HAL open science

The Mays and Musts of Concurrent Strategies

Simon Castellan, Pierre Clairambault, Glynn Winskel

To cite this version:

Simon Castellan, Pierre Clairambault, Glynn Winskel. The Mays and Musts of Concurrent Strategies. 2021. hal-03323995

HAL Id: hal-03323995

https://hal.science/hal-03323995

Preprint submitted on 23 Aug 2021

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.

The Mays and Musts of Concurrent Strategies

Simon Castellan[1], Pierre Clairambault[2], and Glynn Winskel[3]

[1]Inria, Univ Rennes, IRISA, France
[2] Univ Lyon, EnsL, UCBL, CNRS, LIP, F-69342, LYON Cedex 07, France
[3]Computer and Information Sciences, University of Strathclyde, Scotland

Abstract

Concurrent strategies based on event structures are examined from the viewpoint of 'may' and 'must' testing in traditional process calculi. In their pure form concurrent strategies fail to expose the deadlocks and divergences that can arise in their composition. This motivates an extension of the bicategory of concurrent strategies to treat the 'may' and 'must' behaviour of strategies under testing. One extension adjoins neutral moves to strategies but in so doing loses identities w.r.t. composition. This in turn motivates another extension in which concurrent strategies are accompanied by stopping configurations; the ensuing stopping strategies inherit the structure of a bicategory from that of strategies. The technical developments converge in providing characterisations of the 'may' and 'must' equivalences and preorders on strategies.

1 Introduction

This article relates to work on process calculi of the 1980's but from a modern perspective of processes as strategies, specifically as distributed/concurrent strategies based on event structures. It expands on two areas close to Samson Abramsky's heart, game semantics and concurrency: on a development of concurrent games based on event structures which extends his early ideas with Paul-André Melliès of deterministic concurrent strategies as closure operators [1]; and equivalences on concurrent processes through testing [2].

Robin Milner and Tony Hoare's work of late seventies and early eighties drew attention to equivalences on processes; Milner's on forms of *bisimulation* [3] and Hoare's on *failures* equivalence [4]. Hoare had described failure equivalence informally as the minimum extension of trace equivalence that takes account of the possibility of failure due to deadlock. Matthew Hennessy and his PhD student

Rocco de Nicola provided a rationale through an idea of testing processes [5]. For them a test was a process with distinguished "success" states at which an action ✓ could occur. Putting a test in parallel composition with a process, *may* lead to success if some run does or *must* lead to success if all runs do. Processes can be regarded as equivalent if they have the same 'may' and 'must' behaviour w.r.t. tests. Modulo subtleties to do with the divergence of processes, Hennessy and de Nicola recovered failure equivalence as testing equivalence. What about Milner's central equivalence? Samson Abramsky investigated the extent to which bisimulation could be viewed as a testing equivalence [2]: it could, but only at the cost of strengthening the power of tests considerably, by allowing testing to run and copy processes quite liberally.

Here we shall examine the 'may' and 'must' equivalence of concurrent strategies based on event structures [6, 7]—foreshadowed in the early definitions of concurrent strategy [1, 8, 9, 10]. Informally, a *strategy* for Player in a two-party game against Opponent, expresses a choice of Player moves, most often in reaction to moves made by Opponent, unpredictable for Player but for the constraints of the game. We shall implicitly regard a strategy as a strategy for Player. We regard Opponent as the environment uncontrollable by Player. We can express both the game—its moves and their constraints—and a strategy—its choice of Player moves subject to the moves of Opponent—as event structures. This chimes with our view of strategies and games as highly distributed. Player and Opponent are more accurately thought of as teams of players and opponents acting at possibly very different locations. Though we take the rather abstract view of location advocated by Petri in his concept of local state as a condition (or place): then locality reveals itself through the causal dependence and independence of events.

Event structures are the concurrent analogue of trees; just as transition systems unfold to trees, so Petri nets unfold to event structures. Whereas an unfolded behaviour of a transition system comprises sequences of actions/events, the unfolded behaviour of a Petri net, in which events make local changes to conditions, comprises partial orders of causal dependency between event occurrences [11]. Event structures are a central model for concurrent computation, related to other models by adjunctions [12]. This plants concurrent strategies based on event structures firmly within theories of concurrency and interaction—anticipated in Abramsky's presentation of game semantics, with its emphasis on composition of strategies as given by their parallel interaction followed by hiding. Perhaps more controversially, the view of processes as strategies suggests refinements to the assumptions usual in process calculi. In concurrent strategies, gone is the usual symmetry between a process and its environment; the conditions on a concurrent strategy take account of the unpredictability and uncontrollability of Opponent moves. This affects the appropriate equivalences to impose between concurrent strategies.

There is surely a long history behind the idea of composing strategies. Certainly the idea plays a key role in John Conway's "On Numbers and Games" [13], the categorical underpinnings of which were exposed by André Joyal [14]. For two-party games there is the obvious operation of reversing the roles of the

two participants, Player and Opponent; this operation, forming the *dual* $G^\perp$ of a game G, played the role of *negation* for Conway. A useful convention is to regard a strategy in a game G as a strategy for Player; then a strategy for Opponent, or counter-strategy, is a strategy in the dual game $G^\perp$. If the games are broad enough, they often support a form of parallel composition, $G \| H$; for Conway it was the *sum* of games. A strategy *from* a game G *to* a game H is a strategy σ *in* the game $G^\perp \| H$. Given another strategy this time from the game H to the game K, *i.e.* a strategy τ in the game $H^\perp \| K$, we can let the strategies *interact* as $\tau \otimes \sigma$, essentially by playing them against each other over the common game H; there the strategies σ and τ adopt complementary roles—where one makes a move of Player in H the other sees a move of Opponent and *vice versa*.

The interaction $\tau \otimes \sigma$ involves moves in the parallel composition of all three games, $G^\perp \| H \| K$, though in writing the parallel composition in this way an imprecision has crept in: whereas the moves over $G^\perp$ and K described by $\tau \otimes \sigma$ are choices of moves for Player or moves open to choices of Opponent, those over H are either instantiations of Opponent moves of σ by Player moves of τ, or the converse, instantiations of Opponent moves of τ by Player moves of σ. As such the moves of $\tau \otimes \sigma$ over H behave like synchronisations between complementary moves of σ and τ, and as events internal to the interaction. Though internal, the events over H can affect the behaviour of the interaction by introducing deadlocks or divergence. In the *composition* of strategies it is usual to hide the internal events of interaction to obtain a strategy $\tau \odot \tau$ in the game $G^\perp \| K$, where the game H is elided to obtain a strategy from G to K. However when the original strategies σ or τ are nondeterministic significant behavioural distinctions can be lost in hiding internal events. In particular, the hidden events can affect the 'must' behaviour of the composition of strategies.

1.1 Contributions of the paper

This brings us to the concerns of this paper. It motivates the definition of *bare* concurrent strategies in which internal events are exposed as neutral moves in the strategy. Through bare strategies we can examine the 'may' and 'must' testing of strategies. Although bare strategies compose their composition does not have identities, so they fail to form a bicategory. We have explored two ways to recover a bicategory while remaining faithful to the 'must' behaviour of strategies. One is through "essential events" in which one strips a bare strategy down to just those neutral moves critical to its behaviour [15]. The other, that we follow here, is through extending strategies with the extra structure of *stopping configurations* [16]. By distinguishing certain configurations as stopping we keep track of those visible configurations at which the strategy may appear to get stuck through the occurrence of hidden neutral moves.

Stopping configurations are the event-structure analogue of Russ Harmer and Guy McCusker's "divergences" [17], though event structures add the refinement of locality and independence to the concept. In an interleaving model, in which behaviour is captured through sequences of actions, divergence anywhere has a

global effect; generally, in the parallel composition of two processes if one can perform an infinite sequence of actions, these may block progress of the other process, unless additional fairness assumptions are enforced. This is not so in a model such as event structures where the independence/concurrency of actions is explicit. In a nondeterministic strategy a Player move is not blocked by the occurrence of moves with which it is independent. Whereas an interleaving model may require weak fairness assumptions these are generally built into the behaviour of strategies as event structures [18]. This makes for subtle differences in the nature of 'must' testing in concurrent games w.r.t. traditional games.

Perhaps surprisingly, in many situations a concurrent strategy may be replaced by its simpler "rigid image" in the game, despite this often forgetting nondeterministic branching—rigid-image strategies form a category rather than just a bicategory [19]—and indeed this remains true for strategies with stopping configurations; none of the 'may' or 'must' behaviour is lost.

Our technical contribution concludes with characterisations of the 'may' and 'must' equivalences and preorders on strategies. The 'may' equivalence of strategies is captured through their inducing the same set of finite traces; a trace being understood as a sequence of moves in the game. This echoes the earlier results of Ghica and Murawski when showing their non-alternating games model is fully-abstract for Idealized Parallel Algol with respect to may-convergence [8]. For 'must' equivalence, our result is to be compared with that of Harmer and McCusker for their sequential games model, based on Hyland-Ong games with explicit divergences [17]. But whereas in a sequential setting only the first divergence matters, for us 'must' equivalence of strategies is equivalent to their sharing the same traces of *all* (possibly infinite) stopping configurations. See Example 7.2 and what follows for an in-depth discussion.

Because we restrict attention to linear bicategories of strategies, the tests here are linear too; they do not permit the tested strategy to be copied and rerun. From the point of view of distributed computation linearity is natural: it is often infeasible to copy a distributed system or strategy [20]. Then single-run 'may' and 'must' tests are appropriate.

On the other hand, most programming languages do allow some form of copying and nonlinearity. Through the addition of symmetry we can adjoin pseudo (co)monads—where the traditional laws hold up to symmetry, and model nonlinear features [21, 22]. Through symmetry and pseudo comonads, we can realise a variety of nonlinear forms of testing, in which a test could dynamically copy and retest the strategy of interest. The nature of such broader testing on strategies, the equivalences and logics induced, are not well understood and deserve a systematic study, for which this paper forms a foundation. As shown by Samson Abramsky such broader tests are needed to realise equivalences such as bisimulation as testing equivalences [2].

2 Event structures

An *event structure* comprises $(E, \leq, \mathrm{Con})$, consisting of a set E of *events* which are partially ordered by $\leq$, the *causal dependency relation*, and a nonempty *consistency* relation Con consisting of finite subsets of E. The relation $e' \leq e$ expresses that event e causally depends on the previous occurrence of event e'. That a finite subset of events is consistent conveys that its events can occur together by some stage in the evolution of the process. Together the relations satisfy several axioms. We insist that the partial order is *finitary*, *i.e.*

- $[e] =_{\mathrm{def}} \{e' \mid e' \leq e\}$ is finite for all $e \in E$,

and that consistency satisfies

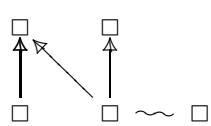
- $\{e\} \in \mathrm{Con}$ for all $e \in E$,
- $Y \subseteq X \in \mathrm{Con}$ implies $Y \in \mathrm{Con}$, and
- $X \in \mathrm{Con}$ & $e \leq e' \in X$ implies $X \cup \{e\} \in \mathrm{Con}$.

There is an accompanying notion of state, or history, those events that may occur up to some stage in the behaviour of the process described. A *configuration* is a, possibly infinite, set of events $x \subseteq E$ which is:

- *consistent,* $X \subseteq x$ and X is finite implies $X \in \mathrm{Con}$; and
- *down-closed,* $e' \leq e \in x$ implies $e' \in x$.

Two events e, e' are called *concurrent* if the set $\{e, e'\}$ is in Con and neither event is causally dependent on the other; then we write $e \; co \; e'$. In games the relation of *immediate* dependency $e \rightarrowtriangle e'$, meaning e and e' are distinct with $e \leq e'$ and no event in between, plays a very important role. We write $[X]$ for the down-closure of a subset of events X. Write $\mathcal{C}^\infty(E)$ for the configurations of E and $\mathcal{C}(E)$ for its finite configurations. (Sometimes we shall need to distinguish the precise event structure to which a relation is associated and write, for instance, $\leq_E$, $\rightarrowtriangle_E$ or co_E.)

Example 2.1. In examples it is often convenient to draw event structures. Often, though not always, consistency is determined in a binary fashion, in that a set of events is consistent if all of its pairs are. Then, it is economical to draw the binary relation of *conflict*, or inconsistency. For example, in the diagram



we illustrate the relations of immediate causal dependency $\longrightarrow\!\!\triangleright$ which yields the Hasse diagram of the partial order of causal dependency between events $\square$, and conflict by the wiggly line $\sim\sim\sim$. Neither the two events related by $\sim\sim\sim$ nor their dependants w.r.t. causal dependency can occur together in a configuration; there is no need draw all the conflicts that follow. $\square$

Let E and E' be event structures. A *map* of event structures $f : E \to E'$ is a partial function on events $f : E \rightharpoonup E'$ such that for all $x \in \mathcal{C}^\infty(E)$ its direct image $fx \in \mathcal{C}^\infty(E')$ and

$$\text{if } e_1, e_2 \in x \text{ and } f(e_1) = f(e_2) \text{ (with both defined), then } e_1 = e_2.$$

(Those maps defined is unaffected if we replace possibly infinite configurations $\mathcal{C}^\infty(E)$ by finite configurations $\mathcal{C}(E)$ above; this is because any configuration is the union of finite configurations and direct image preserves such unions.)

Maps of event structures compose as partial functions, with identity maps given by identity functions. Say a map is *total* if the function f is total. Notice that for a total map f the condition on maps now says it is *locally injective*, in the sense that w.r.t. any configuration x of the domain the restriction of f to a function from x is injective; the restriction of f to a function from x to fx is thus bijective. Say a total map of event structures is *rigid* when it preserves causal dependency.

Although a map $f : E \to E'$ of event structures does not generally preserve causal dependency, it does locally reflect causal dependency: whenever $e, e' \in x$, a configuration of E, and $f(e)$ and $f(e')$ are both defined with $f(e') \leq f(e)$, then $e' \leq e$. Consequently, f preserves the concurrency relation: if $e \; co \; e'$ in E and $f(e)$ and $f(e')$ are both defined then $f(e) \; co \; f(e')$.

3 Constructions

We provide the constructions which we use in the paper.

3.1 Partial-total factorisation

We shall realise an operation of hiding events via a factorisation property of maps of event structures.

Let $(E, \leq, \mathrm{Con})$ be an event structure. Let $V \subseteq E$ be a subset of 'visible' events. Define $E{\downarrow}V =_{\mathrm{def}} (V, \leq_V, \mathrm{Con}_V)$, where $v \leq_V v'$ iff $v \leq v'$ & $v, v' \in V$ and $X \in \mathrm{Con}_V$ iff $X \in \mathrm{Con}$ & $X \subseteq V$. The operation projects E to visible events V.

Consider a partial map of event structures $f : E \to E'$. Let

$$V =_{\mathrm{def}} \{e \in E \mid f(e) \text{ is defined}\}.$$

Then f clearly factors into the composition

$$E \xrightarrow{f_0} E{\downarrow}V \xrightarrow{f_1} E'$$

of f_0, a partial map of event structures taking $e \in E$ to itself if $e \in V$ and undefined otherwise, and f_1, a total map of event structures acting like f on V. We call f_1 the *defined part* of the partial map f. We say a map $f : E \to E'$ is a *projection* if its defined part is an isomorphism.

The *partial-total factorisation* is characterised to within isomorphism by the following universal property: for any factorisation

$$f : E \xrightarrow{g_0} E_1 \xrightarrow{g_1} E'$$

where g_0 is partial and g_1 is total there is a (necessarily total) unique map $h : E{\downarrow}V \to E_1$ such that

$$\begin{tikzcd} E \arrow[r, "f_0"] \arrow[dr, "g_0"'] & E{\downarrow}V \arrow[r, "f_1"] \arrow[d, dashed, "h"] & E' \\ & E_1 \arrow[ur, "g_1"'] & \end{tikzcd}$$

commutes.

3.2 Pullback

Event structures and their maps have pullbacks. For the composition of strategies we shall only need pullbacks of total maps. Consider a pullback

$$\begin{tikzcd} & P \arrow[dl, "\pi_1"'] \arrow[dr, "\pi_2"] \arrow[d, phantom, "\lrcorner"] & \\ A \arrow[dr, "f"'] & & B \arrow[dl, "g"] \\ & C & \end{tikzcd}$$

where f and g are total. Pullbacks are difficult to construct directly on the "prime" event structures we are using here, essentially because they associate each event with a unique minimum causal history. Such constructs are best first carried out in a broader model. Here we build the pullback of event structures out of the stable family of secured bijections.

Definition 3.1. A *secured bijection* comprises a composite bijection

$$\theta_{x,y} : x \cong fx = gy \cong y$$

between configurations $x \in \mathcal{C}^\infty(A)$ and $y \in \mathcal{C}^\infty(B)$ s.t. $fx = gy$, which is *secured* in the sense that the transitive relation generated on $\theta_{x,y}$ by taking

$$(a, b) \leq (a', b') \text{ if } a \leq_A a' \text{ or } b \leq_B b'$$

is a finitary partial order. Let $\mathcal{B}$ be the family of secured bijections. Say a subset $Z \subseteq \mathcal{B}$ is *compatible* iff $\exists \theta' \in \mathcal{B} \forall \theta \in Z.\ \theta \subseteq \theta'$.

Proposition 3.2. *The family $\mathcal{B}$ is a* stable family,[1]
i.e. it is

[1] Here it is useful to allow stable families to have infinite configurations, as originally [23, 24].

- Complete: $\forall Z \subseteq \mathcal{B}.\ Z$ *is compatible* $\implies \bigcup Z \in \mathcal{B}$ *;*
- Stable: $\forall Z \subseteq \mathcal{B}.\ Z \neq \varnothing$ & Z *is compatible* $\implies \bigcap Z \in \mathcal{B}$*;*
- Finitary: $\forall \theta \in \mathcal{B}, (a,b) \in \theta \exists \theta_0 \in \mathcal{B}.\ \theta_0$ *is finite* & $(a,b) \in \theta_0 \subseteq \theta$*; and*
- Coincidence-free: *For all* $\theta \in \mathcal{B}$, $(a,b),(a',b') \in \theta$ *with* $(a,b) \neq (a',b')$,

$$\exists \theta_0 \in \mathcal{B}.\ \theta_0 \subseteq \theta \ \&\ ((a,b) \in \theta_0 \iff (a',b') \notin \theta_0)\,.$$

We now apply a general construction $\mathrm{Pr}(\mathcal{B})$ for obtaining an event structure from the stable family $\mathcal{B}$. Suppose $(a,b) \in \theta$ where $\theta \in \mathcal{B}$ [23, 24]. Because $\mathcal{B}$ is a stable family

$$[(a,b)]_\theta =_{\mathrm{def}} \bigcap \{\phi \in \mathcal{B} \mid \phi \subseteq \theta \ \&\ (a,b) \in \phi\} \in \mathcal{B}$$

and moreover is a finite set; it represents a minimal way in which (a,b) can occur. We build the pullback of event structures taking such minimal elements as events.

Proposition 3.3. *Defining* $\mathrm{Pr}(\mathcal{B}) = (P, \mathrm{Con}, \leq)$ *where:*

$$\begin{aligned}
&P = \{[(a,b)]_\theta \mid (a,b) \in \theta \ \&\ \theta \in \mathcal{B}\}\ ,\\
&Z \in \mathrm{Con} \ \textit{iff}\ Z \subseteq P \ \&\ \bigcup Z \in \mathcal{B} \quad \textit{and}\\
&p \leq p' \ \textit{iff}\ p, p' \in P \ \&\ p \subseteq p'
\end{aligned}$$

yields an event structure. There is an order isomorphism

$$\beta : (\mathcal{C}(\mathrm{Pr}(\mathcal{B})), \subseteq) \cong (\mathcal{B}, \subseteq)$$

where $\beta(y) = \bigcup y$ *for* $y \in \mathcal{C}(\mathrm{Pr}(\mathcal{B}))$*; its mutual inverse is* γ *where* $\gamma(\theta) = \{[(a,b)]_\theta \mid (a,b) \in \theta\}$ *for* $\theta \in \mathcal{B}$*.*

There are obvious maps $\pi_1 : \mathrm{Pr}(\mathcal{B}) \to A$ and $\pi_2 : \mathrm{Pr}(\mathcal{B}) \to B$ given by $\pi_1([(a,b)]_\theta) = a$ and $\pi_2([(a,b)]_\theta) = b$. These make the required pullback $\mathrm{Pr}(\mathcal{B})$, π_1, π_2 of event structures. Why? The family $\mathcal{B}$ is a pullback in the category of stable families (its maps are similar to those of event structures). There is a coreflection from the category of event structures to that of stable families. Its right adjoint is Pr which consequently preserves pullbacks, yielding the pullback of event structures when applied to $\mathcal{B}$ [23, 25].

Definition 3.4. We shall write $x \wedge y$ for the configuration $\gamma(\theta_{x,y})$ of $\mathrm{Pr}(\mathcal{B})$ which correponds to a secured bijection $\theta_{x,y} : x \cong fx = gy \cong y$ between $x \in \mathcal{C}^\infty(A)$ and $y \in \mathcal{C}^\infty(B)$. Note that any configuration of the pullback is of the form $x \wedge y$ for unique $x \in \mathcal{C}^\infty(A)$ and $y \in \mathcal{C}^\infty(B)$. Of course, given $x \in \mathcal{C}^\infty(A)$ and $y \in \mathcal{C}^\infty(B)$ we cannot be assured that they form a secured bijection even when $fx = gy$. We shall treat $\wedge$ as a partial operation with $x \wedge y$ only defined when $x \in \mathcal{C}^\infty(A)$ and $y \in \mathcal{C}^\infty(B)$ form a secured bijection.

4 Rigid image

This section is only used late on in the paper when showing how 'may' and 'must' behaviour transfer to the rigid image of a strategy—Section 11.

There is an adjunction between $\mathcal{E}_r$, the category of event structures with rigid maps, to $\mathcal{E}_t$, the category of event structures with total maps. Its right adjoint's action on an event structure B is given as follows. For $x \in \mathcal{C}^\infty(B)$, an *augmentation* of x is a finitary partial order (x, α) where $\forall b, b' \in x.\ b \leq_B b' \implies b \,\alpha\, b'$. We can regard such augmentations as elementary event structures in which all subsets of events are consistent. Order all augmentations by taking $(x, \alpha) \hookrightarrow (x', \alpha')$ iff $x \subseteq x'$ and the inclusion $i : x \hookrightarrow x'$ is a rigid map $i : (x, \alpha) \to (x', \alpha')$. Augmentations under $\hookrightarrow$ form a prime algebraic domain [11, 26], so are isomorphic to the configurations of an event structure, $aug(B)$; its events are the complete primes, which are precisely the augmentations with a top element.

Proposition 4.1. *[27] The inclusion functor $\mathcal{E}_r \hookrightarrow \mathcal{E}_t$ has a right adjoint aug. The category $\mathcal{E}_t$ is isomorphic to the Kleisli category of the monad induced on $\mathcal{E}_r$ by the adjunction.*

Rigid maps $f : A \to B$ have a useful image given by restricting the causal dependency of B to the set of events fA, the direct image of the events of A, and taking a finite set of events to be consistent if they are the image of a consistent set in A. More generally, a total map $f : A \to B$ has a *rigid image* given by the image of its corresponding Kleisli map, the rigid map $\bar{f} : A \to aug(B)$. Put more directly, a total map $f : A \to B$ has a *rigid image* comprising a factorisation $f = f_1 f_0$ where f_0 is rigid epi and f_1 is a total map,

$$\begin{tikzcd} A \arrow[r, two heads, "f_0"] \arrow[dr, "f"'] & B_0 \arrow[d, "f_1"] \\ & B \end{tikzcd},$$

with the following universal property: for any factorisation of $f = f_1' f_0'$ where f_0' is rigid epi, there is a unique map h such that the diagram

$$\begin{tikzcd} A \arrow[r, two heads, "f_0'"] \arrow[rr, bend left, two heads, "f_0"] \arrow[dr, "f"'] & B_0' \arrow[r, dashed, "h"] \arrow[d, dashed, "f_1'"] & B_0 \arrow[dl, "f_1"] \\ & B & \end{tikzcd}$$

commutes; the map h is necessarily also rigid and epi.

From the universal property of rigid image we derive:

Proposition 4.2. *Let $f : A \to B$ and $g : B \to C$ be maps of event structures. Assume that f is rigid and epi. Then, g and $g \circ f$ have the same rigid image.*

5 Event structures with polarity

Both games and strategies will be represented by event structures with polarity. An *event structure with polarity* comprises (A, pol) where A is an event structure with a polarity function $pol_A : A \to \{+, -, 0\}$ ascribing a polarity + (Player), − (Opponent) or 0 (neutral) to its events. The events correspond to (occurrences of) moves. It will be technically useful to allow events of neutral polarity; they arise, for example, in the interaction between a strategy and a counterstrategy. We write A^0 for the event structure with polarity in which all the polarities are reassigned 0, so made neutral. A *game* shall be represented by an event structure with polarity in which no moves are neutral.

Notation 5.1. In an event structure with polarity (A, pol), with configurations x and y, write $x \subseteq^- y$ to mean inclusion in which all the intervening events are moves of Opponent, *i.e.* $pol(y \smallsetminus x) \subseteq \{-\}$. Similarly, $x \subseteq^0 y$ signifies an inclusion in which all the intervening moves are neutral. However, we shall write $x \subseteq^+ y$ for inclusion in which the intervening events are either neutral or moves of Player. (The latter choice reflects the fact that neutral moves in a strategy behave as internal moves of Player.) We say a configuration $x \in \mathcal{C}^\infty(A)$ is *+-maximal* iff x is maximal in $\mathcal{C}^\infty(A)$ w.r.t. $\subseteq^+$, *i.e.* the only way that x extends to a larger configuration is through the occurrence of Opponent moves.

5.1 Operations on games

We introduce two fundamental operations on games.

5.1.1 Dual

The *dual*, $A^\perp$, of a game A, comprises the same underlying event structure as A but with a reversal of polarities. As mentioned in the introduction, we shall implicitly adopt the view of Player and understand a strategy in a game A as strategy for Player. A counterstrategy in a game A is a strategy for Opponent in the game A, *i.e.* a strategy (for Player) in the game $A^\perp$.

5.1.2 Simple parallel composition

This operation simply juxtaposes two games, and more generally two event structures with polarity. Let $(A, \leq_A, \mathrm{Con}_A, pol_A)$ and $(B, \leq_B, \mathrm{Con}_B, pol_B)$ be event structures with polarity. The events of $A \| B$ are $(\{1\} \times A) \cup (\{2\} \times B)$, their polarities unchanged, with the only relations of causal dependency given by $(1, a) \leq (1, a')$ iff $a \leq_A a'$ and $(2, b) \leq (2, b')$ iff $b \leq_B b'$; a finite set X of events is consistent in $A \| B$ iff its components X_A in A and and X_B in B are individually consistent. The unit w.r.t. simple composition is the empty event structure with polarity, written $\varnothing$. We shall adopt the same operation for configurations of a game $A \| B$, regarding a configuration x of the parallel composition as $x_A \| x_B$.

If we are not a little careful we can run into distracting technical issues through $(A \| B) \| C$ not being strictly the same as $A \| (B \| C)$. For our purposes

it will suffice to adopt the convention that when we write *e.g.* $A \| B \| C$ the simple parallel composition of three event structures with polarity we shall mean the event structure with events

$$\{1\} \times A \cup \{2\} \times B \cup \{3\} \times C\,,$$

with causal dependency and consistency copied from those of A, B and C. As in the binary case, we adopt the same notation for configurations and can describe a typical configuration x of $A \| B \| C$ as $x_A \| x_B \| x_C$.

5.2 Strategies between games

A strategy *from* a game A *to* a game B is a strategy in the compound game $A^\perp \| B$. Of course we shall have to define what it means to be a strategy in a game. Given another strategy τ from the game B to a game C, informally we obtain their composition $\tau \odot \sigma$ from A to C by playing the two strategies off against each other in the common game B and hiding the resulting interaction.

The composition of strategies can introduce hidden deadlocks, conflicts and divergences which affect its observable behaviour:

Example 5.2. Let B be the game consisting of two concurrent Player events b_1 and b_2, and C the game with a single Player event c. We illustrate the composition of two strategies σ_1 and σ_2 from the empty game $\varnothing$ to B, with τ from B to C. The strategy σ_1 in the game $\varnothing^\perp \| B$ nondeterministically plays b_1 or b_2. The strategy σ_2 also in the game $\varnothing^\perp \| B$ just plays b_2. The strategy τ in the game $B^\perp \| C$ does nothing if just b_1 is played and plays the single Player event c of C if b_2 is played. The composition $\tau \odot \sigma_1$ in the game $\varnothing^\perp \| C$ may play c or not according as σ_1 plays b_1 or b_2. The composition $\tau \odot \sigma_2$ also in the game $\varnothing^\perp \| C$ must play c. But the two compositions $\tau \odot \sigma_1$ and $\tau \odot \sigma_2$ are indistinguishable once the interaction over the common game B is hidden. $\square$

If we are to distinguish the two compositions of the example, we need to take some account of their internal moves of interaction.

6 Strategies with neutral moves

Thus motivated, we study *bare strategies* with neutral moves, in which we can see the events of interaction not visible in the game. Recall we assume that in games all events have +ve or −ve polarity.

Definition 6.1. A *bare strategy* from a game A to a game B comprises a total map $\sigma : S \to A^\perp \| N \| B$ of event structures with polarity (in which S may also have neutral events) where

(i) N is an event structure consisting solely of neutral events;

(ii) σ is *receptive*,
$\forall x \in \mathcal{C}(S), y \in \mathcal{C}(A^\perp \| N \| B).\ \sigma x \subseteq^- y \implies \exists! x' \in \mathcal{C}(S).\ x \subseteq x'\ \&\ \sigma x' = y\,;$

(iii) σ is innocent in that it is both $+$-innocent and $-$-innocent:
$+$-innocent: if $s \rightarrowtriangle s'$ & $pol(s) = +$ then $\sigma(s) \rightarrowtriangle \sigma(s')$;
$-$-innocent: if $s \rightarrowtriangle s'$ & $pol(s') = -$ then $\sigma(s) \rightarrowtriangle \sigma(s')$.

Note that s' in $+$-innocence and s in $-$-innocence may be neutral events.[2]

A *strategy* from a game A to a game B comprises a total map $\sigma : S \to A^{\perp} \| B$ of event structures with polarity for which the composite $\sigma : S \to A^{\perp} \| B \cong A^{\perp} \| \varnothing \| B$ is a bare strategy [6].

We shall often identify strategies with bare strategies with no neutral events, and (bare) strategies in a game with (bare) strategies from the empty game $\varnothing$.

Consider two bare strategies $\sigma : S \to A^{\perp} \| N \| B$ and $\sigma' : S' \to A^{\perp} \| N \| B$. A map between them, a 2-cell $f : \sigma \Rightarrow \sigma'$, comprises a map $f : S \to S'$ of event structures with polarity such that

$$\begin{tikzcd} S \arrow[r, "f"] \arrow[d, "\sigma"'] & S' \arrow[dl, "\sigma'"] \\ A^{\perp} \| N \| B & \end{tikzcd}$$

commutes. In this way bare strategies in $A^{\perp} \| N \| B$ form a category,

$$\mathbf{BStrat}(A, N, B) .$$

We obtain the category $\mathbf{Strat}(A, B)$ of strategies from A to B from the special case when $N = \varnothing$.

6.1 Strategies from bare strategies

We obtain a strategy as the visible part of a bare strategy when we hide neutral events via a projection p:

Proposition 6.2. *Let $\sigma : S \to A^{\perp} \| N \| B$ be a bare strategy—so satisfying properties (i), (ii) and (iii) of Definition 6.1. Then, σ satisfies an additional property:*

(iv) in the partial-total factorisation of the composition of σ with the projection $A^{\perp} \| N \| B \to A^{\perp} \| B$,

$$\begin{tikzcd} S \arrow[r, "p"] \arrow[d, "\sigma"'] & S_{\downarrow} \arrow[d, "\sigma_{\downarrow}"] \\ A^{\perp} \| N \| B \arrow[r] & A^{\perp} \| B \end{tikzcd}$$

the defined part $\sigma_{\downarrow}$ is a strategy, which we call the visible part *of σ.*

[2]This definition of *linear innocence*, which applies in the presence of neutral events, appears in the work of Claudia Faggian and Mauro Piccolo [10]. It is not to be confused with the innocence of Martin Hyland and Luke Ong, to which it only relates indirectly; to disambiguate the two notions "courtesy" has been used for that here. An extension of Hyland-Ong innocence to concurrent games is given in [28]. Bare strategies have also been called "partial" strategies [16] and "uncovered" strategies [15].

(Conversely, (iv) together with receptivity and no incidence of a +ve event immediately preceding a neutral event in S, suffice to establish that σ is a bare strategy.)

With the notation of the lemma above, write

$$x_{\downarrow} =_{\mathrm{def}} px \in \mathcal{C}^{\infty}(S_{\downarrow})$$

for the visible image of a configuration $x \in \mathcal{C}^{\infty}(S)$. The hiding operation on strategies extends to a functor

$$(_)_{\downarrow} : \mathbf{BStrat}(A, N, B) \to \mathbf{Strat}(A, B)\,;$$

a 2-cell $f : \sigma \Rightarrow \sigma'$ between bare strategies restricts to a 2-cell $f_{\downarrow} : \sigma_{\downarrow} \Rightarrow \sigma'_{\downarrow}$ between their visible parts. It acts so

$$f_{\downarrow} x_{\downarrow} = (fx)_{\downarrow}$$

for all $x \in \mathcal{C}^{\infty}(S)$.

6.2 Composition

We can compose two bare strategies

$$\sigma : S \to A^{\perp} \| M \| B \quad \text{and} \quad \tau : T \to B^{\perp} \| N \| C$$

by pullback. Ignoring polarities temporarily, and padding with identity maps, we obtain $\tau \otimes \sigma$ via the pullback

$$\begin{tikzcd}
 & T \otimes S \arrow[dl] \arrow[dr] \arrow[d, phantom, "\vee"] & \\
S \| N \| C \arrow[dr, "\sigma \| N \| C"'] & & A \| M \| T \arrow[dl, "A \| M \| \tau"] \\
 & A \| M \| B \| N \| C &
\end{tikzcd}$$

as the ensuing map

$$\tau \otimes \sigma : T \otimes S \to A^{\perp} \| (M \| B^{0} \| N) \| C$$

once we reinstate polarities and make the events of B neutral.

As a pullback the configurations of $T \otimes S$ are built from configurations of S and T. Let $x \in \mathcal{C}^{\infty}(S)$ and $y \in \mathcal{C}^{\infty}(T)$. Let $\sigma x = x_{A^{\perp}} \| x_0 \| x_B$ and $\tau y = y_{B^{\perp}} \| y_0 \| y_C$. Define

$$y \otimes x =_{\mathrm{def}} (x \| y_0 \| y_C) \wedge (x_{A^{\perp}} \| x_0 \| y)$$

which will be defined and a configuration in $\mathcal{C}^{\infty}(T \otimes S)$ if $x_B = y_{B^{\perp}}$ and the corresponding bijection secured. The following property, useful later, is a consequence of the receptivity of σ and τ.

Lemma 6.3. *A configuration $y \otimes x \in \mathcal{C}^\infty(T \otimes S)$ is +-maximal in $\mathcal{C}^\infty(T \otimes S)$ iff x is +-maximal in $\mathcal{C}^\infty(S)$ and y is +-maximal in $\mathcal{C}^\infty(T)$.*

Given a 2-cell $f : \sigma \Rightarrow \sigma'$ between bare strategies in $A^\perp \| N \| B$ and $g : \tau \Rightarrow \tau'$ between bare strategies in $B^\perp \| M \| C$, from the universality of pullback we obtain the 2-cell

$$g \otimes f : \tau \otimes \sigma \Rightarrow \tau' \otimes \sigma'$$

between the two compositions in $A^\perp \| (N \| B^0 \| M) \| C$. It acts so

$$(g \otimes f)(y \otimes x) = (gy) \otimes (fx)$$

on a typical configuration $y \otimes x$. This extends composition of bare strategies to a functor

$$\otimes : \mathbf{BStrat}(B, N, C) \times \mathbf{BStrat}(A, M, B) \to \mathbf{BStrat}(A, M \| B^0 \| N, C)\,.$$

Composition of bare strategies restricts to a functor between rigid 2-cells.

We obtain the composition of strategies as the composite functor

$$\begin{aligned} \odot : \mathbf{Strat}(B, C) \times \mathbf{Strat}(A, B) &\cong \mathbf{BStrat}(B, \varnothing, C) \times \mathbf{BStrat}(A, \varnothing, B) \\ &\xrightarrow{\otimes} \mathbf{BStrat}(A, \varnothing \| B^0 \| \varnothing, C) \xrightarrow{(\text{-})_\downarrow} \mathbf{Strat}(A, C)\,. \end{aligned}$$

Though we generally elide the isomorphisms regarding strategies as bare strategies without neutral events, and write

$$\tau \odot \sigma =_{\text{def}} (\tau \otimes \sigma)_\downarrow$$

for the composition of strategies $\sigma \in \mathbf{Strat}(A, B)$ and $\tau \in \mathbf{Strat}(B, C)$. Describing the strategies $\sigma : S \to A^\perp \| B$ and $\tau : T \to B^\perp \| C$ as having composition

$$\tau \odot \sigma : T \odot S \to A^\perp \| C\,,$$

we can present a typical configuration of $T \odot S$ as

$$y \odot x =_{\text{def}} (y \otimes x)_\downarrow$$

for $x \in \mathcal{C}^\infty(S)$ and $y \in \mathcal{C}^\infty(T)$.

Composition is preserved in extracting the visible part from bare strategies:

Lemma 6.4. *Let $\sigma : S \to A^\perp \| M \| B$ and $\tau : T \to B^\perp \| N \| C$ be bare strategies. Then,*

$$(\tau \otimes \sigma)_\downarrow = \tau_\downarrow \odot \sigma_\downarrow\,.$$

For $x \in \mathcal{C}^\infty(S)$ and $y \in \mathcal{C}^\infty(T)$,

$$(y \otimes x)_\downarrow = y_\downarrow \odot x_\downarrow\,,$$

with one side defined if the other is.

6.3 The copycat strategy

The copycat strategy is the identity for composition of strategies. We present its construction and key property from [6].

Lemma 6.5. *Let A be an event structure with polarity. There is an event structure with polarity $\mathbb{CC}_A$ having the same events and polarity as $A^\perp \| A$ but with causal dependency $\leq_{\mathbb{CC}_A}$ given as the transitive closure of the relation*

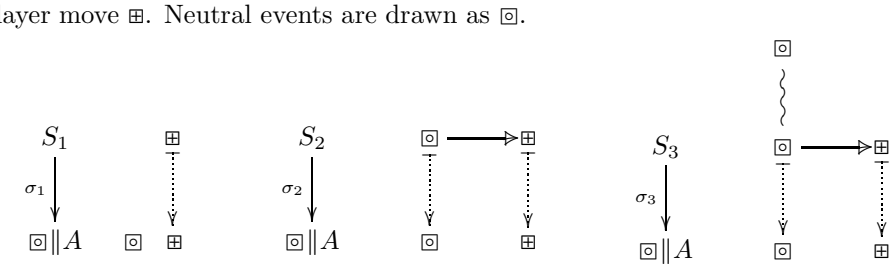
$$\leq_{A^\perp \| A} \cup \{(\bar{c}, c) \mid c \in A^\perp \| A \ \&\ pol_{A^\perp \| A}(c) = +\}$$

and finite subsets of $\mathbb{CC}_A$ consistent if their down-closure w.r.t. $\leq_{\mathbb{CC}_A}$ are consistent in $A^\perp \| A$. (For $c \in A^\perp \| A$ we use $\bar{c}$ to mean the corresponding copy of c, of opposite polarity, in the alternative component, i.e. $\overline{(1,a)} = (2,a)$ and $\overline{(2,a)} = (1,a)$.)
The configurations of $\mathbb{CC}_A$ have the form $x \| y$ where $y \sqsubseteq_A x$, i.e. $y \sqsupseteq^- x \cap y \subseteq^+ x$, for $x, y \in \mathcal{C}^\infty(A)$. (The relation $\sqsubseteq_A$ is a partial order, called the Scott order *[29].) The* copycat *strategy for A is the map $\mathbb{cc}_A : \mathbb{CC}_A \to A^\perp \| A$ which acts as identity on events. We have $\sigma \cong \sigma \odot \mathbb{cc}_A \cong \mathbb{cc}_B \odot \sigma$, for any strategy $\sigma \in \mathbf{Strat}(A, B)$.*

The axioms on strategies are precisely those needed to ensure that copycat behaves as identity w.r.t. composition $\odot$ and thus obtain a bicategory **Strat** of games and strategies [6]. Of course copycat is not the identity for the composition of bare strategies; that composition will generally have extra neutral events introduced through interactions.

7 'May' and 'must' tests

Consider the following three bare strategies in the game A comprising a single Player move $\boxplus$. Neutral events are drawn as $\boxdot$.



From the point of view of observing the move over the game A the first two bare strategies, σ_1 and σ_2, differ from the the third, σ_3. In a maximal play both σ_1 and σ_2 must result in the observation of the single move of A. However, in σ_3 one maximal play is that in which the topmost neutral event of S_3 has occurred, in conflict with the only way of observing the single move of A.

We follow Hennessy and de Nicola in making these ideas precise [5].

Definition 7.1. Let σ be a bare strategy in a game A. Let $\tau : T \to A^\perp \| N \| \boxplus$ be a 'test' bare strategy from A to the game consisting of a single Player move $\boxplus$. Write $\checkmark =_{\text{def}} (3, \boxplus)$.

Say σ *may pass* τ iff there exists $y \otimes x \in \mathcal{C}^\infty(T \otimes S)$, where $x \in \mathcal{C}^\infty(S)$ and $y \in \mathcal{C}^\infty(T)$, with the image τy containing $\checkmark$. (Note that we may w.l.o.g. assume that the configuration $y \otimes x$ is finite.)

Say σ *must pass* τ iff for all $y \otimes x \in \mathcal{C}^\infty(T \otimes S)$, where $x \in \mathcal{C}^\infty(S)$ and $y \in \mathcal{C}^\infty(T)$ are $\subseteq^+$-maximal, the image τy contains $\checkmark$.

Say two bare strategies are *'may'* (respectively, *'must') equivalent* iff the tests they may (respectively, must) pass are the same.

The definitions extend in the obvious fashion to bare strategies of type $A^\perp \| N \| B$.

A bare strategy is 'may' equivalent, but need not be 'must' equivalent, to the strategy which is its defined part; 'must' inequivalence is lost in moving from bare strategies to strategies.

Example 7.2. As an illustration of the subtle nature of testing for 'must' equivalence, consider the following bare strategies in the game A, as drawn:

$$\begin{tikzcd} a & b & c & d & e \\ \boxplus \arrow[r] & \boxminus \arrow[r] & \boxplus \arrow[r] & \boxminus \arrow[r] & \boxplus \end{tikzcd}$$

The game A consists of five events with the polarity and causal dependency shown. One bare strategy σ_1 is

$$\begin{tikzcd} \boxplus \arrow[r] & \boxminus \arrow[r] & \boxplus \arrow[r] \arrow[d, squiggly] & \boxminus \arrow[r] & \boxplus \\ & & \boxdot, & & \end{tikzcd}$$

with one neutral event, while the other σ_2 is

$$\begin{tikzcd} \boxplus \arrow[r] & \boxminus \arrow[r] & \boxplus \arrow[r] \arrow[d, squiggly] & \boxminus \arrow[r] & \boxplus \arrow[d, squiggly] \\ & & \boxdot & & \boxdot, \end{tikzcd}$$

with two neutral events. Through the possible occurrence of neutral events the bare strategy σ_1 has a $+$-maximal configuration with just events $\{a, b\}$ visible from the game, while σ_2 has in addition a $+$-maximal configuration comprising visible moves $\{a, b, c, d\}$. The following test strategy distinguishes σ_1 and σ_2

$$\begin{tikzcd} \boxminus \arrow[r] \arrow[d] & \boxplus \arrow[r] & \boxminus \arrow[r] & \boxplus \arrow[r] \arrow[dll, squiggly] & \boxminus \arrow[d] \\ \boxplus\checkmark & & & & \boxplus\checkmark\,. \end{tikzcd}$$

in that σ_1 must pass the test while σ_2 need not. (The test is a strategy in $A^\perp \| \boxplus\checkmark$; thus the change in polarity of moves in A.) Composed with the test, σ_1 may fail to perform the leftmost $\boxplus\checkmark$ but if so must then perform the rightmost $\boxplus\checkmark$; whereas σ_2 may fail to perform both. $\square$

This example might be puzzling to readers familiar with Harmer and McCusker's fully abstract model for (sequential) finite non-determinism [17], in particular regarding their handling of 'must' equivalence through the addition of *divergences*. Indeed, there, if a trace has already potentially triggered a divergence, then any execution going past that divergence has already lost all hope for 'must' convergence. Accordingly, in [17], only the first divergence matters—further divergences are not recorded.

In contrast, Example 7.2 indicates how in concurrent strategies, all divergences matter. This is not an artificiality of concurrent strategies, but something inherent to the observational power of the tests they support: here the distinguishing tests manage to observe beyond the first divergence by running two threads in parallel. The first thread aims to directly register success; by +-maximality it will be run eventually if the observed strategy triggers the first divergence. The second thread follows the execution of the program, then immediately cancels the first thread if the program goes past the divergence; and then proceeds to test for the second divergence.

8 Strategies with stopping configurations

Bare strategies lack identities w.r.t. composition, so they do not form a bicategory. Fortunately, for 'may' and 'must' equivalence it is not necessary to use bare strategies; for 'may' equivalence strategies suffice; whereas for 'must' equivalence it is sufficient to carry with a strategy the extra structure of *stopping configurations*—to be thought of as images of +-maximal configurations in an underlying bare strategy. As we shall see, composition and copycat extend to composition and copycat on strategies with stopping configurations, while maintaining a bicategory. We tackle the simpler case in which games are assumed to be race-free. (The extension to games which are not race-free is outlined in [16].) We recall when an event structure with polarity is race-free and the allied notion of deterministic strategy:

Definition 8.1. Say A, an event structure with polarity, is *race-free* iff whenever $x \subseteq^+ y$ and $x \subseteq^- z$ for configurations x, y, z of A then $y \cup z$ is also a configuration.

Say S, an event structure with polarity, is *deterministic* iff whenever $x \subseteq^+ y$ and $x \subseteq z$ for configurations x, y, z of S then $y \cup z$ is also a configuration. Say a bare strategy $\sigma : S \to A^\perp \| N \| B$ is *deterministic* iff S is deterministic; with a strategy being deterministic iff it is so as a bare strategy.

Lemma 8.2. *[6, 30] Let A be a game. The copycat strategy æ_A is deterministic iff the game A is race-free.*

Let $\sigma : S \to A^\perp \| N \| B$ be a bare strategy between race-free games A and B.

Recall its associated partial-total factorisation

$$\begin{tikzcd} S \arrow[r, "p"] \arrow[d, "\sigma"'] & S_{\downarrow} \arrow[d, "\sigma_{\downarrow}"] \\ A^{\perp} \| N \| B \arrow[r] & A^{\perp} \| B \end{tikzcd}$$

where p is a projection sending configurations x of S to configurations $x_{\downarrow}$ of $S_{\downarrow}$. The visible part of σ is a strategy $\sigma_{\downarrow}$. Define the *stopping* configurations in $\mathcal{C}^{\infty}(S_{\downarrow})$ to be

$$\mathrm{Stop}(\sigma) =_{\mathrm{def}} \{x_{\downarrow} \mid x \in \mathcal{C}^{\infty}(S) \text{ is } +\text{-maximal}\}\,.$$

So, in other words, the stopping configurations are the visible images of configurations which are maximal w.r.t. neutral or Player moves. Note that $\mathrm{Stop}(\sigma)$ will include all the $+$-maximal configurations of $S_{\downarrow}$: any $+$-maximal configuration y of $S_{\downarrow}$ is the image under p of its down-closure $[y]$ in S, and by Zorn's lemma this extends (necessarily by neutral events) to a maximal configuration x of S with image y under p; the configuration x is $+$-maximal by the $+$-maximality of y. If σ is deterministic, then $\mathrm{Stop}(\sigma)$ consists of precisely the $+$-maximal configurations of $S_{\downarrow}$. If σ is a strategy, *i.e.* it has no neutral events, then $\mathrm{Stop}(\sigma)$ is just the set consisting of all $+$-maximal configurations of S.

Definition 8.3. A *stopping strategy* in a game A comprises (σ, M_S), a strategy $\sigma : S \to A$ together with a subset $M_S \subseteq \mathcal{C}^{\infty}(S)$ called *stopping configurations*. As usual, a stopping strategy from a game A to game B is a stopping strategy in the game $A^{\perp} \| B$. We let $\mathrm{St} : \sigma \mapsto (\sigma_{\downarrow}, \mathrm{Stop}(\sigma))$ denote the operation motivated above from bare strategies σ to stopping strategies.

Remark. There is the issue of what axioms to adopt on stopping configurations. We do not insist that stopping configurations include all $+$-maximal configurations as this property will not be preserved in taking the rigid image of a stopping strategy—Example 11.3.

Given two stopping strategies $\sigma : S \to A^{\perp} \| B$, M_S and $\tau : T \to B^{\perp} \| C$, M_T we define their *interaction*,

$$(\tau, M_T) \circledast (\sigma, M_S) =_{\mathrm{def}} (\tau \circledast \sigma, M_T \circledast M_S)\,,$$

with the stopping configurations of the interaction $T \circledast S$ as

$$M_T \circledast M_S = \{y \circledast x \mid x \in M_S \ \&\ y \in M_T\}$$

—sensible because of Lemma 6.3. Define their *composition* by

$$(\tau, M_T) \odot (\sigma, M_S) =_{\mathrm{def}} (\tau \odot \sigma, M_T \odot M_S)\,,$$

where the stopping configurations of $T \odot S$ form the set

$$M_T \odot M_S = \{y \odot x \mid x \in M_S \ \&\ y \in M_T\}\,.$$

To make stopping strategies into a bicategory we must settle on an appropriate notion of 2-cell. The following choice of definition is useful for 'must' equivalence—see Lemma 9.5.

Definition 8.4. A 2-cell $f : (\sigma, M_S) \Rightarrow (\sigma', M_{S'})$ between stopping strategies is a 2-cell of strategies $f : \sigma \Rightarrow \sigma'$ such that $fM_S \subseteq M_{S'}$. We write $\mathbf{SStrat}(A, B)$ for the category of stopping strategies from game A to game B; its maps are 2-cells.

Composition extends to 2-cells between stopping strategies: their composition as 2-cells between strategies is easily shown to preserve stopping configurations.

Proposition 8.5. *For games A, B and C composition of stopping strategies is a functor* $\odot : \mathbf{SStrat}(B, C) \times \mathbf{SStrat}(A, B) \to \mathbf{SStrat}(A, C)$.

We should also extend copycat $\mathrm{cc}_A : \mathrm{CC}_A \to A^\perp \| A$ to a stopping strategy. Because we are assuming A is race-free, we do this by taking

$$M_{\mathrm{CC}_A} =_{\mathrm{def}} \{(x \| x) \in \mathcal{C}^\infty(\mathrm{CC}_A) \mid x \in \mathcal{C}^\infty(A)\}.$$

Because A is race-free, M_{CC_A} comprises all the $+$-maximal configurations of CC_A. Then, $(\mathrm{cc}_A, M_{\mathrm{CC}_A})$ is an identity w.r.t. the extended composition.

With the operations and constructions above, stopping strategies inherit the structure of a bicategory **SStrat** from strategies; the objects are restricted to race-free games in order to have the above simple form of stopping configurations for copycat.

8.1 Bare strategies and stopping strategies

We turn to relations between bare strategies and stopping strategies. Recall from Definition 8.3 the operation

$$\mathrm{St} : \sigma \mapsto (\sigma_\downarrow, \mathrm{Stop}(\sigma))$$

which takes a bare strategy σ to a stopping strategy. It preserves composition:

Lemma 8.6. *Let $\sigma : S \to A^\perp \| M \| B$ and $\tau : T \to B^\perp \| N \| C$ be bare strategies. Then,*

$$St(\tau \otimes \sigma) = St(\tau) \odot St(\sigma).$$

Proof. By Lemma 6.4, it suffices to show

$$\mathrm{Stop}(\tau \otimes \sigma) = \mathrm{Stop}(\tau) \odot \mathrm{Stop}(\sigma).$$

Configurations of $\mathrm{Stop}(\tau \otimes \sigma)$ are of the form $(y \otimes x)_\downarrow$ where $y \otimes x$ is $+$-maximal for $x \in \mathcal{C}^\infty(S)$ and $y \in \mathcal{C}^\infty(T)$. By Lemma 6.3 these coincide with configurations $(y \otimes x)_\downarrow$ for which both x and y are $+$-maximal. Configurations of $\mathrm{Stop}(\tau) \odot \mathrm{Stop}(\sigma)$ take the form $y_\downarrow \odot x_\downarrow$ where $x \in \mathcal{C}^\infty(S)$ and $y \in \mathcal{C}^\infty(T)$ are $+$-maximal. But by Lemma 6.4, $(y \otimes x)_\downarrow = y_\downarrow \odot x_\downarrow$, ensuring the desired equality. ∎

We have seen that there is a functor $(_)_{\downarrow} : \mathbf{BStrat}(A, N, B) \to \mathbf{Strat}(A, B)$ from bare strategies to their visible part. However, it is not the case that St is a functor from bare strategies $\mathbf{BStrat}(A, N, B)$ to stopping strategies $\mathbf{SStrat}(A, B)$. Given an arbitrary 2-cell $f : \sigma \Rightarrow \sigma'$ between bare strategies $f_{\downarrow}$ can fail to preserve stopping configurations. However:

Proposition 8.7. *Let $\sigma : S \to A^{\perp} \| N \| B$ and $\sigma' : S' \to A^{\perp} \| N \| B$ be bare strategies. Say a 2-cell $f : \sigma \Rightarrow \sigma'$ is* +-reflecting *iff, for $x \in \mathcal{C}(S)$, $y \in \mathcal{C}(S')$,*

$$fx \subseteq^{+} y \implies \exists x' \in \mathcal{C}(S).\ x \subseteq x' \ \&\ fx' = y\,.$$

Let $\mathbf{BStrat}^{+}(A, N, B)$ be the subcategory where 2-cells are +-reflecting. Then

$$St : \mathbf{BStrat}^{+}(A, N, B) \to \mathbf{SStrat}(A, B)$$

taking $f : \sigma \Rightarrow \sigma'$ to $f_{\downarrow} : St(\sigma) \Rightarrow St(\sigma')$ is a functor.

9 'May' and 'Must' testing

We can rephrase 'may' and 'must' testing in terms of stopping strategies.

Definition 9.1. Let (σ, M_S) be a stopping strategy in a game A. Let $\tau : T \to A^{\perp} \| N \| \boxplus$ be a 'test' bare strategy from A to a the game consisting of a single Player move $\boxplus$. Write $St(\tau)$ as (τ_0, M_0) where $\tau_0 : T_0 \to A \| \boxplus$ is the visible part of τ and M_0 are its stopping configurations, obtained as images of the +-maximal configurations of T. Write $\checkmark =_{\text{def}} (2, \boxplus)$.

Say (σ, M_S) *may pass* τ iff there exists $y \otimes x \in \mathcal{C}^{\infty}(T_0 \otimes S)$, where $x \in \mathcal{C}^{\infty}(S)$ and $y \in \mathcal{C}^{\infty}(T_0)$, with the image $\tau_0 y$ containing $\checkmark$. (Note again, we may w.l.o.g. assume that the configurations x and y are finite.)

Say (σ, M_S) *must pass* τ iff for all $y \otimes x \in M_0 \otimes M_S$, where $x \in \mathcal{C}^{\infty}(S)$ and $y \in \mathcal{C}^{\infty}(T_0)$, the image $\tau_0 y$ contains $\checkmark$.

Say two stopping strategies are *'may'*, respectively *'must'*, *equivalent* iff the tests they may, respectively must, pass are the same.

Proposition 9.2. *With the notation above,*

(σ, M_S) may pass τ *iff there exists $y \odot x \in \mathcal{C}^{\infty}(T_0 \odot S)$, where $x \in \mathcal{C}^{\infty}(S)$ and $y \in \mathcal{C}^{\infty}(T_0)$, with the image $\tau_0 y$ containing $\checkmark$ —the configurations x, y may be assumed finite; and*

(σ, M_S) must pass τ *iff for all $y \odot x \in M_0 \odot M_S$, where $x \in M_S$ and $y \in M_0$, the image $\tau_0\ y$ contains $\checkmark$.*

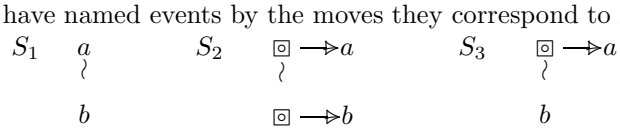
Lemma 9.3. *Let A be a race-free game. Let σ be a bare strategy in A. Then,*

σ may pass a test τ *iff* $St(\sigma)$ may pass τ*;*

σ must pass a test τ *iff* $St(\sigma)$ must pass τ*.*

Proof. Directly from the definitions, for the 'if' of the 'must' case, using Lemma 6.3. □

Example 9.4. It is tempting to think of neutral events as behaving like the internal "tau" events of CCS [3]. However, in the context of concurrent strategies, because of their asynchronous nature, they behave rather differently. Consider three bare strategies, over a game comprising of just two concurrent +ve events, say a and b. The bare strategies have the following event structures in which we have named events by the moves they correspond to in the game:



No pair would be weakly bisimilar due to the presence of pre-emptive internal events [3]. However, all three become isomorphic under St so are 'may' and 'must' equivalent to each other. □

2-cells between stopping strategies respect 'may' and 'must' behaviour in the sense of the following lemma.

Lemma 9.5. *Let $f : (\sigma, M_S) \Rightarrow (\sigma', M_{S'})$ be a 2-cell between stopping strategies. Then for any test τ,*

(σ, M_S) may pass τ implies $(\sigma', M_{S'})$ may pass τ; and

$(\sigma', M_{S'})$ must pass τ implies (σ, M_S) must pass τ.

Moreover, if f is a rigid epi and $fM_S = M_{S'}$, then (σ, M_S) and $(\sigma', M_{S'})$ are both 'may' and 'must' equivalent.

Proof. In this proof, we shall identify a test with its image (τ, M_T) under St, as a strategy $\tau : T \to A^\perp \| \boxplus$ with stopping configurations M_T.

Let $f : (\sigma, M_S) \Rightarrow (\sigma', M_{S'})$ be a 2-cell. Assume $\sigma : S \to A$ and $\sigma' : S' \to A$.

Suppose (σ, M_S) may pass (τ, M_T). Then there is a (finite) configuration which we write $y \otimes x$ of $T \otimes S$, built as a secured bijection out of $y \in \mathcal{C}(T)$ and $x \in \mathcal{C}(S)$, whose image in the game contains $\checkmark$. The secured bijection built out of y and x induces a secured bijection built out of y and fx; this is because fx has no more causal dependency than x with which it is in bijection. This determines a configuration $y \otimes fx$, with image containing $\checkmark$.

Suppose $(\sigma', M_{S'})$ must pass (τ, M_T). Any $y \otimes x \in M_T \otimes M_S$ images under $\tau \otimes f$ to $y \otimes fx \in M_T \otimes M_{S'}$. As $(\sigma', M_{S'})$ must pass τ, the configuration $y \otimes fx$ has image containing $\checkmark$, ensuring that $y \otimes x$ does too.

Finally suppose that f is rigid epi and $fM_S = M_{S'}$. We have just shown that f preserves the passing of 'may' tests and reflects the passing of 'must' tests. Because f is rigid epi it also reflects the passing of 'may' tests. Because f is rigid and $fM_S = M_{S'}$ it preserves the passing of 'must' tests: any secured bijection $y \otimes fx$ in $M_T \otimes M_{S'}$ ensures by the rigidity of f a secured bijection $y \otimes x$ in $M_T \otimes M_S$; as (σ, M_S) must pass τ we have the image in the game of $y \otimes x$ contains $\checkmark$ ensuring the image of $y \otimes fx$ does too. □

Tests based on bare strategies are more discriminating than tests based on (pure) strategies:

Example 9.6. Let a game comprise a single Player move. Consider two stopping strategies:

σ_1, the empty strategy with the empty configuration $\varnothing$ as its single stopping configuration;

σ_2, the strategy performing the single Player move $\boxplus$ with stopping configurations $\varnothing$ and $\{\boxplus\}$. (We can easily realise this stopping strategy via St from a bare strategy with event structure $\boxplus \sim \boxdot$.)

By Lemma 9.5, we have $(\sigma_2, \{\varnothing, \{\boxplus\}\})$ must pass τ implies $(\sigma_1, \{\varnothing\})$ must pass τ, for any test τ. (The above would not hold if we had not included $\varnothing$ in the stopping configurations of σ_2.)

Using the fact that we need only consider rigid images of tests—shown later in Section 11, a little argument by cases establishes the converse implication too, provided we restrict just to tests which are strategies. The stopping strategies would be must equivalent w.r.t. tests based just on strategies.

However with tests based on bare strategies we can distinguish them. Consider the test τ comprising three events, one of them neutral, with only nontrivial causal dependency $\boxminus \rightarrowtriangle \boxdot$ and $\boxdot$ in conflict with the 'tick' event $\boxplus$. Then, it is not the case that $(\sigma_2, \{\varnothing, \{\boxplus\}\})$ must pass τ —the occurrence of the neutral event blocks success in a maximal execution—while $(\sigma_1, \{\varnothing\})$ must pass τ. Notice how the presence of the neutral event in the test turns the possibility that σ_2 can perform the Player move into a possibility of its failing the test. □

10 'May' and 'Must' behaviour characterised

10.1 Preliminaries, traces of a strategy

Let S be an event structure. A possibly infinite sequence

$$s_1, s_2, \cdots, s_n, \cdots$$

in S constitutes a *serialisation* of a configuration $x \in \mathcal{C}^\infty(S)$ if $x = \{s_1, s_2, \cdots, s_n, \cdots\}$ and $\{s_1, \cdots, s_i\} \in \mathcal{C}(S)$ for all i at which the sequence is defined. We will often identify such a countable enumeration of a set with its associated total order. Note that in this way we can regard a serialisation as an elementary event structure in which causal dependency takes the form of a total order; a serialisation of a configuration is associated with a map to S whose image is the configuration.

Let $\sigma : S \to A$ be a strategy in a game A. A *trace* in σ is a possibly infinite sequence

$$\alpha = (\sigma(s_1), \sigma(s_2), \cdots, \sigma(s_n), \cdots)$$

of events in A obtained from a serialisation

$$s_1, s_2, \cdots, s_n, \cdots$$

of a configuration $x \in \mathcal{C}^\infty(S)$. Clearly α is a serialisation of $\sigma x \in \mathcal{C}^\infty(A)$. From the local injectivity of σ, the configuration x will be finite/infinite according as the trace is finite/infinite. We say that α is a trace of the configuration x in σ, or that x has trace α in σ.

Proposition 10.1. *Let $\sigma : S \to A$ be a strategy.*
(i) Any countable configuration of S has a trace.
(ii) Let $x \in \mathcal{C}^\infty(S)$ and α be an enumeration

$$a_1, a_2, \cdots, a_n, \cdots$$

of σx. Then, α is a trace of x in σ iff for all $s, s' \in x$ if $s \rightarrowtriangle s'$ then $\sigma(s)$ precedes $\sigma(s')$ in the enumeration α.

Proof. (i) Let x be a countable configuration of S w.r.t. the strategy $\sigma : S \to A$. This follows because there is a serialisation $x = \{s_1, s_2, \cdots, s_n, \cdots\}$, in which $\{s_1, \cdots, s_i\}$ is down-closed in S at all i in the enumeration. To see this, from its countability we may assume a countable enumeration of x, which need not be a serialisation. Define $s_1 \in x$ to be the earliest event of the enumeration for which $[s_1) = \varnothing$ in S; such an s_1 is ensured to exist by the well-foundedness of causal dependency provided $x \neq \varnothing$. Inductively, define s_n to be the earliest event of the enumeration which is in $x \setminus \{s_1, \cdots, s_{n-1}\}$ and for which $[s_n) \subseteq \{s_1, \cdots, s_{n-1}\}$; again the well-foundedness of causal dependency ensures such an s_n exists provided $x \setminus \{s_1, \cdots, s_{n-1}\} \neq \varnothing$. It is elementary to check this provides a serialisation of x.

(ii) *"Only if":* Directly from the definition of trace of a configuration. *"If":* Via the local bijection between x and σx given by σ we obtain an enumeration

$$s_1, s_2, \cdots, s_n, \cdots$$

of x matching α in that $\sigma(s_i) = a_i$. The assumption that $s \rightarrowtriangle s'$ implies $\sigma(s)$ precedes $\sigma(s')$ in the enumeration α, entails $\{s_1, \cdots, s_i\} \in \mathcal{C}(S)$ for all i. Hence the enumeration of x is a serialisation making α a trace of x. □

Lemma 10.2. *Let $\sigma : S \to A$ be a strategy in a game A. Let $x \in \mathcal{C}^\infty(S)$. Let α be a serialisation of σx which is not a trace of $x \in \mathcal{C}^\infty(S)$. Then, there are $s, s' \in x$ with $pol(s) = -$ and $pol(s') = +$ and $s \rightarrowtriangle_S s'$ with (note the order reversal) $\sigma(s') \leq_\alpha \sigma(s)$ in α (regarded as a total order).*

Proof. By assumption, any trace of x differs from α. We deduce there is $s \rightarrowtriangle s'$ in x with $\sigma(s) \not\leq \sigma(s')$ in the total order of α; otherwise we could serialise x to obtain the trace α —Proposition 10.1(ii). Now, $\sigma(s) \not\leq_A \sigma(s')$ in A as any serialisation must respect the order $\leq_A$. Hence, by the innocence of σ, we must have $pol(s) = -$ and $pol(s') = +$. Because α is totally ordered, $\sigma(s') \leq \sigma(s)$ in α. □

10.2 Characterisation of the 'may' preorder

For stopping strategies (with games assumed race-free) we have:

Theorem 10.3. *Let (σ_1, M_1) and (σ_2, M_2) be stopping strategies in a common game. Then,*

(σ_1, M_1) may pass τ implies (σ_2, M_2) may pass τ, for all tests τ,
iff
all finite traces of σ_1 are traces of σ_2.

Proof. Assume strategies $\sigma_1 : S_1 \to A$ and $\sigma_2 : S_2 \to A$. *"if":* Assume all finite traces of σ_1 are traces of σ_2. Suppose (σ_1, M_1) may pass test τ with event structure T. Then there is a successful configuration $w \otimes x_1 \in \mathcal{C}(T \otimes S_1)$, where $x_1 \in \mathcal{C}(S_1)$ and $w \in \mathcal{C}(T)$; it is successful in the sense that its image contains the success event $\checkmark$. Take a serialisation of $w \otimes x_1$; this induces a serialisation of x_1 to yield a trace. Then, by assumption, σ_2 has a configuration $x_2 \in \mathcal{C}(S_2)$ with the same trace, so a matching serialisation. Consequently the pairing $w \otimes x_2$ is defined with $w \otimes x_2 \in \mathcal{C}(T \otimes S_2)$; sharing the same image as $w \otimes x_1$ it is also successful.

"only if": We show the contraposition: assuming not all traces of σ_1 are traces of σ_2, we produce a test τ for which σ_1 may pass τ while it is not the case that σ_2 may pass τ.

Assume a trace α_1 of $x_1 \in \mathcal{C}(S_1)$ is not a trace of any $x_2 \in \mathcal{C}(S_2)$. Note that the trace α_1, and correspondingly x_1, must have at least one +ve event as otherwise, by receptivity, σ_2 could match the trace α_1. Any trace of x_2, with $\sigma_2 x_2 = \sigma_1 x_1$, differs from α_1. By Lemma 10.2, we deduce there are $s, s' \in x_2$ such that $s \rightarrowtriangle_2 s'$ with $pol(s) = -$ and $pol(s') = +$ and $\sigma_2(s') \leq_1 \sigma_2(s)$ in the total order α_1.

Thus for each $x_2 \in \mathcal{C}(S_2)$ with $\sigma_2 x_2 = \sigma_1 x_1$ we can choose $\theta(x_2) = (s, s')$ so that $s \rightarrowtriangle_2 s'$ in x_2 with $pol(s) = -$ and $pol(s') = +$ and $\sigma_2(s') \leq_1 \sigma_2(s)$ in α_1.

We now describe a test $\tau : T \to A^\perp \| \boxplus$ which will discriminate between σ_1 and σ_2. Let T_1' be the elementary event structure comprising events $T_1 =_{\text{def}} \sigma_1 x_1$ saturated with all accessible Opponent moves (note, in $A^\perp$), *i.e.* events

$$T_1' = \{a \in A \mid pol_{A^\perp}([a] \smallsetminus T_1) \subseteq \{-\}\}$$

with order that of $A^\perp$ augmented with $\sigma_2(s') \leq_1 \sigma_2(s)$ for every choice $\theta(x_2) = (s, s')$ where $x_2 \in M_2$ and $\sigma_2 x_2 = \sigma_1 x_1$; the ensuing relation on T_1 is included in the total order α_1 so forms a partial order in which every element has only finitely many elements below it. (By design, T_1' "disagrees" with the causal dependency of each $x_2 \in \mathcal{C}(S_2)$ for which $\sigma_2 x_2 = \sigma_1 x_1$.) The polarities of events of T_1' are those of its events in $A^\perp$. On T_1' the map τ takes an event to its same event in $A^\perp$.

Let T be the event structure with polarity obtained from T_1' by adjoining a fresh 'success' event $\boxplus$ with additional causal dependency so $t_1 \leq_T \boxplus$ iff t_1 is $-$ve; as noted above there has to be at least one +ve event in x_1 and thus, by the reversal of polarity, at least one $t_1 \in T_1$ of $-$ve polarity. Then the obvious map $\tau : T \to A^\perp \| \boxplus$ is a strategy, and a suitable test for σ_1 and σ_2.

We have (i) σ_1 may pass τ, while (ii) it is not the case that σ_2 may pass τ.

To see (i), remark that the relation of causal dependency on T_1 is included in the the total order of the trace α_1 of x_1. Hence $\tau \otimes \sigma_1$ has a successful configuration $(T_1 \cup \{\boxplus\}) \otimes x_1$.

To show (ii), consider any finite configuration of $\tau \otimes \sigma_2$. It has the form $w \otimes x_2$ where $w \in \mathcal{C}(T)$ and $x_2 \in \mathcal{C}(S_2)$. The configuration $w \otimes x_2$ is unsuccessful because $\boxplus \notin w$, as we now show. By design, τ and σ_2 enforce opposing causal dependencies on a pair of synchronisations needed for $T_1 \otimes x_2$ to be defined whenever $x_2 \in \mathcal{C}(S_2)$ with $\sigma_2 x_2 = T_1$. At least two events of opposing polarity in T_1 are excluded from any pairing $w \otimes x_2$; one must be a $-$ve event of T_1 on which $\boxplus$ causally depends; hence $\boxplus \notin w$. □

That the characterisation of the 'may' preorder above only depends on finite traces is not surprising, and familiar from previous work; the full-abstraction results of Dan Ghica and Andrzej Murawski w.r.t. 'may' behaviour rely only on finite traces [8].

Clearly the proof above does not rely on stopping configurations or tests being bare rather than pure strategies; the test used in the proof patently has no neutral events. The extra discriminating power of tests based on bare strategies, illustrated in Example 9.6, does play an essential role in the analgous result in the 'must' case, to be considered now.

10.3 Characterisation of the 'must' preorder

Recall an event structure $E = (E, \leq, \mathrm{Con})$ is *consistent-countable* iff there is a function $\chi : E \to \omega$ from the events such that

$$\{e_1, e_2\} \in \mathrm{Con} \ \& \ \chi(e_1) = \chi(e_2) \implies e_1 = e_2 \,.$$

Any configuration $x \in \mathcal{C}^\infty(E)$ of a consistent-countable event structure E is countable and so may be serialised as

$$x = \{e_1, e_2, \cdots, e_n, \cdots\}$$

so that $\{e_1, \cdots, e_n\} \in \mathcal{C}(E)$ for any finite subsequence. For the must case we assume that games are consistent-countable. It follows that strategies $\sigma : S \to A$ in consistent-countable games A have S consistent-countable. W.r.t. such a strategy σ, we have traces of all configurations.

Theorem 10.4. *Assume game A is consistent-countable. Let (σ_1, M_1) and (σ_2, M_2) be stopping strategies in A. Then,*

(σ_2, M_2) must pass τ implies (σ_1, M_1) must pass τ, for all tests τ,

iff

all traces of stopping configurations M_1 are traces of stopping configurations M_2.

Proof. "if": Assume all traces of stopping configurations M_1 are traces of stopping configurations M_2. A stopping configuration of $\tau \otimes \sigma_1$ has the form $w \otimes x_1$ where w and x_1 are stopping configurations of τ and σ_1, respectively. A serialisation of $w \otimes x_1$ into a (possibly infinite) sequence induces a serialisation of $x_1 \in M_1$. By assumption, there is $x_2 \in M_2$ with the same trace in A as x_1. Consequently, $w \otimes x_2$ is a configuration of $\tau \otimes \sigma_2$ with the same image in $A \| \boxplus$.

Moreover, $w \circledast x_2$ is a stopping configuration of $\tau \circledast \sigma_2$. Supposing (σ_2, M_2) must pass a test τ, the image of $w \circledast x_2$ contains $\checkmark$ whence the image of $w \circledast x_1$ contains $\checkmark$ ensuring (σ_1, M_1) must pass a test τ.

"only if": We show the contraposition: assuming not all traces of stopping configurations M_1 are traces of stopping configurations M_2, we produce a test τ for which (σ_2, M_2) must pass τ while it is not the case that (σ_1, M_1) must pass τ.

Assume a trace α_1 of $x_1 \in M_1$ is not a trace of any $x_2 \in M_2$.

In particular, consider any $x_2 \in M_2$ with $\sigma_2 x_2 = \sigma_1 x_1$. Then, any trace of x_2 differs from α_1. By Lemma 10.2, there are $s, s' \in x_2$ such that $s \rightarrowtriangle_2 s'$ with $pol(s) = -$ and $pol(s') = +$ and $\sigma_2(s') \leq_1 \sigma_2(s)$ in the total order α_1.

Thus for each $x_2 \in M_2$ with $\sigma_2 x_2 = \sigma_1 x_1$ we can choose $\theta(x_2) = (s, s')$ so that $s \rightarrowtriangle_2 s'$ in x_2 with $pol(s) = -$ and $pol(s') = +$ and $\sigma_2(s') \leq_1 \sigma_2(s)$ in α_1.

We build an event structure with polarity T and a test as bare strategy $\tau : T \to A^\perp \| N \| \boxplus$. We build the events of T as $T_1' \cup N \cup T_2$, a union of sets of events, assumed disjoint, described as follows.

- Let T_1' be the elementary event structure comprising events $T_1 =_{\text{def}} \sigma_1 x_1$ saturated with all accessible Opponent moves, *i.e.* events

  $$T_1' = \{a \in A \mid pol_{A^\perp}([a] \smallsetminus T_1) \subseteq \{-\}\}$$

  with order that of A augmented with $\sigma_2(s') \leq_1 \sigma_2(s)$ for every choice $\theta(x_2) = (s, s')$ where $x_2 \in M_2$ and $\sigma_2 x_2 = \sigma_1 x_1$; the ensuing relation on T_1 is included in the total order α_1 so forms a partial order in which every element has only finitely many elements below it. (By design, T_1' "disagrees" with the causal dependency of each $x_2 \in M_2$ for which $\sigma_2 x_2 = \sigma_1 x_1$.) The polarities of events of T_1' are those of its events in $A^\perp$. On T_1' the map τ takes an event to its same event in $A^\perp$.

- N comprises a copy of the set of events of $-$ve polarity in T_1; all the events of N have neutral polarity; an event of N is sent by τ to its copy.

- T_2 comprises a copy of the set of events T_1; all the events of T_2 have $+$ve polarity; they are all sent by τ to $\checkmark =_{\text{def}} (3, \boxplus)$.

- Causal dependency on T is that of T_1' augmented with dependencies from events of T_1 of $-$ve polarity to their corresponding copies in N.

- The consistency relation of T is that minimal relation which ensures that: any two distinct events of T_2 are in conflict; a $+$ve event of T_1 conflicts with its corresponding copy in T_2; and a neutral event in N conflicts with its corresponding copy in T_2. Formally,

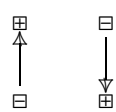
  $X \in \mathrm{Con}_T$ iff $X \subseteq_{\text{fin}} T_1 \cup N \cup T_2$ & $|X \cap T_2| \leq 1$ &
  $(\forall t_1 \in X \cap T_1^+, t_2 \in X \cap T_2.\ t_1, t_2$ are not copies of a common event) &
  $(\forall n \in X \cap N, t_2 \in X \cap T_2.\ n, t_2$ are not copies of a common event).

Note that all the events over $\checkmark$, which together comprise the set T_2, can occur initially but can become blocked as moves are made in T_1. In particular, the set $T_1 \cup N$ is a +-maximal configuration of T with image in $A^\perp \| N \| \boxplus$ not containing any event over $\checkmark$. On the other hand any +-maximal configuration of T not including all the events T_1 will contain an event over $\checkmark$. Hence $\mathrm{St}(\tau)$ has an unsuccessful stopping configuration consisting of precisely all the events of T_1—it does not have an event over $\checkmark$—while all stopping configurations of $\mathrm{St}(\tau)$ which do not contain all the events of T_1 are successful—they contain an event over $\checkmark$.

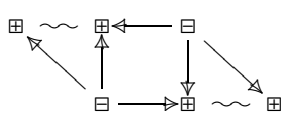
Consequently, (i) it is not the case that (σ_1, M_1) must τ, while (ii) (σ_2, M_2) must τ. To see (i), remark that the relation of causal dependency on T_1 is included in the the total order of the trace α_1 of x_1. Hence $\mathrm{St}(\tau) \otimes \sigma_1$ has a stopping configuration $T_1 \otimes x_1$ which is unsuccessful and thus (σ_1, M_1) fails the must test τ. To show (ii), consider any stopping configuration of $\mathrm{St}(\tau) \otimes \sigma_2$. It comprises $w \otimes x_2$ where w is a stopping configuration of $\mathrm{St}(\tau)$ and $x_2 \in M_2$, a stopping configuration of σ_2. Now $w \not\supseteq T_1$, as by design τ and σ_2 enforce opposing causal dependencies on a pair of synchronisations needed for $T_1 \otimes x_2$ to be defined whenever $x_2 \in M_2$ with $\sigma_2 x_2 = T_1$. Thus w is successful in that it contains an event over $\checkmark$. Hence (σ_2, M_2) must pass τ. This completes the proof. $\square$

Remark. By Example 9.6, the result above would not hold if tests were based solely on pure strategies.

Example 10.5. Let A be the game



Let σ_1 be the stopping strategy given by the identity map $\mathrm{id}_A : A \to A$ together with the +-maximal configurations of A. Let σ_2 be the stopping strategy derived from the event structure



in which there are additional occurrences of Player moves awaiting both moves of Opponent; the map to A is the obvious one and its stopping configurations the +-maximal ones. It can be checked that the two stopping strategies share the same traces of stopping configurations so are 'must' equivalent. $\square$

Infinite stopping configurations play an essential role in the 'must' behaviour of strategies:

Example 10.6. Let the game A consist of an infinite chain of alternating Player-Opponent moves:

$$\boxplus \longrightarrow \boxminus \longrightarrow \boxplus \longrightarrow \boxminus \longrightarrow \cdots \longrightarrow \boxplus \longrightarrow \boxminus \longrightarrow \cdots$$

The strategy σ_1 has as consistent components a copy of A itself and copies of all its initial finite sequences of events ending with an Opponent move; events of different components are inconsistent with each other—we refrain from drawing the wiggly conflicts. The map σ_1 is obvious. (The construction is an instance of the sum of strategies—see Section **??**.)

$$\begin{array}{l}
\boxplus \longrightarrow \boxminus \\
\boxplus \longrightarrow \boxminus \longrightarrow \boxplus \longrightarrow \boxminus \\
\boxplus \longrightarrow \boxminus \longrightarrow \boxplus \longrightarrow \boxminus \longrightarrow \boxplus \longrightarrow \boxminus \\
\qquad\qquad\qquad\qquad \vdots \\
\boxplus \longrightarrow \boxminus \longrightarrow \boxplus \longrightarrow \boxminus \longrightarrow \cdots \longrightarrow \boxplus \longrightarrow \boxminus \\
\qquad\qquad\qquad\qquad \vdots \\
\boxplus \longrightarrow \boxminus \longrightarrow \boxplus \longrightarrow \boxminus \longrightarrow \cdots \longrightarrow \boxplus \longrightarrow \boxminus \longrightarrow \boxplus \longrightarrow \cdots
\end{array}$$

The lowest component is a copy of A. The stopping configurations are all its $+$-maximal configurations, *i.e.* those sets consisting of a whole component or an initial subsequence of a component which ends in a Player move.

The strategy σ_2 is like σ_1 but without the extra infinite component of shape A. Its stopping configurations are all its $+$-maximal configurations, so are necessarily all finite.

The traces of σ_1's and σ_2's *finite* stopping configurations coincide. However the trace of the infinite stopping configuration of σ_1 cannot be a trace of σ_2. Accordingly, from Theorem 10.4, there is a test strategy which σ_2 must pass while σ_1 does not. A distinguishing test $\tau : T \to A^{\perp} \| \boxplus\checkmark$ has T comprising the infinite event structure shown below.

$$\begin{array}{ccccccccccccccc}
\boxminus & \longrightarrow & \boxplus & \longrightarrow & \boxminus & \longrightarrow & \boxplus & \longrightarrow & \boxminus & \longrightarrow & \boxplus & \longrightarrow & \boxminus & \longrightarrow & \boxplus \longrightarrow \cdots \\
\downarrow & & & & \downarrow & & & & \downarrow & & & & \downarrow & & \\
\boxplus\checkmark & & & & \boxplus\checkmark & & & & \boxplus\checkmark & & & & \boxplus\checkmark & & \cdots
\end{array}$$

The strategy σ_2 must pass τ; all $+$-maximal configurations of $\tau \otimes \sigma_2$ are finite and contain a $\checkmark$-event. Whereas the strategy σ_1 can fail to enable a $\checkmark$-event through its extra infinite stopping configuration. $\square$

11 The rigid image of a stopping strategy

In this section we rely on the material of Section 4, in particular that a strategy $\sigma : S \to A$ in a game A has a rigid image

$$\begin{tikzcd} S \arrow[r, two heads, "f"] \arrow[dr, "\sigma"'] & S_0 \arrow[d, "\sigma_1"] \\ & A \end{tikzcd},$$

where f is rigid epi and the rigid image σ_1 is a total map; the map σ_1 automatically inherits the properties required of a strategy. (The construction and key properties of rigid image are unaffected by the extra structure of polarity.) As has been remarked earlier [25, 19], rigid-image strategies have the advantage of forming a category rather than a bicategory. Extended with stopping configurations they can support 'may' and 'must' behaviour.

Definition 11.1. Let (σ, M_S) be a stopping strategy. Let σ_1 be the rigid image of σ with accompanying 2-cell $f : \sigma \Rightarrow \sigma_1$ where f is rigid epi. We define the *rigid image* of (σ, M_S) to be (σ_1, fM_S). A *rigid-image* stopping strategy is one which is its own rigid-image.

As a direct consequence of the last part of Lemma 9.5, we are assured the rigid image of a stopping strategy does not lose any 'may' and 'must' behaviour.

Proposition 11.2. *A stopping strategy is both 'may' and 'must' equivalent to its rigid image.*

As far as 'may' and 'must' behaviour is concerned it is sensible to regard two stopping strategies as equivalent if they share a common rigid image. Rigid-image equivalence transfers to an equivalence between bare strategies: two bare strategies are equivalent if under St we obtain equivalent stopping strategies. W.r.t. 'may' and 'must' behaviour we can choose to work in the category of rigid-image stopping strategies.

What axioms hold of stopping configurations? Such axioms should be preserved by the composition of stopping strategies and rigid image. They should also be complete in the sense that any stopping strategy which satisfied them is rigid-image equivalent to the stopping strategy of some bare strategy. We do not presently know a complete set of axioms for stopping configurations. Candidate axioms on a stopping strategy $\sigma : S \to A$ with stopping configurations M are

Axiom (i) $\forall x \in \mathcal{C}(S) \exists y \in M.\ x \subseteq y$, and

Axiom (ii) $\forall y \in M, x \in \mathcal{C}^\infty(S).\ x \subseteq y$ & x is $+$-maximal in $S \implies x \in M$.

The example below shows why we do not assume all $+$-maximal configurations are stopping. That property is not preserved by taking the rigid image.

Example 11.3. In forming the rigid image $\sigma_1 : S_1 \to A$ of a strategy $\sigma : S \to A$, related by rigid epi 2-cell $f : \sigma \Rightarrow \sigma_1$, it is possible to have an infinite configuration of S_1 which is not in the direct image under f of any configuration of S; in particular it is possible to have a +-maximal configuration of S_1 which is not a direct image of any +-maximal configuration S. For example, let A comprise an infinite chain of Player events. Take S to be the sum of all finite subchains. The rigid image of S is A itself which has +-maximal configuration comprising all the events in the infinite chain, not the image of any configuration of S_1. Thus, in forming the rigid image of a stopping strategy, we cannot assume that all the +-maximal configurations of the rigid image are stopping. □

12 Strategies as concurrent processes

The paper [16] is a closely related study of concurrent strategies from the perspective of concurrent processes, considering how concurrent games and strategies are objects which we can program. Concurrent strategies are shown to support operations yielding an economic yet rich higher-order concurrent process language, which shares features both with process calculi and nondeterministic dataflow. There a slightly weakened definition of bare strategies plays a key role in providing an operational semantics. It would be satisfying to complete this story by providing inequational proof systems for 'may' and 'must' equivalence based on its syntax for strategies, drawing inspiration from the classic work of Hennessy and de Nicola [5].

Process calculi often allow unrestricted recursion. Strategies, as presented here, form a model of linear logic which restricts the copying of parameters needed in recursive definitions. The treatment of unrestricted recursion requires a move to nonlinear strategies over games with symmetry [21, 22]. The recursive definition of bare strategies and strategies can follow classical ideas; 2-cells include the rigid embeddings and inclusions of [23]. Less clear is how to carry out recursive definitions directly on stopping strategies; the 2-cells we have chosen would seem to be too restrictive; and the nature of stopping configurations, that they can be infinite without finite approximations, would push the development into non-continuous operations—nonstandard, if not in itself a bad thing.

A treatment of *winning* concurrent strategies has been presented [31]. Informally a strategy is winning if it must end up in a winning configuration of the game regardless of the behaviour of Opponent. In idea this is very close to *controllability* in [32]. Because the semantics of composition of composition of strategies in [31] is inattentive to the possibilities of deadlock and divergence, a strategy which is obtained as a composition may be deemed winning there and yet possibly deadlock or diverge before reaching a winning configuration [16]. Fortunately the treatment of winning strategies *ibid.* generalises straightforwardly to stopping strategies which keep track of deadlock and divergence, and thus repair this defect. The role of +-maximal configurations in [31] is replaced by that of stopping configurations: a bare or stopping strategy is winning iff all

its stopping configurations image to winning strategies in the game.

Forearmed with concurrent strategies and games with symmetry it would be interesting to revisit old ideas extending testing to other equivalences beyond those of 'may' and 'must' [2]. Certainly one could wish for a better integration of games and strategies with the classical work on concurrency, process algebra and its equivalences included. The medium of concurrent games and strategies based on event structures also provides an inroad into the formalisation and analysis of probabilistic and quantum languages and processes [33, 34, 35].

Acknowledgments

Thanks to Jonathan Hayman, Sacha Huriot, Martin Hyland, Marc Lasson, Conor McBride and Marc de Visme for encouragement and helpful discussions. Thanks to the anonymous referee. Support of Advanced Grant ECSYM (2011-17) of the European Research Council is acknowledged with gratitude. The second author gratefully acknowledges support by ANR project DyVerSe (ANR-19-CE48-0010-01) and Labex MiLyon (ANR-10-LABX-0070) of Université de Lyon, within the program "Investissements d'Avenir" (ANR-11-IDEX-0007), operated by the French National Research Agency (ANR).

References

[1] Abramsky, S., Melliès, P.A.: Concurrent games and full completeness. In: LICS '99, IEEE Computer Society (1999)

[2] Abramsky, S.: Observation equivalence as a testing equivalence. Theor. Comput. Sci. **53** (1987) 225–241

[3] Milner, R.: A Calculus of Communicating Systems. Volume 92 of Lecture Notes in Computer Science. Springer (1980)

[4] Brookes, S., Hoare, C.A.R., Roscoe, A.W.: A theory of communicating sequential processes. J. ACM **31** (1984) 560–599

[5] De Nicola, R., Hennessy, M.: Testing equivalences for processes. Theor. Comput. Sci. **34** (1984) 83–133

[6] Rideau, S., Winskel, G.: Concurrent strategies. In: LICS 2011. (2011)

[7] Castellan, S., Clairambault, P., Rideau, S., Winskel, G.: Games and strategies as event structures. Logical Methods in Computer Science **13**(3) (2017)

[8] Ghica, D.R., Murawski, A.S.: Angelic semantics of fine-grained concurrency. In: FOSSACS'04, LNCS 2987, Springer (2004)

[9] Melliès, P.A., Mimram, S.: Asynchronous games : innocence without alternation. In: CONCUR '07. Volume 4703 of LNCS., Springer (2007)

[10] Faggian, C., Piccolo, M.: Partial orders, event structures and linear strategies. In: TLCA '09. Volume 5608 of LNCS., Springer (2009)

[11] Nielsen, M., Plotkin, G., Winskel, G.: Petri nets, event structures and domains. TCS **13** (1981) 85–108

[12] Winskel, G., Nielsen, M.: Models for concurrency. In Abramsky, S., Gabbay, D., eds.: Semantics and Logics of Computation. OUP (1995)

[13] Conway, J.: On Numbers and Games. Wellesley, MA: A K Peters (2000)

[14] Joyal, A.: Remarques sur la théorie des jeux à deux personnes. Gazette des sciences mathématiques du Québec, 1(4) (1997)

[15] Castellan, S., Clairambault, P., Hayman, J., Winskel, G.: Non-angelic concurrent game semantics. In: Proceedings of FOSSACS 2018. Volume 10803 of Lecture Notes in Computer Science., Springer (2018) 3–19

[16] Castellan, S., Hayman, J., Lasson, M., Winskel, G.: Strategies as concurrent processes. ENTCS **308** (2014)

[17] Harmer, R., McCusker, G.: A fully abstract game semantics for finite nondeterminism. In: 14th Annual IEEE Symposium on Logic in Computer Science, Trento, Italy, July 2-5, 1999. (1999) 422–430

[18] Winskel, G.: Events in computation. PhD thesis, University of Edinburgh (1980)

[19] Castellan, S., Clairambault, P., Winskel, G.: Distributed strategies made easy. In Larsen, K.G., Bodlaender, H.L., Raskin, J., eds.: 42nd International Symposium on Mathematical Foundations of Computer Science, MFCS 2017, August 21-25, 2017 - Aalborg, Denmark. Volume 83 of LIPIcs., Schloss Dagstuhl - Leibniz-Zentrum für Informatik (2017) 81:1–81:13

[20] Winskel, G.: Linearity and nonlinearity in distributed computation. In: Linear Logic in Computer Science. London Mathematical Society Lecture Note Series (CUP) (2004) 151–188

[21] Castellan, S., Clairambault, P., Winskel, G.: Symmetry in concurrent games. In: Joint Meeting of the Twenty-Third EACSL Annual Conference on Computer Science Logic (CSL) and the Twenty-Ninth Annual ACM/IEEE Symposium on Logic in Computer Science (LICS), CSL-LICS '14, Vienna, Austria, July 14 - 18, 2014, ACM (2014)

[22] Castellan, S., Clairambault, P., Winskel, G.: Thin games with symmetry and concurrent hyland-ong games. Logical Methods in Computer Science **15**(1) (2019)

[23] Winskel, G.: Event structure semantics for CCS and related languages. In: ICALP'82. Volume 140 of LNCS., Springer, A full version is available from Winskel's homepage (1982)

[24] Winskel, G.: Event structures. In: Advances in Petri Nets. Volume 255 of LNCS., Springer (1986) 325–392

[25] Winskel, G.: Event Structures, Stable Families and Concurrent Games. http://www.cl.cam.ac.uk/~gw104/ecsym-notes.pdf (2016)

[26] Winskel, G.: Prime algebraicity. Theor. Comput. Sci. **410**(41) (2009) 4160–4168

[27] Winskel, G.: Event structures with symmetry. Electr. Notes Theor. Comput. Sci. 172: 611-652 (2007)

[28] Castellan, S., Clairambault, P., Winskel, G.: 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, IEEE Computer Society (2015) 232–243

[29] Winskel, G.: Strategies as profunctors. In: FOSSACS 2013. Lecture Notes in Computer Science, Springer (2013)

[30] Winskel, G.: Deterministic concurrent strategies. Formal Asp. Comput. **24**(4-6) (2012) 647–660

[31] Clairambault, P., Gutierrez, J., Winskel, G.: The winning ways of concurrent games. In: LICS 2012: 235-244. (2012)

[32] Alur, R., Henzinger, T.A., Kupferman, O.: Alternating-time temporal logic. Journal of the ACM **49**(5) (2002) 672–713

[33] Winskel, G.: Distributed probabilistic and quantum strategies. Electr. Notes Theor. Comput. Sci. 298: 403-425 (2013)

[34] Castellan, S., Clairambault, P., Paquet, H., Winskel, G.: The concurrent game semantics of probabilistic PCF. In Dawar, A., Grädel, E., eds.: Proceedings of the 33rd Annual ACM/IEEE Symposium on Logic in Computer Science, LICS 2018, Oxford, UK, July 09-12, 2018, ACM (2018) 215–224

[35] Clairambault, P., de Visme, M., Winskel, G.: Game semantics for quantum programming. Proc. ACM Program. Lang. **3**(POPL) (2019) 32:1–32:29