \parallel \tau_1) \parallel (A_2 \parallel \tau_2) \\ & (A_1 \parallel B_1 \parallel C_1) \parallel (A_2 \parallel B_2 \parallel C_2) & \\ \swarrow \gamma_{S_1, C_1, S_2, C_2} & & \swarrow \gamma_{A_1, T_2, A_2, T_2} \\ & (T_1 \parallel T_2) \otimes (S_1 \parallel S_2) \xrightarrow{\delta} & \\ \swarrow (\sigma_1 \otimes \sigma_2) \parallel (C_1 \parallel C_2) & & \swarrow (A_1 \parallel A_2) \parallel (\tau_1 \otimes \tau_2) \\ (S_1 \parallel S_2) \parallel (C_1 \parallel C_2) & & (A_1 \parallel A_2) \parallel (T_1 \parallel T_2) \\ \swarrow (\sigma_1 \parallel \sigma_2) \parallel (C_1 \parallel C_2) & & \swarrow (A_1 \parallel A_2) \parallel (\tau_1 \otimes \tau_2) \\ & (A_1 \parallel A_2) \parallel (B_1 \parallel B_2) \parallel (C_1 \parallel C_2) & \end{array}$$

where δ is the obvious map. By universal property of the pullback that gives an isomorphism:

$$\gamma' : (T_1 \otimes S_1) \parallel (T_2 \otimes S_2) \cong (T_1 \parallel T_2) \otimes (S_1 \parallel S_2)$$

which commutes (up to $\gamma_{A_1, C_1, A_2, C_2}$) with the hiding maps $\mathfrak{h}_{\sigma_1, \tau_1} \parallel \mathfrak{h}_{\sigma_2, \tau_2}$ and $\mathfrak{h}_{\sigma_1 \otimes \sigma_2, \tau_1 \otimes \tau_2}$, so using Proposition 4.6 and the easy fact that maps with hiding witnesses are stable by parallel composition, it follows that γ' corresponds to a unique isomorphism:

$$(T_1 \odot S_1) \parallel (T_2 \odot S_2) \cong (T_1 \parallel T_2) \odot (S_1 \parallel S_2)$$

between strategies $(\tau_1 \odot \sigma_1) \otimes (\tau_2 \odot \sigma_2)$ and $(\tau_1 \parallel \tau_2) \odot (\sigma_1 \parallel \sigma_2)$. \square

5.2. Lifting and symmetric monoidal structure of CG. The strategies serving as structural morphisms for the symmetric closed monoidal structure are very simple variants of copycat $\mathfrak{c}_A : A \dashv\dashv A$. In order to construct the symmetric monoidal structure of CG, we describe a systematic way of generating such morphisms from more elementary maps of esps.

Definition 5.3. Let $f : A \rightarrow B$ be a receptive, courteous map of esps³. Then, the map:

$$\begin{aligned} \bar{f} &: \mathbb{C}_A \rightarrow A^\perp \parallel B \\ a &\mapsto (A^\perp \parallel f) \circ \mathfrak{c}_A(a) \end{aligned}$$

is a strategy called the **lifting** of f . Likewise, if $f : B^\perp \rightarrow A^\perp$ is receptive and courteous, we define its **co-lifting**:

$$\begin{aligned} \bar{f}^\perp &: \mathbb{C}_B \rightarrow A^\perp \parallel B \\ c &\mapsto (f \parallel B) \circ \mathfrak{c}_B(c) \end{aligned}$$

The fact that they are strategies follows from the fact that courteous receptive maps are stable under composition.

The following key lemma links composition of strategies with lifted maps and composition of the corresponding maps in \mathcal{E} .

Lemma 5.4. Let $f : B \rightarrow C$ be a receptive courteous map of esps, and $\sigma : S \rightarrow A^\perp \parallel B$ be a strategy. Then, the unitor $\lambda_\sigma : \mathbb{C}_B \odot S \rightarrow S$ is an isomorphism between

$$\begin{aligned} \bar{f} \odot \sigma &: \mathbb{C}_B \odot S \rightarrow A^\perp \parallel C \\ (A^\perp \parallel f) \circ \sigma &: S \rightarrow A^\perp \parallel C \end{aligned}$$

Likewise, for $f : B^\perp \rightarrow A^\perp$ receptive courteous and $\sigma : S \rightarrow B^\perp \parallel C$ a strategy, ρ_σ is an isomorphism between:

$$\begin{aligned} \sigma \odot \bar{f}^\perp &: S \odot \mathbb{C}_B \rightarrow A^\perp \parallel C \\ (f \parallel C) \circ \sigma &: S \rightarrow A^\perp \parallel C \end{aligned}$$

Proof. By definition, the following two diagrams commute:

$$\begin{array}{ccc} S & \xrightarrow{\quad S \quad} & S \\ \downarrow \sigma & & \downarrow \sigma \\ A^\perp \parallel B & \xrightarrow{\quad A^\perp \parallel B \quad} & A^\perp \parallel B \end{array} \quad \begin{array}{ccc} \mathbb{C}_B & \xrightarrow{\quad \mathbb{C}_B \quad} & \mathbb{C}_B \\ \downarrow \mathfrak{c}_B & & \downarrow \bar{f} \\ B^\perp \parallel B & \xrightarrow{\quad B^\perp \parallel f \quad} & B^\perp \parallel C \end{array}$$

³This means that, technically, f is a strategy on B – though we are not thinking of it that way.

Therefore, by Lemma 4.4, it follows that the following diagram commutes:

$$\begin{array}{ccc} \mathbb{C}_B \odot S & \xrightarrow{\mathbb{C}_B \odot S} & \mathbb{C}_B \odot S \\ \downarrow \alpha_B \odot \sigma & & \downarrow \bar{f} \odot \sigma \\ A^\perp \parallel B & \xrightarrow{A^\perp \parallel f} & A^\perp \parallel C \end{array}$$

Combined with the isomorphism $\alpha_B \odot \sigma \cong \sigma$, this concludes the proof. The other case is symmetric. \square

From the lemma above it immediately follows that lifting is functorial:

Lemma 5.5. Let $f : A \rightarrow B$ and $g : B \rightarrow C$ be receptive courteous maps, then we have an isomorphism:

$$\bar{g} \odot \bar{f} \cong \overline{g \circ f}$$

Proof. Immediate consequence of Lemma 5.4. \square

Using this, we can lift the symmetric closed monoidal structure of \mathcal{E} to CG. In particular, there are natural isomorphisms in \mathcal{E} which are componentwise receptive and courteous, and so are their inverses.

$$\begin{aligned} \rho_A & : & A \parallel 1 & \rightarrow & A \\ \lambda_A & : & 1 \parallel A & \rightarrow & A \\ s_{A,B} & : & A \parallel B & \rightarrow & B \parallel A \\ \alpha_{A,B,C} & : & (A \parallel B) \parallel C & \rightarrow & A \parallel (B \parallel C) \end{aligned}$$

(the reuse of symbols from Section 4 for these structural morphisms should not cause any confusion). These isomorphisms can then be lifted to strategies:

$$\begin{aligned} \bar{\rho}_A & : & A \parallel 1 & \dashrightarrow & A \\ \bar{\lambda}_A & : & 1 \parallel A & \dashrightarrow & A \\ \bar{s}_{A,B} & : & A \parallel B & \dashrightarrow & B \parallel A \\ \bar{\alpha}_{A,B,C} & : & (A \parallel B) \parallel C & \dashrightarrow & A \parallel (B \parallel C) \end{aligned}$$

which inherit from \mathcal{E} all the coherence laws of the symmetric monoidal structure by Lemma 5.5. It remains to prove that these families are natural.

Lemma 5.6. The families $\rho_A, \lambda_A, s_{A,B}, \alpha_{A,B,C}$ are natural in all their components.

Proof. A direct verification. For illustration, we detail the naturality of $s_{A,B}$.

Let $\sigma : S \rightarrow A_1^\perp \parallel A_2$, and $\tau : T \rightarrow B_1^\perp \parallel B_2$. We need to check:

$$\overline{s_{A_2, B_2}} \odot (\sigma \otimes \tau) \cong (\tau \otimes \sigma) \odot \overline{s_{A_1, B_1}}$$

But there is an obvious isomorphism $\overline{s_{A_1, B_1}} \cong \overline{s_{A_1^\perp, B_1^\perp}^{-1}}^\perp$. So by both parts of Lemma 5.4, this amounts to finding an isomorphism between the two maps:

$$\begin{array}{ccc} S \parallel T & \xrightarrow{((A_1 \parallel B_1)^\perp \parallel s_{A_2, B_2}) \circ \gamma_{A_1^\perp, A_2, B_1^\perp, B_2} \circ (\sigma \parallel \tau)} & (A_1 \parallel B_1)^\perp \parallel (B_2 \parallel A_2) \\ T \parallel S & \xrightarrow{(s_{A_1^\perp, B_1^\perp}^{-1} \parallel (B_2 \parallel A_2)) \circ \gamma_{B_1^\perp, B_2, A_1^\perp, A_2} \circ (\tau \parallel \sigma)} & (A_1 \parallel B_1)^\perp \parallel (B_2 \parallel A_2) \end{array}$$

and it is a simple verification to check that $s_{S,T}$ does the trick. \square

This concludes the symmetric monoidal structure of CG.

5.3. Compact closed structure. The dual of a game A is simply defined as A^\perp . We have two strategies:

$$\begin{aligned}\eta_A &: \mathbb{C}_A \rightarrow 1^\perp \parallel (A^\perp \parallel A) \\ \epsilon_A &: \mathbb{C}_A \rightarrow (A \parallel A^\perp)^\perp \parallel 1\end{aligned}$$

defined in the obvious way. We have:

Proposition 5.7. The strategies $\eta_A : 1 \multimap A^\perp \parallel A$ and $\epsilon_A : A \parallel A^\perp \multimap 1$ satisfy the laws for a compact closed category.

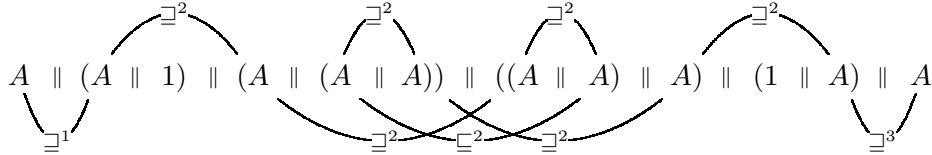
Proof. We need to check the two equations of duals in compact closed categories:

$$\begin{aligned}\mathfrak{c}_A &\cong \overline{\lambda_A} \odot (\epsilon_A \otimes \mathfrak{c}_A) \odot \overline{\alpha_{A,A^\perp,A}^{-1}} \odot (\mathfrak{c}_A \otimes \eta_A) \odot \overline{\rho_A}^{-1} \\ \mathfrak{c}_{A^\perp} &\cong \overline{\rho_{A^\perp}} \odot (\mathfrak{c}_{A^\perp} \otimes \epsilon_A) \odot \overline{\alpha_{A^\perp,A,A^\perp}} \odot (\eta_A \otimes \mathfrak{c}_{A^\perp}) \odot \overline{\lambda_{A^\perp}}^{-1}\end{aligned}$$

These two isomorphisms are symmetric; we only check the first. Let us write $\sigma : S \rightarrow A^\perp \parallel A$ for the resulting composition, and

$$\xi : U \rightarrow A \parallel (A \parallel 1) \parallel (A \parallel (A \parallel A)) \parallel ((A \parallel A) \parallel A) \parallel (1 \parallel A) \parallel A$$

for the corresponding 5-ary composition. By Lemma 4.9, there is a hiding map $\mathfrak{h} : U \multimap S$, commuting with the projection to the game. From the characterisation of configurations of pullbacks, and after eliminating redundancies, configurations of U correspond to the data of a configuration in each component A above, satisfying the following constraints:



where, moreover, configurations whose maximal events are visible (and so correspond to configurations of S) are those where the \sqsubseteq^1 are replaced by \sqsubseteq^+ , the \sqsubseteq^2 are replaced by equalities and the \sqsubseteq^3 are replaced by \sqsubseteq^- . Such configurations exactly correspond to those of \mathbb{C}_A . \square

This concludes the proof that CG is a compact closed category.

6. CONCLUSIONS

In this paper, we gave a detailed exposition of the results of [RW11], along with some extensions. We presented a notion of concurrent games based on event structures, which is a concurrent analogue of Joyal's compact closed category of Conway games [Joy77].

We first defined pre-strategies, as certain event structures describing the evolution of concurrent processes on an interface presented as a game. We defined strategies as those pre-strategies stable under the action of an asynchronous forwarder, presented as the copycat strategy. Finally, we proved that composition of strategies obeys the laws of a bicategory, and that just as Joyal's, the corresponding quotient category is compact closed. As explained in [Win13b], it relates to the compact closed bicategory of profunctors via a lax functor.

6.0.1. *Further work.* The developments presented in this paper are just the beginning of the story. Since the appearance of [RW11], this framework has been used as a basis for a number of extensions. In [CGW12], games were equipped with winning conditions. It was proved that winning strategies also form a bicategory, and that just as in the sequential case, well-founded games that satisfy a further condition called *race-freeness* are determined. This was later extended to all Borel winning conditions [GW14], provided in addition that concurrency is bounded in the game. Winning conditions were also generalized to a quantitative notion of payoff in [CW13], and a value theorem was proved. As witnessed by these determinacy results, and despite concurrency, our games remain total information games (unlike *e.g.* [dAHK07]). We investigated in [Win12, CGW13] an extension to partial information games, where determinacy is lost. The fourth author also extended the setting to probabilistic and quantum strategies [Win13a].

In our basic setting, games are affine: each event can occur at most *once*. It is key for many applications (most notably to semantics) that one allows the replication of events, in such a way that distinct copies are indistinguishable from each other. To this effect, we equipped games with a notion of symmetry expressing indistinguishability of events and configurations. Strategies then have to respect this additional structure, by treating symmetric configurations uniformly. This can be done in two ways: the first option is to saturate strategies by forcing them to play non-deterministically all symmetric events. In [CCW14], we developed a bicategory of saturated strategies on games with symmetry, using it to allow replication and construct analogues of AJM [AJM00] and HO [HO00] games. In [CCW15] we developed a second option, and showed that with some minimality assumption on strategies one could obtain a bicategory of uniform strategies while avoiding saturation and the addition of redundant non-deterministic choices. We showed that this gave a cartesian closed category, supporting an intensionally fully abstract model of PCF where independent sub-computations are performed in parallel.

6.0.2. *Perspectives.* There is a lot of ongoing work on the topic of concurrent games on event structures. On the fundamental side, we have looked for a generalization of the basic setting presented here that accommodates better events with *disjunctive causality*, *i.e.* that can occur for several distinct yet compatible reasons [dVW16]. On the semantic side, we have several research directions. To name a few, we want to represent non-interference as determinism in concurrent languages; to enrich strategies to keep information about possible local deadlocks or divergences; to investigate further strategies from the point of view of concurrent processes [CHLW14]; and to mix symmetry with probabilities in order to build a denotational model combining probabilities, non-determinism and concurrency.

But beyond semantics, our concurrent games give a powerful and precise description of the evolution of concurrent processes. We wish to extend this basic framework in order to set a standard for a concurrent notion of games and strategies. We hope this framework will then be a relevant and useful tool for various purposes, from handling algorithmic issues in concurrency to investigating its logical properties.

6.0.3. *Acknowledgements.* We gratefully acknowledge the support of the French LABEX MILYON (ANR-10-LABX-0070), and of the ERC Advanced Grant ECSYM.

REFERENCES

- [AJM00] Samson Abramsky, Radha Jagadeesan, and Pasquale Malacaria. Full abstraction for PCF. *Inf. Comput.*, 163(2):409–470, 2000.
- [AM99] Samson Abramsky and Paul-André Melliès. Concurrent games and full completeness. In *LICS*, pages 431–442. IEEE Computer Society, 1999.
- [CC16] Simon Castellan and Pierre Clairambault. Causality vs. interleavings in concurrent game semantics. In Josée Desharnais and Radha Jagadeesan, editors, *27th International Conference on Concurrency Theory, CONCUR 2016, August 23-26, 2016, Québec City, Canada*, volume 59 of *LIPIcs*, pages 32:1–32:14. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 2016.
- [CCW14] Simon Castellan, Pierre Clairambault, and Glynn Winskel. Symmetry in concurrent games. In Henzinger and Miller [HM14], page 28.
- [CCW15] Simon Castellan, Pierre Clairambault, and Glynn Winskel. 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, 2015.
- [CGW12] Pierre Clairambault, Julian Gutierrez, and Glynn Winskel. The winning ways of concurrent games. In *Proceedings of the 27th Annual IEEE Symposium on Logic in Computer Science, LICS 2012, Dubrovnik, Croatia, June 25-28, 2012*, pages 235–244. IEEE Computer Society, 2012.
- [CGW13] Pierre Clairambault, Julian Gutierrez, and Glynn Winskel. Imperfect information in logic and concurrent games. In Bob Coecke, Luke Ong, and Prakash Panangaden, editors, *Computation, Logic, Games, and Quantum Foundations. The Many Facets of Samson Abramsky - Essays Dedicated to Samson Abramsky on the Occasion of His 60th Birthday*, volume 7860 of *Lecture Notes in Computer Science*, pages 7–20. Springer, 2013.
- [CHLW14] Simon Castellan, Jonathan Hayman, Marc Lasson, and Glynn Winskel. Strategies as concurrent processes. *Electr. Notes Theor. Comput. Sci.*, 308, 2014.
- [CW13] Pierre Clairambault and Glynn Winskel. On concurrent games with payoff. *Electr. Notes Theor. Comput. Sci.*, 298:71–92, 2013.
- [dAH00] Luca de Alfaro and Thomas A. Henzinger. Concurrent omega-regular games. In *LICS '00*, pages 141–154. IEEE Computer Society, 2000.
- [dAHK07] Luca de Alfaro, Thomas A. Henzinger, and Orna Kupferman. Concurrent reachability games. *Theor. Comput. Sci.*, 386(3):188–217, 2007.
- [dVW16] Marc de Visme and Glynn Winskel. Strategies with parallel causes. 2016.
- [FP09] Claudia Faggian and Mauro Piccolo. Partial orders, event structures and linear strategies. In Pierre-Louis Curien, editor, *TLCA*, volume 5608 of *LNCS*, pages 95–111. Springer, 2009.
- [GGMW13] Blaise Genest, Hugo Gimbert, Anca Muscholl, and Igor Walukiewicz. Asynchronous games over tree architectures. In Fedor V. Fomin, Rusins Freivalds, Marta Z. Kwiatkowska, and David Peleg, editors, *Automata, Languages, and Programming - 40th International Colloquium, ICALP 2013, Riga, Latvia, July 8-12, 2013, Proceedings, Part II*, volume 7966 of *Lecture Notes in Computer Science*, pages 275–286. Springer, 2013.
- [Gir87] J.Y. Girard. Linear logic. *Theoretical computer science*, 50(1):1–101, 1987.
- [GM08] Dan R. Ghica and Andrzej S. Murawski. Angelic semantics of fine-grained concurrency. *Ann. Pure Appl. Logic*, 151(2-3):89–114, 2008.
- [GW14] Julian Gutierrez and Glynn Winskel. On the determinacy of concurrent games on event structures with infinite winning sets. *J. Comput. Syst. Sci.*, 80(6):1119–1137, 2014.
- [Hir13] Tom Hirschowitz. Full abstraction for fair testing in CCS. In Reiko Heckel and Stefan Milius, editors, *CALCO*, volume 8089 of *LNCS*, pages 175–190. Springer, 2013.
- [HM14] Thomas A. Henzinger and Dale Miller, 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*. ACM, 2014.
- [HO00] J. M. E. Hyland and C.-H. Luke Ong. On full abstraction for PCF: I, II, and III. *Inf. Comput.*, 163(2):285–408, 2000.
- [HP12] Tom Hirschowitz and Damien Pous. Innocent strategies as presheaves and interactive equivalences for CCS. *Sci. Ann. Comp. Sci.*, 22(1):147–199, 2012.

- [Joy77] Andr Joyal. Remarques sur la thorie des jeux deux personnes. *Gazette des Sciences Mathematiques du Quebec* 1(4), pages 46 – 52, 1977.
- [Kel72] G.M. Kelly. Many-variable functorial calculus. i. In G.M. Kelly, M. Laplaza, G. Lewis, and Saunders Mac Lane, editors, *Coherence in Categories*, volume 281 of *Lecture Notes in Mathematics*, pages 66–105. Springer Berlin Heidelberg, 1972.
- [Lai01] James Laird. A game semantics of idealized CSP. *Electr. Notes Theor. Comput. Sci.*, 45:232–257, 2001.
- [LFHY14] Ugo Dal Lago, Claudia Faggian, Ichiro Hasuo, and Akira Yoshimizu. The geometry of synchronization. In Henzinger and Miller [HM14], pages 35:1–35:10.
- [Mar98] Donald A Martin. The determinacy of blackwell games. *The Journal of Symbolic Logic*, 63(04):1565–1581, 1998.
- [Mel09] Paul-André Melliès. Categorical semantics of linear logic. *Panoramas et synthèses*, 27:15–215, 2009.
- [MM07] Paul-André Melliès and Samuel Mimram. Asynchronous games: Innocence without alternation. In Luís Caires and Vasco Thudichum Vasconcelos, editors, *CONCUR*, volume 4703 of *LNCS*, pages 395–411. Springer, 2007.
- [NPW81] Mogens Nielsen, Gordon Plotkin, and Glynn Winskel. Petri nets, event structures and domains. *Theoretical Computer Science*, 13:85–108, 1981.
- [RW11] Silvain Rideau and Glynn Winskel. Concurrent strategies. In *LICS*, pages 409–418. IEEE Computer Society, 2011.
- [Sta13] Michael Stay. Compact closed bicategories, 2013.
- [Win80] Glynn Winskel. *Events in computation*. PhD Thesis, Edinburgh University, 1980.
- [Win82] Glynn Winskel. Event structure semantics for CCS and related languages. In *ICALP’82*, volume 140 of *LNCS*. Springer, 1982.
- [Win86] Glynn Winskel. Event structures. In Wilfried Brauer, Wolfgang Reisig, and Grzegorz Rozenberg, editors, *Advances in Petri Nets*, volume 255 of *LNCS*, pages 325–392. Springer, 1986.
- [Win12] Glynn Winskel. Winning, losing and drawing in concurrent games with perfect or imperfect information. In *Festschrift for Dexter Kozen*, volume 7230 of *LNCS*. Springer, 2012.
- [Win13a] Glynn Winskel. Distributed probabilistic and quantum strategies. *Electr. Notes Theor. Comput. Sci.*, 298:403–425, 2013.
- [Win13b] Glynn Winskel. Strategies as profunctors. In Frank Pfenning, editor, *FoSSaCS*, volume 7794 of *LNCS*, pages 418–433. Springer, 2013.