Observably Deterministic Concurrent Strategies and Intensional Full Abstraction for Parallel-or*

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

1 **Univ Lyon, CNRS, ENS de Lyon, UCB Lyon 1, LIP**
`first.last@ens-lyon.fr`
2 **Computer Laboratory, University of Cambridge**
`Glynn.Winskel@cl.cam.ac.uk`

Abstract

Although Plotkin's *parallel-or* is inherently deterministic, it has a non-deterministic interpretation in games based on (prime) event structures – in which an event has a unique causal history – because they do not directly support disjunctive causality. *General event structures* can express disjunctive causality and have a more permissive notion of determinism, but do not support hiding. We show that (structures equivalent to) *deterministic* general event structures do support hiding, and construct a new category of games based on them with a deterministic interpretation of $\mathbf{aPCF}_{\mathtt{por}}$, an affine variant of **PCF** extended with parallel-or. We then exploit this deterministic interpretation to give a relaxed notion of determinism (observable determinism) on the plain event structures model. Putting this together with our previously introduced concurrent notions of well-bracketing and innocence, we obtain an intensionally fully abstract model of $\mathbf{aPCF}_{\mathtt{por}}$.

1998 ACM Subject Classification F.3.2 Semantics of Programming Languages

Keywords and phrases Game semantics, parallel-or, concurrent games, event structures, full abstraction

Digital Object Identifier 10.4230/LIPIcs.FSCD.2017.12

1 Introduction

Plotkin's *parallel-or* is, in a sense, the most well-understood of programming language primitives. Recall that it is a primitive $\mathsf{por} : \mathbb{B} \to \mathbb{B} \to \mathbb{B}$ defined by the three equations:

$$\mathsf{por}\ \mathtt{tt}\ \bot = \mathtt{tt} \qquad \mathsf{por}\ \bot\ \mathtt{tt} = \mathtt{tt} \qquad \mathsf{por}\ \mathtt{ff}\ \mathtt{ff} = \mathtt{ff}$$

As Plotkin famously proved [16], it is the primitive to be added to the paradigmatic purely functional higher-order programming language **PCF** in order to get a perfect match (*full abstraction*) with respect to Scott domains. Since Plotkin's result, full abstraction has been the gold standard for semanticists. Indeed, whereas an adequate model is always *sound* to reason about program equivalence; a *fully abstract model* is also *complete*: two programs are observationally equivalent if and only if they have the same denotation. Though since Plotkin's paper fully abstract models of various programming languages have been proposed (often through game semantics, with *e.g.* state [3, 1], control [5, 14], concurrency [11], ...); they are usually quite a bit more complicated than plain Scott domains. Parallel-or is well-understood, in that it has a simple input-output (*extensional*) behaviour, and its presence in **PCF** reduces observational equivalence to input-output behaviour.

* This work was partially supported by French LABEX MILYON (ANR-10-LABX-0070), and by the ERC Advanced Grant ECSYM.

But is it *really* so well-understood? While parallel-or is simple *extensionally*, how is it best understood *intensionally*? What is the operational behaviour of $\mathsf{por}\, M\, N$? Any programmer will sense the subtleties in implementing such a primitive. It immediately spawns two threads, starting evaluating M and N *in parallel*. If both terminate with $\mathtt{ff}$, it terminates with $\mathtt{ff}$ as well after both have finished. But it suffices that *one* of the threads returns $\mathtt{tt}$, for parallel-or to return $\mathtt{tt}$ immediately, discarding the other. If *both* return $\mathtt{tt}$ then one of the threads will be discarded, but which one depends on the scheduler. From an input-output perspective it does not matter which one is selected as they race to produce the same result, but the race nonetheless happens from an operational viewpoint; and a sufficiently intensional semantics will show it. Such behaviour is useful in practice, for instance to speed up deciding the existence of a "good" branch in a search tree by spawning a thread for each branch to explore.

From a game semantics perspective, understanding this combination of racy concurrency and deterministic extensional behaviour has been an unexpected challenge. In earlier work, we dealt with deterministic parallelism using games and strategies based on event structures [17], showing an intensional full abstraction result for a parallel implementation of **PCF** [8]. We also showed how our tools supported shared memory concurrency [6]. Given that, it is no surprise that the racy behaviour mentioned above can be represented with event structures, yielding an adequate model of *e.g.* **PCF** plus parallel-or. But game semantics based on event structures is *causal*: each event comes with a unique causal history. Dependency is *conjunctive*: an event can only occur after *all* its dependencies. This feature is key for the notion of concurrent innocence leading to our definability result [8] and in fact for the very construction of our basic category of games [7]. This has the unfortunate consequence of forcing the strategy for parallel-or to be *non-deterministic*: the race in the operational behaviour of $\mathsf{por}\,\mathtt{tt}\,\mathtt{tt}$ yields a choice between two events competing to return $\mathtt{tt}$, each depending on one argument of por. But interpreting por through non-determinism really is a workaround, as appears through the resulting failure of full abstraction.

Constructing a deterministic intensional model for parallel-or has proved demanding. Plays-based models [11] fail as they inline the non-deterministic choice of the scheduler, and cannot express even pure parallelism deterministically. We saw above that plain event structure games do not work either. Giving a deterministic model for parallel-or involves modifying the latter to allow *disjunctive causality*: a given event may have several distinct causal histories, and an event occurrence does not carry information on its specific causal history. Alternatives to event structures allowing this have existed for a long time: *general event structures* [19]. However, we will see that they do not, in general, support *hiding* – which is required to build a category of strategies. We will show that they do support it modulo some further conditions, making them adequate to model parallel-or deterministically.

Beyond the historical twist of giving a fully abstract games model for a language with parallel-or (as game semantics was originally driven by the full abstraction problem for **PCF** *without* parallel-or [13, 2]), a treatment of disjunctive causality is indispensable in a complete game semantics framework for concurrency. The following example illustrates how mundane and widespread it is: the causal history of a packet arriving from the network is simply its *route*. Clearly, introducing an event for each route is best avoided if possible – especially since all further events depending on it will be hereditarily duplicated as well. Deterministic models for disjunctive causality serve two purposes: they allow for *(1)* a more general, less intensional notion of determinism, and *(2)* a coarser equivalence relation between strategies that abstracts away benign races. These issues, that we address in this paper, have become recurrent obstacles in our research programme; and the historical importance of parallel-or in semantics made it the perfect candidate to test and showcase the solution.

Contributions. Concretely, we build an intensionally fully abstract games model of $\mathbf{aPCF}_{\mathtt{por}}$, an *affine* version of **PCF** with parallel-or. Our tools are designed with the extension in the presence of *symmetry* [8] to full $\mathbf{PCF}_{\mathtt{por}}$ in mind; but presenting them in an affine case allows us to focus on the issues pertaining to parallel-or and disjunctive causality, orthogonal to symmetry. Presenting everything at once in conference format is not reasonable.

Full abstraction for $\mathbf{aPCF}_{\mathtt{por}}$ has two facets: on the one hand, we need to import the conditions of innocence and well-bracketing from [8], which rely on *conjunctive* causality. On the other hand, we need a disjunctive notion of determinism. Consequently our model will involve: *(1)* the standard category **CG** of concurrent games on event structures, and *(2)* the main novelty of our paper, a new category **Disj** supporting disjunctive determinism. Glueing the two together, we can import in **CG** the notion of determinism from **Disj**, thereafter dubbed *observational determinism*, and prove intensional full abstraction.

Related work. *Game semantics* is a branch of denotational semantics. Originally driven by the full abstraction problem for **PCF**, it has grown in the past 25 years into a powerful and versatile methodology to construct compositionally intensional representations of program execution. This lead on the one hand to many full abstraction results (particularly striking in the presence of state as the fully abstract models are then effectively presentable [3, 1]), and on the other hand to applications, ranging from program analysis (model-checking or equivalence checking) to compilation and hardware synthesis [10].

Within game semantics, the present paper is part of a line of work on so-called "truly concurrent" game semantics first pushed by Melliès and colleagues [4, 15], and which advocates the use of causal structures such as event structures and asynchronous transition systems in interactive semantics of programming languages. This line of work has seen a lot of activity in the past five years, prompted in part by a new category of games and strategies on event structures generalizing all prior work, introduced by Rideau and the last author [17]; but also the presheaf-based framework for concurrent games by Hirschowitz and colleagues [12].

Finally, our account of disjunctive causality relies heavily on determinism for *hiding* to work. The third author and Marc de Visme have developed *event structures with disjunctive causality (edc)* [9], supporting both disjunctive causality and hiding in a non-deterministic setting. Links between these two approaches are being explored; at the very least, extra axioms would have to be imposed on edcs in order to mimic the work here.

Outline. In Section 2 we introduce $\mathbf{aPCF}_{\mathtt{por}}$ and give its (non-deterministic) event structure games model **CG**. In Section 3 we introduce the deterministic model **Disj** for parallel-or. Finally in Section 4 we glue the two, generalize the conditions of [8] and prove full abstraction.

2 Causal game semantics for affine PCF with parallel-or

2.1 The language $\mathrm{aPCF}_{\mathtt{por}}$

To alleviate notation, we only take booleans $\mathbb{B}$ as base type. We start with **affine PCF** (**aPCF**). Its types are either $\mathbb{B}$, or $A \multimap B$ for some types A and B. Its terms are those of the affine λ-calculus with a divergence $\bot$ and boolean primitives (constants and conditionals):

$$M, N \quad ::= \quad \lambda x.\, M \mid M\, N \mid x \mid \mathtt{tt} \mid \mathtt{ff} \mid \text{if } M \text{ then } N_1 \text{ else } N_2 \mid \bot$$

We skip the affine typing rules, which are standard – the typing of if M then N_1 else N_2 is additive, *i.e.* N_1 and N_2 may share resources, and the N_i have boolean type. The

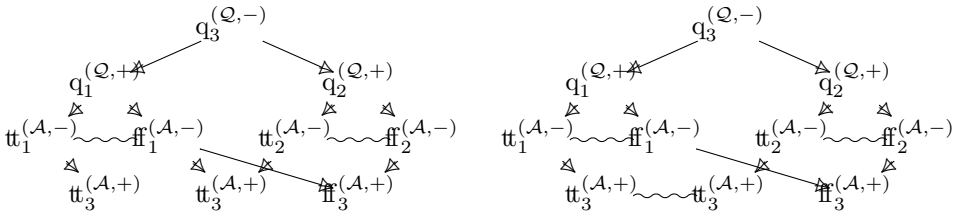


Figure 1 Two strategies on $\mathbb{B}_1 \multimap \mathbb{B}_2 \multimap \mathbb{B}_3$, for parallel-lor and parallel-or

(call-by-name) operational semantics, also standard, yield an evaluation relation $M \Downarrow v$ between closed terms and *values* (booleans or abstractions). We write $M \Downarrow$ when there is some v such that $M \Downarrow v$, and $M \Uparrow$ otherwise. Two terms $\Gamma \vdash M, N : A$ are **observationally equivalent** ($M \simeq_{\text{obs}} N$) iff for all contexts $\mathcal{C}[-]$ such that $\mathcal{C}[M], \mathcal{C}[N]$ are closed terms of type $\mathbb{B}$, $\mathcal{C}[M] \Downarrow \Leftrightarrow \mathcal{C}[N] \Downarrow$. Recall that an interpretation $[\![-]\!]$ in some model $\mathcal{M}$ is **fully abstract** if for all M, N, $M \simeq_{\text{obs}} N$ iff $[\![M]\!] = [\![N]\!]$ – it is **intensionally fully abstract** if $\mathcal{M}$ quotiented by the semantic equivalent of $\simeq_{\text{obs}}$ is fully abstract.

In [16], Plotkin proved that Scott domains are fully abstract not for **PCF**, but for its extension with the so-called **parallel-or** operation:

$$\frac{M \Downarrow \mathtt{tt}}{\mathsf{por}\ M\ N \Downarrow \mathtt{tt}} \qquad \frac{N \Downarrow \mathtt{tt}}{\mathsf{por}\ M\ N \Downarrow \mathtt{tt}} \qquad \frac{M \Downarrow \mathtt{ff} \qquad N \Downarrow \mathtt{ff}}{\mathsf{por}\ M\ N \Downarrow \mathtt{ff}}$$

The combinator por is *not sequential*: it is not the case that it evaluates one of its arguments first. In particular, $\mathsf{por}\ \mathtt{tt}\ \bot \Downarrow \mathtt{tt}$ and $\mathsf{por}\ \bot\ \mathtt{tt} \Downarrow \mathtt{tt}$.

2.2 Strategies as event structures

As usual in game semantics, computation is a dialogue between the program and its environment. The *moves*, or *events*, are either due to the program (Player, $+$) or its environment (Opponent, $-$). They are either variable calls (Questions, $\mathcal{Q}$) or returns (Answers, $\mathcal{A}$). Unlike traditional game semantics, in our line of work such dialogues are *partially ordered.*

Figure 1 presents two concurrent strategies, both playing on (the game for) $\mathbb{B}_1 \multimap \mathbb{B}_2 \multimap \mathbb{B}_3$, where subscripts are for disambiguation. The diagrams are read from top to bottom. The first event in both strategies is an Opponent question on $\mathbb{B}_3$, initiating the computation. Then Player (the program) starts evaluating in parallel its two arguments (q_1 and q_2). These may return $\mathtt{tt}$ or $\mathtt{ff}$, the wiggly line indicating that they cannot reply *both*. Depending on these, Player may eventually answer q_3. In both diagrams, $\mathtt{ff}_3$ requires *both* arguments to evaluate to $\mathtt{ff}$; however they differ as to the events that trigger an answer $\mathtt{tt}_3$. These diagrams will be made more formal later, but we invite the readers to examine them and convince themselves that the first diagram represents a parallel implementation of the *left or* (diverging if its first argument diverges), whereas the second diagram represents *parallel or*.

Event structures. Such diagrams are formalized as *event structures*. We use here event structures with binary conflict, whereas those of [17, 7] have a more general set of *consistent sets*. Binary conflict is sufficient for our purposes, and preserved by all operations we need.

▶ **Definition 1.** A (prime) **event structure** (with binary conflict, **es** for short) is $(E, \leq_E, \#_E)$ where E is a set of events, $\leq_E$ is a partial order on E called **causality** and $\#_E$ is an irreflexive symmetric binary relation called *conflict*, such that:

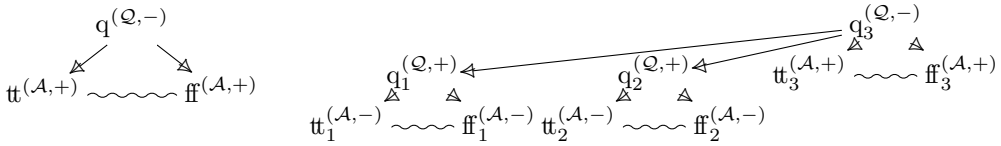


Figure 2 Representations of the arenas $[\![\mathbb{B}]\!]$ and $[\![\mathbb{B}_1 \multimap \mathbb{B}_2 \multimap \mathbb{B}_3]\!]$.

- $\forall e \in E,\ [e] = \{e' \in E \mid e' \leq_E e\}$ is finite,
- $\forall e \# e',\ \forall e' \leq_E e'',\ e \# e''$.

We will often omit the indices in $\leq_E$, $\#_E$ if they are obvious from the context.

In the second axiom, we say that the conflict (e, e'') is **inherited** from (e, e'). If a conflict (e, e') is not inherited (meaning dependencies of e and e' are pairwise **compatible**, *i.e.* non-conflicting), we say that it is a **minimal conflict** and denote it by $e \frown\!\!\!\frown e'$. The *states* of an event structure E, called **configurations**, are the finite sets $x \subseteq E$ that are both consistent (events are pairwise compatible) and **down-closed** (*i.e.* for all $e \in x$, for all $e' \leq e$, then $e' \in x$) – the set of configurations on E is written $\mathscr{C}(E)$, and is partially ordered by inclusion. Configurations with a maximal element are called **prime configurations**, they are those of the form $[e]$ for $e \in E$. We will also use the notation $[e) = [e] \setminus \{e\}$. Between configurations, the **covering relation** $x \lessdot y$ means that y is obtained from x by adding exactly one event: y is an **atomic extension** of x. We might also write $x \overset{e}{\lessdot}$ to mean that $e \notin x$ and $x \cup \{e\} \in \mathscr{C}(E)$. Finally, when drawing event structures, we will not represent the full partial order $\leq$ but the **immediate causality** generating it, defined as $e \rightarrowtriangle e'$ whenever $e < e'$ and for any $e \leq e'' \leq e'$, either $e = e''$ or $e'' = e'$. The diagrams of Figure 1 represent event structures; only displaying immediate causality $\rightarrowtriangle$ and minimal conflict $\frown\!\!\!\frown$.

Arenas. Besides causality and conflict, events in Figure 1 carry *labels* $(\mathrm{q}, \mathrm{tt}, \mathrm{ff}, \dots)$: formally, those will come from the *game*, or the *arena*. Arenas are the semantic representatives of types. They are certain event structures with polarities and Questions/Answers labeling.

▶ **Definition 2.** An **arena** is a triple $(A, \mathsf{pol}_A, \lambda_A)$ where A is an event structure, $\mathsf{pol}_A : A \to \{-, +\}$ and $\lambda_A : A \to \{\mathcal{Q}, \mathcal{A}\}$ are labelings for polarity and Questions/Answers, such that:

- The order $\leq_A$ is *forest-shaped*: for $a_1, a_2 \leq a \in A$, either $a_1 \leq a_2$ or $a_2 \leq a_1$.
- The relation $\rightarrowtriangle_A$ is *alternating*: if $a_1 \rightarrowtriangle_A a_2$, then $\mathsf{pol}_A(a_1) \neq \mathsf{pol}(a_2)$.
- If $\lambda_A(a_2) = \mathcal{A}$, then there is $a_1 \rightarrowtriangle a_2$, and $\lambda_A(a_1) = \mathcal{Q}$,
- A is *race-free*: if $a_1 \frown\!\!\!\frown a_2$, then $\mathsf{pol}_A(a_1) = \mathsf{pol}_A(a_2)$.

An arena A is **negative** if all its minimal events have negative polarity.

Conflict aside, our arenas resemble those of [13]: the justification relation traditionally denoted by $\vdash_A$ is simply $\rightarrowtriangle_A$. Basic arenas include the empty arena 1, and the arena $[\![\mathbb{B}]\!]$ displayed in Figure 2, often written $\mathbb{B}$ by abuse of notation, for booleans.

We mention here some constructions on arenas. The **dual** $A^\perp$ of A is obtained by taking $\mathsf{pol}_{A^\perp} = -\mathsf{pol}_A$, and leaving the rest unchanged. The **simple parallel composition** $A \parallel B$ is obtained as having events the tagged disjoint union $\{1\} \times A \cup \{2\} \times B$, and all components inherited. The **product** $A \,\&\, B$ of negative A and B is obtained as $A \parallel B$, with all events of A in conflict with events of B. As types will be denoted by *negative* arenas, we need a negative arena to interpret $\multimap$. This is done by setting A to depend on (negative) minimal events of B. If B has at most one minimal event, this is easy:

▶ **Definition 3.** Let A, B be negative arenas. Assume that B is **well-opened**, *i.e.* $\min(B)$ has at most one event. If $B = 1$, then $A \multimap B = 1$. Otherwise, $\min(B) = \{b_0\}$. We define $A \multimap B$ as $A^\perp \parallel B$, with the additional causal dependency $(2, b_0) \leq (1, a)$ for all $a \in A$.

Defining $A \multimap B$ for well-opened B is sufficient to interpret $\mathbf{aPCF}_{\mathtt{por}}$ types, but would be insufficient in the presence of a tensor type; with *e.g.* $\mathbb{B} \multimap (\mathbb{B} \otimes \mathbb{B})$. The complication here is due to a *disjunctive causality* at the level of types: the left $\mathbb{B}$ is caused by either of the occurrences of $\mathbb{B}$ on the right. As for parallel-or, the inability of event structures to express that can be worked around by introducing two conflicting copies of the left $\mathbb{B}$, one for each causal justification – this is done in [6], and is reminiscent of the standard arena construction of [13]. This works well for some purposes, but the informed reader may see why this threatens definability for a concurrent language with tensor: indeed a counter-strategy can then behave differently depending on the cause of an occurrence of the left $\mathbb{B}$.

We avoid the issue here, and only build $\multimap$ for $\mathbf{aPCF}_{\mathtt{por}}$ types. Interestingly the issue vanishes in Section 3: **Disj** supports disjunctive causality in both arenas and strategies.

Prestrategies on arenas. Strategies are certain event structures labeled by arenas. More formally, the labeling function is required to be a *map of event structures*:

▶ **Definition 4.** A **prestrategy** on arena A is a function on events $\sigma : S \to A$ *s.t.* σ is a **map of event structures**: it preserves configurations ($\forall x \in \mathscr{C}(S), \sigma\, x \in \mathscr{C}(A)$) and is locally injective ($\forall s_1, s_2 \in x \in \mathscr{C}(S), \sigma\, s_1 = \sigma\, s_2 \implies s_1 = s_2$).

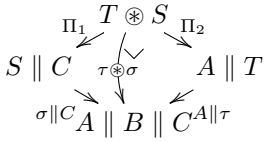
Event structures and their maps form a category $\mathcal{E}$. Figure 1 displays such prestrategies $\sigma : S \to A$ – the event structure drawn is S, and events are annotated by their image in A through σ. *Strategies*, introduced in the next section, will be subject to further conditions.

2.3 An interpretation of $\mathbf{aPCF}_{\mathtt{por}}$ as strategies

Following the methodology of denotational semantics, giving an interpretation of $\mathbf{aPCF}_{\mathtt{por}}$ consists in constructing a category out of certain prestrategies, with enough structure to interpret $\mathbf{aPCF}_{\mathtt{por}}$. For lack of space we can only sketch part of the construction, a more detailed account of the construction, originally from [17], can be found in [7].

A **prestrategy from** A **to** B is a prestrategy $\sigma : S \to A^\perp \parallel B$. Given also $\tau : T \to B^\perp \parallel C$, we wish to *compose* them. As usual in game semantics, this involves *(1)* parallel interaction where the strategies freely communicate, and *(2)* hiding. We focus on *(1)* first.

Interaction. The interaction of $\sigma : S \to A^\perp \parallel B$ and $\tau : T \to B^\perp \parallel C$ is an event structure $T \circledast S$, labeled by $A \parallel B \parallel C$ via $\tau \circledast \sigma : T \circledast S \to A \parallel B \parallel C$. The omitted definition [7] of $T \circledast S$ follows the lines of the third author's event structure semantics for parallel composition in CCS [20]. It is uniquely determined (up to iso) as a *pullback* [7] in $\mathcal{E}$:



It is the "smallest" labeled event structure accommodating the constraints of σ and τ.

Hiding. Composition should yield a prestrategy from A to C. Accordingly events of $T \circledast S$ that map to A or C are available to the outside world, and called *visible*. On the other hand, those mapping to B are private synchronization events, dubbed *invisible*; that we wish to *hide*. The proposition below formalizes that event structures are expressive enough for that.

▶ **Proposition 5.** Event structures **support hiding**: for E an event structure and V any subset of events, then there exists an event structure $E \downarrow V$ having V as events, and as configurations exactly those $x \cap V$ for $x \in \mathscr{C}(E)$.

The components of $E \downarrow V$ (causality, conflict) are simply inherited from E.

For σ and τ as above, we first form $T \odot S$ as $T \circledast S \downarrow V$ with V the visible events. The **composition** $\tau \odot \sigma : T \odot S \to A^\perp \parr C$ is simply the restriction of $\tau \circledast \sigma$; and is a prestrategy.

A category. Composition is associative up to isomorphism of prestrategies, where $\sigma_1 : S_1 \to A$ and $\sigma_2 : S_2 \to A$ are isomorphic (written $\sigma_1 \cong \sigma_2$) if there is an iso between S_1 and S_2 in $\mathcal{E}$ making the obvious triangle commute. Finally, for any arena A there is a **copycat prestrategy** $\text{ⓒ}_A : \mathbb{C}_A \to A^\perp \parr A$, which has events and immediate conflict those of $A^\perp \parr A$, and with causality that of $A^\perp \parr A$ where additionally each positive event is set to depend on its negative counterpart on the other side. Details can be found in [17, 7].

Copycat is idempotent, but is not an identity in general. Rideau and Winskel [17] characterise the prestrategies for which it acts as an identity as the *strategies*:

▶ **Theorem 6.** *For $\sigma : S \to A^\perp \parr B$, $\text{ⓒ}_B \odot \sigma \odot \text{ⓒ}_A \cong \sigma$ iff σ is a **strategy**: it is **receptive** (for $x \in \mathscr{C}(S)$, if $\sigma\, x \xrightarrow{a^-} \subset$ there is a unique $x \xrightarrow{s^-} \subset$ s.t. $\sigma\, s = a$) and **courteous** (if $s_1 \rightarrowtriangle_S s_2$ and $\mathsf{pol}(s_1) = +$ or $\mathsf{pol}(s_2) = -$, then $\sigma\, s_1 \rightarrowtriangle_{A^\perp \parr B} \sigma\, s_2$).*

We mention (see [7]) that **CG** is a *compact closed category*, *i.e.* fit to interpret the *linear λ-calculus*. As $\mathbf{aPCF}_{\mathtt{por}}$ is *affine*, we work in a derived category of *negative strategies*.

Negative arenas and strategies. Negative arenas were defined earlier. We also have:

▶ **Definition 7.** A strategy $\sigma : S \to A^\perp \parr B$ is **negative** iff $\forall s \in \min(S)$, $\mathsf{pol}(s) = -$.

There is a subcategory $\mathbf{CG}_-$ of **CG** with *negative arenas* as objects, and *negative strategies* as morphisms. The negativity assumption on strategies ensures that 1 is terminal (the only negative strategy on $A^\perp \parr 1$, for A negative, is the empty strategy e_A), which allows us to interpret weakening. The tensor product is defined as $A \otimes B = A \parr B$ on arenas, and as the obvious relabeling $\sigma_1 \otimes \sigma_2 : S_1 \parr S_2 \to (A_1 \parr A_2)^\perp \parr B_1 \parr B_2$ on strategies, for $\sigma_1 : S_1 \to A_1^\perp \parr B_1$ and $\sigma_2 : S_2 \to A_2^\perp \parr B_2$. Besides, $\mathbf{CG}_-$ has **products**: for $\sigma : S \to A^\perp \parr B$ and $\tau : A^\perp \parr C$ their pairing $\langle \sigma, \tau \rangle : S \,\&\, T \to A^\perp \parr (B \,\&\, C)$ is the obvious labeling, and we have projections $\varpi_A : \mathbb{C}_A \to (A \,\&\, B)^\perp \parr A$ and $\varpi_B : \mathbb{C}_B \to (A \,\&\, B)^\perp \parr B$. For well-opened C, the negative strategies on $A^\perp \parr B^\perp \parr C$ and $A^\perp \parr B \multimap C$ are exactly the same. Because $A \multimap B$ is only defined for well-opened B, $\mathbf{CG}_-$ is not monoidal closed; but well-opened arenas form an *exponential ideal*:

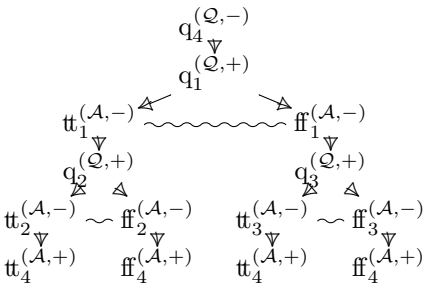


Figure 3 if : $\mathbb{B}_1^\perp \parr (\mathbb{B}_2 \,\&\, \mathbb{B}_3)^\perp \parr \mathbb{B}_4$

▶ **Proposition 8.** $\mathbf{CG}_-$ is a symmetric monoidal category with products, with 1 terminal. For A and well-opened B, $A \multimap B$ is an exponential object; and is well-opened.

Following standard lines, we interpret $\mathbf{aPCF}_{\mathtt{por}}$ in $\mathbf{CG}_-$: contexts $\Gamma = x_1 : A_1, \ldots, x_n : A_n$ are interpreted as $[\![\Gamma]\!] = [\![A_1]\!] \parr \cdots \parr [\![A_n]\!]$. Types are interpreted as well-opened arenas, each type construction by its arena counterpart. Terms $\Gamma \vdash M : A$ are interpreted as negative

strategies $[\![M]\!] \in \mathbf{CG}_{\_}([\![\Gamma]\!], [\![A]\!])$. Divergence $\bot$ is interpreted as the strategy (also written $\bot$) with no positive moves, closed by receptivity. Constants $\mathtt{tt}, \mathtt{ff}$ are the obvious strategies on $\mathbb{B}$. For $\Gamma \vdash M : \mathbb{B}$, $\Delta \vdash N_1, N_2 : \mathbb{B}$, $[\![\text{if } M \text{ then } N_1 \text{else } N_2]\!] = \mathbf{if} \odot ([\![M]\!] \otimes \langle [\![N_1]\!], [\![N_2]\!] \rangle)$, where **if** is given in Figure 3. Finally, parallel-or is the strategy on the right of Figure 1.

The interpretation $[\![-]\!]$ is adequate: for all $\vdash M : \mathbb{B}$, $M \Downarrow$ iff $[\![M]\!] \neq \bot$. However, we will be better equipped to prove that in Section 4. For now, the desired induction invariant for soundness $M \Downarrow v \Rightarrow [\![M]\!] = [\![v]\!]$ fails, as the model keeps track of too much intensional information: $[\![\mathsf{por}\ \mathtt{tt}\ \mathtt{tt}]\!]$ has two conflicting $\mathtt{tt}^+$ events, and is therefore not isomorphic to $[\![\mathtt{tt}]\!]$.

2.4 Disjunctive causality and observational determinism

We have seen that $\mathbf{CG}_{\_}$ is an adequate model of $\mathbf{aPCF}_{\mathtt{por}}$. However, it is not *intensionally fully abstract*: strategies distinguish more than terms of $\mathbf{aPCF}_{\mathtt{por}}$. In fact, as in [6] one can interpret *e.g.* linearly used boolean references in $\mathbf{CG}_{\_}$; those can obviously distinguish more than the input-output behaviour of terms. We can remove these by requiring conditions of visibility, well-bracketing and innocence as in [8]; and we will do so in Section 4.2.

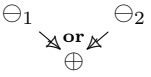
One condition of [8] is however inadequate: *determinism*. Indeed, the strategy of Figure 1 is non-deterministic, and no deterministic strategy can implement parallel-or (indeed, all deterministic strategies on first-order types implement sequential functions [8]). Formalizing a notion of *observational determinism*, accepting the strategy for parallel-or but rejecting genuinely non-deterministic strategies, proved very challenging.

Disjunctive causality. As we pointed out in the introduction, the non-determinism of parallel-or comes from the incapacity of our version of event structures, sometimes referred to as *prime* event structures for disambiguation, to express *disjunctive causality*. So the reader may wonder: why do we bother with prime event structures at all? After all, there are alternatives. *General event structures* [19] allow events to be enabled in several distinct ways. We give their equivalent presentation [21] in terms of *configuration families*:

▶ **Definition 9.** A **configuration family (cf)** is a pair $(|\mathcal{A}|, \mathcal{A})$ (often just written $\mathcal{A}$) where $|\mathcal{A}|$ is a set of events, and $\mathcal{A}$ is a set of **configurations**, which are finite subsets of $|\mathcal{A}|$, such that $\emptyset \in \mathcal{A}$, and satisfying the two further axioms:

- *Completeness.* For $x, y \in \mathcal{A}$, if $x \uparrow y$ (they are **compatible**, *i.e.* there exists $z \in \mathcal{A}$ such that $x \subseteq z$ and $y \subseteq z$), then $x \cup y \in \mathcal{A}$.
- *Coincidence-freeness.* If $x \in \mathcal{A}$, for all distinct $e_1, e_2 \in x$ there exists $y \subseteq x$ such that $y \in \mathcal{A}$ and $e_1 \in y \Leftrightarrow e_2 \notin y$.

For every (prime) event structure A, $\mathscr{C}(A)$ is a configuration family. However configuration families are more general. In a configuration family, the same event can occur for different reasons. For instance, on the set of events $\{\ominus_1, \ominus_2, \oplus\}$, Figure 4 represents a configuration family where the event $\oplus$ can occur either because of $\ominus_1$ or $\ominus_2$, represented symbolically as in the diagram on the right – importantly, an occurrence of $\oplus$ carries no data on its effective cause. With the same symbolic representation, the causally disjunctive nature of parallel-or is made explicit in Figure 5. It seems clear that this is a more accurate description of parallel-or. In particular, it is *deterministic*, in a sense to be made formal in the next section.



Are configuration families sufficiently expressive as a basis to construct a category of strategies, as we did with prime event structures in the previous section? Recall that for that we used two properties of event structures: that the corresponding category $\mathcal{E}$ has *pullbacks*

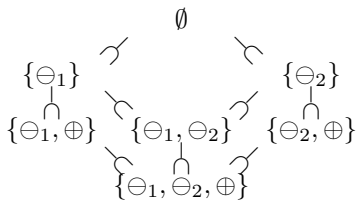


Figure 4 Disj. caus. in configuration families

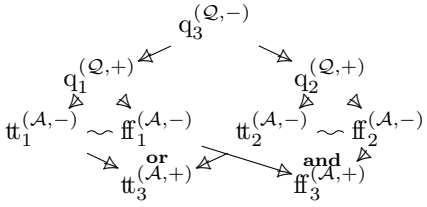
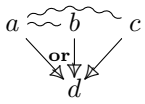


Figure 5 por as a disj. strategy

(for interactions), and then that event structures support *hiding*. We will show very soon that the first criterion holds. However, we run into an issue for the second:

▶ **Proposition 10.** Configuration families do *not* support hiding. There is a cf $\mathcal{A}$ and $V \subseteq |A|$ such that the set $\{x \cap V \mid x \in \mathcal{A}\}$ is not a configuration family.

Proof. Set $|\mathcal{A}| = \{a, b, c, d\}$, and configurations those specified by: (d is caused either by a, or by b and c together). With $V = \{b, c, d\}$, the obtained hidden set fails completeness: it contains $\{b\}, \{d\}, \{b, c, d\}$ but not $\{b, d\}$. ◀



It is part for this reason that prime event structures are used when building categories of games and strategies – as indeed, hiding is crucial. Removing either coincidence-freeness or completeness loses the correspondence with general event structures [21], and leads to various pathologies further down the road – for instance we lose the key lemma:

▶ **Lemma 11.** *Let $\mathcal{A}$ a set of configurations on $|\mathcal{A}|$ satisfying completeness. Then, it is coincidence-free iff for any $x, y \in \mathcal{A}$* s.t. *$x \subseteq y$, there is a covering chain in $\mathcal{A}$: $x \multimap \subset \ldots \multimap \subset y$.*

3 Disjunctive deterministic games

In this section we introduce the main contribution of the paper, a category **Disj** of disjunctive deterministic strategies supporting the interpretation of parallel-or. This relies on a notion of *deterministic* configuration families, and the observation that those *do* support hiding.

3.1 Deterministic configuration families

First we adjoin cfs with *polarities*, and define determinism (for Player).

▶ **Definition 12.** A cf **with polarities (cfp)** (*resp.* **partial polarities (cfpp)**) is a cf $\mathcal{A}$ with $\mathsf{pol}_{\mathcal{A}} : |A| \to \{-, +\}$ (*resp.* $\mathsf{pol}_{\mathcal{A}} : |A| \to \{-, 0, +\}$).

A cfpp $\mathcal{A}$ is **deterministic** (dcfp/dcfpp) iff for all $x \subseteq^p_{\mathcal{A}} y$ and $x \subseteq z$, we have $y \cup z \in \mathcal{A}$ – we write $x \subseteq^p_{\mathcal{A}} y$ (resp. $x \subseteq^-_{\mathcal{A}} y, x \subseteq^+_{\mathcal{A}} y$) to mean that $x \subseteq y$ and $\mathsf{pol}(y \setminus x) \subseteq \{0, +\}$ (resp. $\mathsf{pol}_{\mathcal{A}}(y \setminus x) \subseteq \{-\}$, $\mathsf{pol}_{\mathcal{A}}(y \setminus x) \subseteq \{+\}$).

This means that only Opponent makes irreversible choices regarding the evolution of the play. We show that dcfpps support hiding of neutral events. In the sequel, given a dcfpp $\mathcal{A}$, V always denotes the set of non-neutral events in $|A|$. We write $\mathcal{A}_{\downarrow}$ for the candidate cf on events V with configurations those of the form $x_{\downarrow} = x \cap V$ for $x \in \mathcal{A}$. The following key lemma is proved by successive applications of determinism and completeness.

▶ **Lemma 13.** *For $x \in \mathcal{A}_{\downarrow}$, $y \in \mathcal{A}$ such that $x \subseteq y_{\downarrow}$, for any $x' \in \mathcal{A}$ a witness for x* (i.e. *$x'_{\downarrow} = x$), there is $y \subseteq y' \in \mathcal{A}$ such that $x' \subseteq y'$.*

From Lemma 13 follows that for $x, y \in \mathcal{A}$, if $x_\downarrow \uparrow y_\downarrow$ in $\mathcal{A}_\downarrow$, then $x \uparrow y$ in $\mathcal{A}$. Using that observation together with Lemma 13 and determinism, it is straightforward to prove:

▶ **Proposition 14.** For $\mathcal{A}$ a dcfpp, $\mathcal{A}_\downarrow$ is a dcfp.

3.2 Deterministic disjunctive strategies and composition

A **pregame** is simply a cfp – for $\mathcal{A}, \mathcal{B}$ cfps, $\mathcal{A}^\perp$ has the same events and configurations as $\mathcal{A}$ but inverted polarity. We will also write $\mathcal{B}^0$ for the cfpp with the same configurations as $\mathcal{B}$ but polarity set as globally 0. The cf $\mathcal{A} \parallel \mathcal{B}$ has events the tagged disjoint union, and configurations $x_A \parallel x_B$ for $x_A \in \mathcal{A}$, $x_B \in \mathcal{B}$; $\mathcal{A} \,\&\, \mathcal{B}$ has the same events as $\mathcal{A} \parallel \mathcal{B}$, but only those configurations with one side empty.

Unlike in Definition 4, *disjunctive prestrategies* are simply substructures of the (pre)games.

▶ **Definition 15.** A **prestrategy** on $\mathcal{A}$ is a dcfp σ on $|\mathcal{A}|$, such that $\sigma \subseteq \mathcal{A}$ – written $\sigma : \mathcal{A}$.

▶ **Example 16.** On events (with polarities) $\{\mathrm{q}^-, \mathrm{tt}^+, \mathrm{ff}^+\}$, $\mathscr{C}(\mathbb{B})$ is a cfp – that by abuse of notation we still write $\mathbb{B}$. Then, Figure 5 denotes a prestrategy on $\mathbb{B}^\perp \parallel \mathbb{B}^\perp \parallel \mathbb{B}$.

Though disjunctive prestrategies are not primarily defined as maps, it will be helpful in composing them that they *can* be. Given two cfs $\mathcal{A}$ and $\mathcal{B}$, a **map from $\mathcal{A}$ to $\mathcal{B}$** is a function on events $f : |\mathcal{A}| \to |\mathcal{B}|$ which preserves configurations and is locally injective. Configuration families and their maps forms a category **Fam**. Note that if A is an event structure, $\mathscr{C}(A)$ is a configuration family on $|A|$. The definition of maps of cfs is compatible with that of maps of event structures, making $\mathscr{C}(-) : \mathcal{E} \to \mathbf{Fam}$ a full and faithful functor. Finally, a prestrategy $\sigma : \mathcal{A}$ on $\mathcal{A}$ can be regarded as an identity-on-events **Fam**-morphism $\sigma \to \mathcal{A}$.

Like $\mathcal{E}$, **Fam** has pullbacks. The **interaction** of σ on $\mathcal{A}^\perp \parallel \mathcal{B}$ and τ on $\mathcal{B}^\perp \parallel \mathcal{C}$ is the pullback of identity-on-events maps $\sigma \parallel \mathcal{C} \to \mathcal{A} \parallel \mathcal{B} \parallel \mathcal{C}$ and $\mathcal{A} \parallel \tau \to \mathcal{A} \parallel \mathcal{B} \parallel \mathcal{C}$.

▶ **Proposition 17.** The pullback above is (up to iso) the cf $\tau \circledast \sigma$ with events $|\mathcal{A}| \parallel |\mathcal{B}| \parallel |\mathcal{C}|$ and configurations those $x \in (\sigma \parallel \mathcal{C}) \cap (\mathcal{A} \parallel \tau)$ that are **secured**: for distinct $p_1, p_2 \in x$ there is $x \supseteq y \in (\sigma \parallel \mathcal{C}) \cap (\mathcal{A} \parallel \tau)$ *s.t.* $p_1 \in y \Leftrightarrow p_2 \notin y$.

Equivalently, $x \in (\sigma \parallel \mathcal{C}) \cap (\mathcal{A} \parallel \tau)$ is secured iff it has a *covering chain*, *i.e.* a sequence: $\emptyset = x_0 \text{–}\!\subset x_1 \text{–}\!\subset \ldots \text{–}\!\subset x_n = x$ where for all $0 \leq i \leq n$, $x_i \in (\sigma \parallel \mathcal{C}) \cap (\mathcal{A} \parallel \tau)$.

We regard $\tau \circledast \sigma$ as a cfpp by setting as polarities those of $\mathcal{A}^\perp \parallel \mathcal{B}^0 \parallel \mathcal{C}$. To use Proposition 14 and finish defining composition, $\tau \circledast \sigma$ needs to be deterministic; a sufficient condition for that is that σ and τ are *receptive*. A prestrategy σ on $\mathcal{A}$ is **receptive** iff for all $x \in \sigma$, if $x \cup \{a^-\} \in \mathcal{A}$ then $x \cup \{a^-\} \in \sigma$.

▶ **Proposition 18.** If σ and τ are receptive prestrategies, then $\tau \circledast \sigma$ is a *deterministic* cfpp. Then, $\tau \odot \sigma = (\tau \circledast \sigma)_\downarrow$ is a receptive prestrategy on $\mathcal{A}^\perp \parallel \mathcal{C}$.

Hence, composition is well-defined on receptive prestrategies – it is also associative. To get a category, we now prove the copycat strategy to be an identity.

3.3 Copycat and the compact closed category Disj

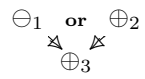
We cannot replicate in pregames the definition of copycat sketched in Section 2.3. However, as observed in [18, 7], if A is an arena, the *configurations* of $\mathbb{CC}_A$ are those configurations of $A^\perp \parallel A$, necessarily of the form $x_l \parallel x_r$, such that every positive event of x_r (*w.r.t.* A) is already in x_l, and every positive event in x_l (*w.r.t.* $A^\perp$) is already in x_r. In other words, $x_l \supseteq_A^- x_l \cap x_r \subseteq_A^+ x_r$, written $x_r \sqsubseteq_A x_l$ in [18, 7] and and referred to as the "Scott order".

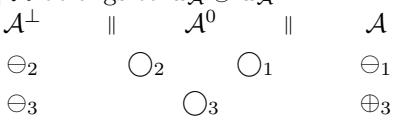
Accordingly, on a pregame $\mathcal{A}$, given $x, y \in \mathcal{A}$ we write $x \sqsubseteq_{\mathcal{A}} y$ iff $x \cap y \in \mathcal{A}$ and the relation above holds. The candidate prestrategy **copycat** $\mathfrak{c}_A$ comprises all $x \parallel y \in \mathcal{A}^\perp \parallel \mathcal{A}$ *s.t.* $y \sqsubseteq_{\mathcal{A}} x$. Indeed $\mathfrak{c}_{\mathcal{A}}$ is a configuration family and is receptive; but prestrategies must be *deterministic*. It turns out that as in [22], $\mathfrak{c}_{\mathcal{A}}$ is only deterministic when $\mathcal{A}$ is *race-free*:

▶ **Proposition 19.** Let $\mathcal{A}$ be a pregame. Then, $\mathfrak{c}_{\mathcal{A}}$ is a prestrategy iff $\mathcal{A}$ is **race-free**: for all $x, y, z \in \mathcal{A}$ such that $x \subseteq^+_{\mathcal{A}} y$ and $x \subseteq^-_{\mathcal{A}} z$, we have $y \uparrow_{\mathcal{A}} z$.

We aim to reproduce Theorem 6 in this new setting, and characterise the prestrategies left invariant under composition with copycat. However, there is a new subtlety here: for arbitrary $\mathcal{A}$, $\mathfrak{c}_{\mathcal{A}}$ might not even be idempotent!

▶ **Example 20.** Consider the cfp on events $A = \{\ominus_1, \oplus_2, \oplus_3\}$ given on the right. This is a race-free pregame, so by Proposition 19, $\mathfrak{c}_{\mathcal{A}} \odot \mathfrak{c}_{\mathcal{A}}$ is a prestrategy on $\mathcal{A}^\perp \parallel \mathcal{A}$. However, it is *distinct from* $\mathfrak{c}_{\mathcal{A}}$. Indeed, the following configuration of $\mathcal{A}^\perp \parallel \mathcal{A}^0 \parallel \mathcal{A}$ belongs to $\mathfrak{c}_{\mathcal{A}} \circledast \mathfrak{c}_{\mathcal{A}}$:





After hiding, there is a change in the causal history of $\oplus_3$, not authorized by $\mathfrak{c}_{\mathcal{A}}$ but authorised by $\mathcal{A}$: it is caused by $\ominus_2$ on the left, but $\ominus_1$ on the right. This issue comes from the fact that in $\mathcal{A}$, the same event can be caused either by a positive or a negative move. Such behaviour will be banned in games:

▶ **Proposition 21.** If $\mathcal{A}$ is a race-free pregame, the following are equivalent: *(1)* $\mathfrak{c}_{\mathcal{A}}$ is idempotent, *(2)* $\sqsubseteq_{\mathcal{A}}$ is a partial order, *(3)* $\mathcal{A}$ is **co-race-free**: for all $x, y, z \in \mathcal{A}$ with $x \supseteq^- y, x \supseteq^+ z$, we have $y \cap z \in \mathcal{A}$.

A **game** will be a race-free, co-race-free pregame. Copycat on any game is an idempotent prestrategy – *strategies* are those prestrategies that compose well with copycat. We prove:

▶ **Theorem 22.** *Let σ be a receptive prestrategy on $\mathcal{A}^\perp \parallel \mathcal{B}$. Then, $\mathfrak{c}_{\mathcal{B}} \odot \sigma \odot \mathfrak{c}_{\mathcal{A}} = \sigma$ iff σ is **courteous**: for any $x \multimap\subset x \cup \{a_1^+\} \multimap\subset x \cup \{a_1^+, a_2^+\}$ in σ, if $x \cup \{a_2^+\} \in \mathcal{A}^\perp \parallel \mathcal{B}$, then $x \cup \{a_2^+\} \in \sigma$ as well. In this case, σ is a **strategy**, written: $\sigma : \mathcal{A}^\perp \parallel \mathcal{B}$.*

It follows that there is a category **Disj** with games as objects, and strategies $\sigma : \mathcal{A}^\perp \parallel \mathcal{B}$ as morphisms from $\mathcal{A}$ to $\mathcal{B}$. It is fairly easy to show that just as **CG**, this category is compact closed. But unlike **CG**, **Disj** supports a deterministic interpretation of parallel-or: indeed the set of configurations of $\mathbb{B}^\perp \parallel \mathbb{B}^\perp \parallel \mathbb{B}$ denoted by the diagram of Figure 5 is a strategy.

3.4 An SMCC and deterministic interpretation of $\mathbf{aPCF}_{\mathtt{por}}$

As for **CG**, **Disj** lacks structure to interpret $\mathbf{aPCF}_{\mathtt{por}}$: the family of bottom strategies $e_{\mathcal{A}} : \mathcal{A}^\perp \parallel 1$ (where again 1 is the empty game) fails naturality. As before, we hence restrict **Disj** to a subcategory of *negative* games and strategies.

▶ **Definition 23.** A cfp $\mathcal{A}$ is **negative** if any non-empty $x \in \mathcal{A}$ includes a negative event.

Copycat on negative $\mathcal{A}$ is negative and negative strategies are stable under composition; so there is a subcategory $\mathbf{Disj}_-$ of **Disj** with negative games as objects and negative strategies as morphisms, inheriting a symmetric monoidal structure from **Disj**. Moreover 1 is terminal, and $\mathbf{Disj}_-$ has products given by $\mathcal{A} \,\&\, \mathcal{B}$.

To prove the monoidal closure, as in **CG**, we have to deal with the fact that for $\mathcal{A}$ and $\mathcal{B}$ negative, $\mathcal{A}^\perp \parallel \mathcal{B}$ is not necessarily negative, so we define:

▶ **Definition 24.** Let $\mathcal{A}$ and $\mathcal{B}$ be two negative games. The game $\mathcal{A} \multimap \mathcal{B}$ has the same events (and polarity) as $\mathcal{A}^\perp \parallel \mathcal{B}$, but non-empty configurations those $x_A \parallel x_B \in \mathcal{A}^\perp \parallel \mathcal{B}$ such that x_B is non-empty (and hence includes a negative event).

Recall from Definition 3 that the arrow arena $A \multimap B$ was only defined for B *well-opened* (*i.e.* with at most one minimal event). This was to avoid constructing arenas for types like $\mathbb{B} \multimap (\mathbb{B} \otimes \mathbb{B})$ (invalid for $\mathbf{aPCF}_{\mathtt{por}}$, but valid in an extension with a tensor type), where the left hand side $\mathbb{B}$ can be opened because of either of the right hand side occurrences of $\mathbb{B}$. In **Disj**, when constructing $\mathcal{A} \multimap \mathcal{B}$ there is no well-openness condition on $\mathcal{B}$: exploiting disjunctive causality we can express that $\mathcal{A}$ is opened only after *some* event of $\mathcal{B}$ – it does not matter which one. The negative strategies in $\mathcal{A}^\perp \parallel \mathcal{B}^\perp \parallel \mathcal{C}$ and $\mathcal{A}^\perp \parallel \mathcal{B} \multimap \mathcal{C}$ are the same, from which (using the compact closed structure of **Disj**) it follows that $\mathbf{Disj}_!$ is monoidal closed, instead of only having an exponential ideal. Like $\mathbf{CG}_!$, it supports an adequate interpretation of $\mathbf{aPCF}_{\mathtt{por}}$. Unlike $\mathbf{CG}_!$, its strategies are *deterministic*.

4 Glueing $\mathrm{CG}_!$ and $\mathrm{Disj}_!$, and full abstraction

By moving from $\mathbf{CG}_!$ to $\mathbf{Disj}_!$, we gain determinism. However, $\mathbf{Disj}_!$ is *not* intensionally fully abstract – there are some strategies that distinguish syntactically undistinguishable terms of $\mathbf{aPCF}_{\mathtt{por}}$. For instance, one has a strategy on $(\mathbb{B} \multimap \mathbb{B} \multimap \mathbb{B}) \multimap \mathbb{B}$ that calls its argument, feeds it $\mathtt{tt}$ as first argument, and $\mathtt{tt}$ as second argument only if the first argument has been evaluated; it then copies the final result to toplevel. Doing so, it distinguishes the observationally equivalent $\lambda xy.\,\mathrm{if}\ x\ \mathrm{then}\ (\mathrm{if}\ y\ \mathrm{then}\ \mathtt{tt}\ \mathrm{else}\ \bot)\ \mathrm{else}\ \bot$ and $\lambda xy.\,\mathrm{if}\ y\ \mathrm{then}\ (\mathrm{if}\ x\ \mathrm{then}\ \mathtt{tt}\ \mathrm{else}\ \bot)\ \mathrm{else}\ \bot$.

Getting rid of such strategies is the responsibility of the concept of *innocence* in Hyland-Ong games [13]. But the *P-views* of innocent strategies carry precisely the *causal* information that we have lost when moving from $\mathbf{CG}_!$ to $\mathbf{Disj}_!$! This causal information was crucial in [8] to give concurrent versions of well-bracketing and innocence, so as to capture the behaviour of parallel **PCF** strategies. So, the deterministic account of $\mathbf{aPCF}_{\mathtt{por}}$ is not enough; the causal (where innocence and well-bracketing are defined) and deterministic models should be related, to establish in what sense the causal model is already "observably" deterministic.

4.1 A glued games model

For any arena A (Definition 2), $\mathscr{C}(A)$ is a game – it is race-free and co-race-free. Moreover, arena constructions and game constructions match: for any A, B, $\mathscr{C}(A^\perp) = \mathscr{C}(A)^\perp$, $\mathscr{C}(A \parallel B) = \mathscr{C}(A) \parallel \mathscr{C}(B)$; for negative A, B we have $\mathscr{C}(A \,\&\, B) = \mathscr{C}(A) \,\&\, \mathscr{C}(B)$, and if B is well-opened, $\mathscr{C}(A \multimap B) = \mathscr{C}(A) \multimap \mathscr{C}(B)$. Therefore, we will just take the objects of our glued games model to be arenas. The strategies, though, will be strategies in *both categories*.

▶ **Definition 25.** Let A be an arena. A strategy $\sigma : S \to A$ is **observationally deterministic (odet)** if *(1)* the **display** of σ, $\mathcal{O}(\sigma) = \{\sigma\, x \mid x \in \mathscr{C}(S)\}$ is a disjunctive deterministic strategy on $\mathscr{C}(A)$ (in **Disj**). It follows then that σ is also a **Fam**-morphism $\sigma : \mathscr{C}(S) \to \mathcal{O}(\sigma)$. We also ask that it has *(2)* the **configuration extension property**: for any $x \in \mathscr{C}(S)$, if $\sigma\, x \subseteq y \in \mathcal{O}(\sigma)$, then there is $x \subseteq x' \in \mathscr{C}(S)$ such that $\sigma\, x' = y$.

The configuration extension property ensures that two causal realizations of the same displayed configuration have bisimilar futures. The display operation yields a coarser equivalence on strategies: odet strategies $\sigma, \tau : A$, are **display-equivalent**, written $\sigma \approx \tau$, if $\mathcal{O}(\sigma) = \mathcal{O}(\tau)$. This coarser equivalence is key to ensure soundness of our interpretation, as the interpretation in $\mathbf{CG}_!$ of $\mathsf{por}\ \mathtt{tt}\ \mathtt{tt}$ and $\mathtt{tt}$ are *not* isomorphic, but only display-equivalent.

▶ **Theorem 26.** *There is a compact closed category* **Odet** *with arenas as objects and odet strategies up to* $\approx$ *as morphisms; with a symmetric monoidal subcategory* $\mathbf{Odet}_-$ *of negative arenas and negative strategies with products and* 1 *terminal. It has well-opened arenas as an exponential ideal. Finally,* $\mathbf{Odet}_-$ *supports an adequate interpretation of* $\mathbf{aPCF}_{por}$.

Proof. For odet σ and τ, $\mathcal{O}(\tau \odot \sigma) = \mathcal{O}(\tau) \odot \mathcal{O}(\sigma)$: $\subseteq$ is direct, $\supseteq$ exploits the extension property for σ and τ. So, $\mathcal{O}(\tau \odot \sigma)$ is a deterministic disjunctive strategy, and it has the extension property – hence $\tau \odot \sigma$ is odet. In general, $\mathcal{O}$ links the structure of **CG** to that of **Disj**, forming **Odet**. We sketch adequacy.

Soundness. For all $M \Downarrow v$, $[\![M]\!] = [\![v]\!]$ (by induction on the evaluation derivations).

Adequacy. We prove by induction on the size of M that if $M \Uparrow$, then $[\![M]\!]$ is bottom. The size is an adequate measure because as $\mathbf{aPCF}_{\mathtt{por}}$ is affine, substitution is non-copying and each induction step is on a strictly smaller term. ◀

As a result, interpretations in $\mathbf{CG}_-$ and **Disj** are also adequate. We have accommodated the causal representation of programs permitted by **CG** and the determinism of **Disj**. Now, it remains to import the causal conditions on strategies of [8], and prove full abstraction.

4.2 Conditions

We now recall *innocence* and *well-bracketing*, introduced in [8]. Formulated for *deterministic strategies* (in the sense of $\mathbf{CG}_-$), those would not suffice to prove full abstraction for a genuinely non-deterministic language. For that, further conditions are needed to ensure the locality of conflicts – those will appear in the first author's forthcoming PhD thesis. However, we will show in Lemma 31 that for $\mathbf{aPCF}_{\mathtt{por}}$, distinguishable strategies can be distinguished via *deterministic* (in the sense of $\mathbf{CG}_-$) contexts, so these simpler conditions will suffice.

All our conditions rely crucially on the notion of *grounded causal chain*.

▶ **Definition 27.** A **grounded causal chain** (**gcc**) in S is a sequence $\rho = \rho_0 \rightarrowtriangle \ldots \rightarrowtriangle \rho_n$ where ρ_0 is minimal. The set of gccs of S is written $\mathrm{gcc}(S)$.

By courtesy, gccs of strategies on arenas are always alternating. They give a notion of *thread* of a concurrent strategy. Two gccs $\rho, \rho' \in \mathrm{gcc}(S)$ are **forking** when $\rho \cup \rho'$ is consistent in S, and there is $k \in \mathbb{N}$ s.t. $\rho_i = \rho'_i$ for $i < k$ and $\{\rho_i\}_{i \geq k}$ and $\{\rho'_j\}_{j \geq k}$ are non-empty and disjoint. They are **negatively forking** when $\mathsf{pol}(\rho_{k+1}) = \mathsf{pol}(\rho'_{k+1}) = -$, **positively forking** otherwise. Two forking gccs ρ, ρ' are **joined** at $s \in S$ when $\rho_\omega \rightarrowtriangle s$ and $\rho'_\omega \rightarrowtriangle s$. ($\rho_\omega$ refers to the last event of ρ) as shown in this picture:

$$\rho_0 \rightarrowtriangle \ldots \rightarrowtriangle \rho_k \begin{array}{l} \nearrow \rho_{k+1} \rightarrowtriangle \ldots \rightarrowtriangle \rho_\omega \searrow \\ \searrow \rho'_{k+1} \rightarrowtriangle \ldots \rightarrowtriangle \rho'_\omega \nearrow \end{array} s$$

Innocence. *Innocence* enforces independence between threads forked by Opponent.

▶ **Definition 28.** Let $\sigma : S \to A$ be a strategy on a arena A. It is **innocent** when it is:
Visible: The image $\sigma\,\rho$ of a gcc (regarding $\rho \in \mathrm{gcc}(S)$ as a set) is a configuration of A.
Preinnocent: Two negatively forking gccs are never joined.

Preinnocence is a *locality* condition, forcing Player to deal independently with threads forked by Opponent – in the sequential case, it coincides with Hyland-Ong innocence. In the concurrent case though, it is not by itself stable under composition; that is where visibility comes in. Together they are preserved under composition and under all the operations on strategies involved in the interpretation of $\mathbf{aPCF}_{\mathtt{por}}$ in $\mathbf{CG}_-$ (and hence $\mathbf{Odet}_-$).

Well-bracketing. Traditionally, *well-bracketing* in game semantics rules out strategies that behave like a control operator, manipulating the call stack. It exploits the question/answer labelling, reminiscent of the function calls/returns. In arenas from $\mathbf{aPCF}_{\mathtt{por}}$ types, answers to the same question are always conflicting, so every consistent set has at most one answer to any question. A consistent set X is **complete** when it has exactly one answer to any question. A question is **pending** in a set X if it has no answer in X and maximally so in X.

▶ **Definition 29.** A strategy $\sigma : S \to A$ is **well-bracketed** when *(1)* if $a \in S$ answers $q \in S$, then q is pending in $[a]$ and *(2)* for $\rho, \rho' \in \mathrm{gcc}(S)$ forking at $\rho_k = \rho'_k$, and joined, the segments $\rho_{>k}$ and $\rho_{>k'}$ must be complete.

The affinity condition of [8] comes for free here, thanks to the conflict in $\mathbb{B}$. Well-bracketing is proved stable under composition and other operations in [8]. A $\mathbf{aPCF}_{\mathtt{por}}$**-strategy** is a negative, innocent, well-bracketed and odet strategy.

▶ **Theorem 30.** *There is a symmetric monoidal category* **PorStrat** *of negative arenas and* $\mathbf{aPCF}_{por}$*-strategies, with products, a terminal object and an exponential ideal comprising* $\mathbb{B}$*. Moreover, the interpretation of* $\mathbf{aPCF}_{por}$ *in* $\mathbf{Odet}_{\_}$ *factors through* $\mathbf{PorStrat} \subseteq \mathbf{Odet}_{\_}$*.*

From now on, all strategies will be assumed to be $\mathbf{aPCF}_{\mathtt{por}}$-strategies.

4.3 Intensional full abstraction

Two strategies $\sigma, \tau : A$ are **observationally equivalent** ($\sigma \simeq_{\mathrm{obs}} \tau$) when for all strategies $\alpha : A^{\perp} \parallel \mathbb{B}$, $\alpha \odot \sigma \approx \alpha \odot \tau$. In fact, there is no need to quantify over *all* strategies. A **path-strategy** is a strategy $\sigma : S \to A$ such that there exists a configuration $x \in \mathscr{C}(S)$ that contains all the positive moves of S. Any distinguishing strategy yields by restriction a distinguishing path-strategy – this would fail without affinity, as the restriction would fail uniformity [8], and enforcing uniformity would make it non-deterministic.

▶ **Lemma 31.** *Two distinguishable strategies can be distinguished by a path-strategy.*

▶ **Lemma 32.** *Every path-strategy can be defined by a term of* **aPCF** *up to* $\simeq_{\mathrm{obs}}$*.*

Proof. As in [8] since path-strategies are deterministic – the only difference is the necessity to distribute the context among the subterms to ensure affine typing. ◀

A corollary is that in an affine setting, por adds no distinguishing power: two $\mathbf{aPCF}_{\mathtt{por}}$ terms are observationally equivalent if and only if they cannot be distinguished by a context from affine **PCF** without por. Intensional full abstraction follows by standard techniques:

▶ **Theorem 33.** *The interpretation of* $\mathbf{aPCF}_{por}$ *into* **PorStrat** *is intensionally fully abstract: it preserves and reflects observational equivalence (see Section 2.1).*

5 Conclusion

This leaves many avenues for further investigation. On the semantic front, we plan among other applications to exploit **Odet** to model non-interfering concurrent languages. On the foundational front we need to understand better how the present approach relates to the treatment of disjunctive causality in edcs [9].

Acknowledgments. We are grateful to Tamás Kispéter for interesting discussions about strategies as general event structures.

References

1 Samson Abramsky, Kohei Honda, and Guy McCusker. A fully abstract game semantics for general references. In *LICS, Indianapolis, Indiana, USA, June 21-24, 1998*, pages 334–344. IEEE Computer Society, 1998.

2 Samson Abramsky, Radha Jagadeesan, and Pasquale Malacaria. Full abstraction for PCF. *Inf. Comput.*, 163(2):409–470, 2000.

3 Samson Abramsky and Guy McCusker. Linearity, sharing and state: a fully abstract game semantics for idealized algol with active expressions. *ENTCS*, 3:2–14, 1996.

4 Samson Abramsky and Paul-André Melliès. Concurrent games and full completeness. In *LICS, Trento, Italy, July 2-5, 1999*, pages 431–442. IEEE Computer Society, 1999.

5 Robert Cartwright, Pierre-Louis Curien, and Matthias Felleisen. Fully abstract semantics for observably sequential languages. *Inf. Comput.*, 111(2):297–401, 1994.

6 Simon Castellan and Pierre Clairambault. Causality vs. interleavings in concurrent game semantics. In *CONCUR 2016, August 23-26, 2016, Québec City, Canada*, pages 32:1–32:14.

7 Simon Castellan, Pierre Clairambault, Silvain Rideau, and Glynn Winskel. Games and strategies as event structures. 2016. Submitted, https://arxiv.org/abs/1604.04390.

8 Simon Castellan, Pierre Clairambault, and Glynn Winskel. The parallel intensionally fully abstract games model of PCF. In *LICS, Kyoto, Japan, July 6-10, 2015*, pages 232–243. IEEE Computer Society, 2015.

9 Marc de Visme and Glynn Winskel. Strategies with parallel causes. To appear in CSL 2017.

10 Dan R. Ghica. Applications of game semantics: From program analysis to hardware synthesis. In *LICS, 11-14 August 2009, Los Angeles, CA, USA*, pages 17–26. IEEE Computer Society, 2009.

11 Dan R. Ghica and Andrzej S. Murawski. Angelic semantics of fine-grained concurrency. *Ann. Pure Appl. Logic*, 151(2-3):89–114, 2008.

12 Tom Hirschowitz. Full abstraction for fair testing in CCS (expanded version). *Logical Methods in Computer Science*, 10(4), 2014.

13 J. M. E. Hyland and C.-H. Luke Ong. On full abstraction for PCF: I, II, and III. *Inf. Comput.*, 163(2):285–408, 2000.

14 James Laird. Full abstraction for functional languages with control. In *Proceedings, 12th Annual IEEE Symposium on Logic in Computer Science, Warsaw, Poland, June 29 - July 2, 1997*, pages 58–67. IEEE Computer Society, 1997.

15 Paul-André Melliès and Samuel Mimram. Asynchronous games: Innocence without alternation. In *CONCUR, Lisbon, Portugal, September 3-8, 2007*, volume 4703 of *Lecture Notes in Computer Science*, pages 395–411. Springer, 2007.

16 Gordon D. Plotkin. LCF considered as a programming language. *Theor. Comput. Sci.*, 5(3):223–255, 1977.

17 Silvain Rideau and Glynn Winskel. Concurrent strategies. In *LICS, June 21-24, 2011, Toronto, Ontario, Canada*, pages 409–418, 2011.

18 Glynn Winskel. Strategies as profunctors. In *FOSSACS, Held as Part of ETAPS 2013, Rome, Italy, March 16-24, 2013*, pages 418–433.

19 Glynn Winskel. *Events in computation*. PhD Thesis, Edinburgh University, 1980.

20 Glynn Winskel. Event structure semantics for CCS and related languages. In *ICALP, Aarhus, Denmark, July 12-16, 1982, Proceedings*, pages 561–576, 1982.

21 Glynn Winskel. Event structures. In *Petri Nets: Central Models and Their Properties, Advances in Petri Nets 1986, Part II, Proceedings of an Advanced Course, Bad Honnef, 8.-19. September 1986*, pages 325–392, 1986.

22 Glynn Winskel. Deterministic concurrent strategies. *Formal Asp. Comput.*, 24(4-6):647–660, 2012.