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.CVIT.2016.12

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.

42nd Conference on Very Important Topics (CVIT 2016).
Editors: John Q. Open and Joan R. Acces; Article No. 12; pp. 12:1–12:31
Leibniz International Proceedings in Informatics
LIPICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

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

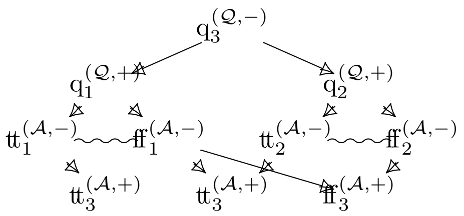


Figure 1 Two strategies on $\mathbb{B}_1 \multimap \mathbb{B}_2 \multimap \mathbb{B}_3$, for parallel-lor and parallel-or

(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, $\mathsf{por}\, \mathtt{tt} \perp \Downarrow \mathtt{tt}$ and $\mathsf{por} \perp \mathtt{tt} \Downarrow \mathtt{tt}$.

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:

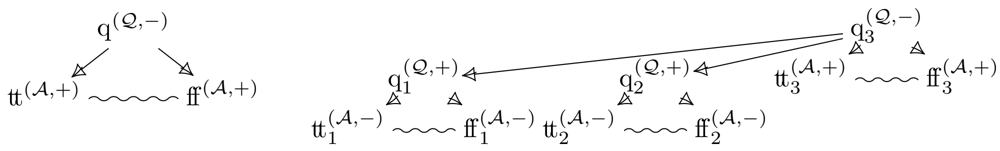


Figure 2 Representations of the arenas $[\![\mathbb{B}]\!]$ and $[\![\mathbb{B}_1 \multimap \mathbb{B}_2 \multimap \mathbb{B}_3]\!]$.

- $\forall e \in E,\ [e] = \{e' \in E \mid e' \leq_E e\}$ 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 $e \sim\!\sim e'$. 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** $x \longrightarrow\!\!\!\!\subset y$ means that y is obtained from x by adding exactly one event: y is an **atomic extension** of x. We might also write $x \overset{e}{\longrightarrow\!\!\!\!\subset}$ 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 $\sim\!\sim$.

Arenas. Besides causality and conflict, events in Figure 1 carry *labels* ($\mathrm{q}, \mathtt{tt}, \mathtt{ff}, \ldots$): 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 $a_1 \sim\!\sim a_2$, 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 ($\forall s_1, s_2 \in x \in \mathscr{C}(S), \sigma\, s_1 = \sigma\, s_2 \implies s_1 = s_2$).

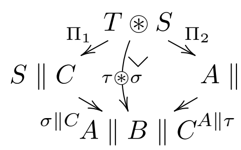
Event structures and their maps form a category $\mathcal{E}$. Figure 1 displays such prestrategies $\sigma : S \to A$ – the event structure drawn is S, and events are annotated by their image in A through σ. *Strategies*, introduced in the next section, will be subject to further conditions.

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** $\tau \odot \sigma : T \odot S \to A^\perp \parallel C$ 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** $\text{æ}_A : \mathbb{C}_A \to A^\perp \parallel A$, which has events and immediate conflict those of $A^\perp \parallel A$, and with causality that of $A^\perp \parallel A$ 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 $\sigma : S \to A^\perp \parallel B$, $\text{æ}_B \odot \sigma \odot \text{æ}_A \cong \sigma$ iff σ is a **strategy**: it is **receptive** (for $x \in \mathscr{C}(S)$, if $\sigma\, x \overset{a^-}{\longrightarrow\!\subset}$ there is a unique $x \overset{s^-}{\longrightarrow\!\subset}$ 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 $\sigma\, s_1 \rightarrowtriangle_{A^\perp \parallel B} \sigma\, s_2$).*

We mention (see [7]) that **CG** is a *compact closed category*, *i.e.* fit to interpret the *linear λ-calculus*. As $\mathbf{aPCF}_{\texttt{por}}$ 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 $\sigma : S \to A^\perp \parallel B$ 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 $A^\perp \parallel 1$, for A negative, is the empty strategy e_A), which allows us to interpret weakening. The tensor product is defined as $A \otimes B = A \parallel B$ on arenas, and as the obvious relabeling $\sigma_1 \otimes \sigma_2 : S_1 \parallel S_2 \to (A_1 \parallel A_2)^\perp \parallel B_1 \parallel B_2$ on strategies, for $\sigma_1 : S_1 \to A_1^\perp \parallel B_1$ and $\sigma_2 : S_2 \to A_2^\perp \parallel B_2$. Besides, $\mathbf{CG}_-$ has **products**: for $\sigma : S \to A^\perp \parallel B$ and $\tau : A^\perp \parallel C$ their pairing $\langle \sigma, \tau \rangle : S \,\&\, T \to A^\perp \parallel (B \,\&\, C)$ is the obvious labeling, and we have projections $\varpi_A : \mathbb{C}_A \to (A \,\&\, B)^\perp \parallel A$ and $\varpi_B : \mathbb{C}_B \to (A \,\&\, B)^\perp \parallel B$. For well-opened C, the negative strategies on $A^\perp \parallel B^\perp \parallel C$ and $A^\perp \parallel B \multimap C$ 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*:

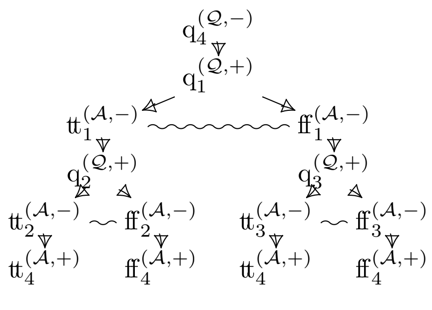


Figure 3 $\mathbf{if} : \mathbb{B}_1^\perp \parallel (\mathbb{B}_2 \,\&\, \mathbb{B}_3)^\perp \parallel \mathbb{B}_4$

▶ **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 $\mathbf{aPCF}_{\texttt{por}}$ in $\mathbf{CG}_-$: contexts $\Gamma = x_1 : A_1, \ldots, x_n : A_n$ are interpreted as $[\![\Gamma]\!] = [\![A_1]\!] \parallel \cdots \parallel [\![A_n]\!]$. 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.

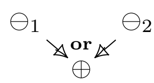
One condition of [8] is however inadequate: *determinism.* Indeed, the strategy of Figure 1 is non-deterministic, and no deterministic strategy can implement parallel-or (indeed, all deterministic strategies on first-order types implement sequential functions [8]). Formalizing a notion of *observational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

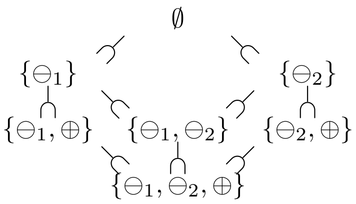


Figure 4 Disj. caus. in configuration families

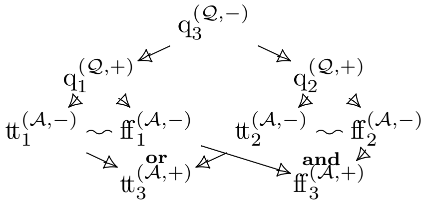
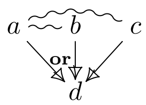


Figure 5 por as a disj. strategy

(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 $\mathcal{A}$: $x \longrightarrow\!\!\!\!\subset \ldots \longrightarrow\!\!\!\!\subset y$.*

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) $\{\mathrm{q}^-, \mathtt{tt}^+, \mathtt{ff}^+\}$, $\mathscr{C}(\mathbb{B})$ 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: $\emptyset = x_0 \longrightarrow\!\!\!\!\subset x_1 \longrightarrow\!\!\!\!\subset \dots \longrightarrow\!\!\!\!\subset x_n = x$ 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

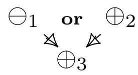
We cannot replicate in pregames the definition of copycat sketched in Section 2.3. However, as observed in [18, 7], if A is an arena, the *configurations* of $\mathbb{C}_A$ are those configurations of $A^\perp \parallel A$, necessarily of the form $x_l \parallel x_r$, such that every positive event of 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** $\text{\ae}_{\mathcal{A}}$ comprises all $x \parallel y \in \mathcal{A}^{\perp} \parallel \mathcal{A}$ *s.t.* $y \sqsubseteq_{\mathcal{A}} x$. Indeed $\text{\ae}_{\mathcal{A}}$ is a configuration family and is receptive; but prestrategies must be *deterministic*. It turns out that as in [22], $\text{\ae}_{\mathcal{A}}$ is only deterministic when $\mathcal{A}$ is *race-free*:

▶ **Proposition 19.** Let $\mathcal{A}$ be a pregame. Then, $\text{\ae}_{\mathcal{A}}$ 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 $\mathcal{A}$, $\text{\ae}_{\mathcal{A}}$ 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, $\text{\ae}_{\mathcal{A}} \odot \text{\ae}_{\mathcal{A}}$ is a prestrategy on $\mathcal{A}^{\perp} \parallel \mathcal{A}$. However, it is *distinct from* $\text{\ae}_{\mathcal{A}}$. Indeed, the following configuration of $\mathcal{A}^{\perp} \parallel \mathcal{A}^{0} \parallel \mathcal{A}$ belongs to $\text{\ae}_{\mathcal{A}} \circledast \text{\ae}_{\mathcal{A}}$:



$\mathcal{A}^{\perp}$	$\parallel$	$\mathcal{A}^{0}$		$\parallel$	$\mathcal{A}$
$\ominus_2$	$\bigcirc_2$		$\bigcirc_1$		$\ominus_1$
$\ominus_3$		$\bigcirc_3$			$\oplus_3$

After hiding, there is a change in the causal history of $\oplus_3$, not authorized by $\text{\ae}_{\mathcal{A}}$ 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)* $\text{\ae}_{\mathcal{A}}$ 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, $\text{\ae}_{\mathcal{B}} \odot \sigma \odot \text{\ae}_{\mathcal{A}} = \sigma$ iff σ is* ***courteous****: for any $x {-\!\!\subset} x \cup \{a_1^{+}\} {-\!\!\subset} x \cup \{a_1^{+}, a_2^{+}\}$ 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 $\mathbf{aPCF}_{\texttt{por}}$

As for **CG**, **Disj** lacks structure to interpret $\mathbf{aPCF}_{\texttt{por}}$: 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 **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 **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 $\mathbf{aPCF}_{\texttt{por}}$, 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 $\mathbf{aPCF}_{\texttt{por}}$. Unlike $\mathbf{CG}_\_$, its strategies are *deterministic*.

4 Glueing $\mathbf{CG}_\_$ and $\mathbf{Disj}_\_$, 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 $\mathbf{aPCF}_{\texttt{por}}$. 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 $\mathbf{aPCF}_{\texttt{por}}$ 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 $\mathscr{C}(A \mathbin{\&} B) = \mathscr{C}(A) \mathbin{\&} \mathscr{C}(B)$, 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* $\mathbf{aPCF}_{\mathit{por}}$*-strategies, with products, a terminal object and an exponential ideal comprising* $\mathbb{B}$*. Moreover, the interpretation of* $\mathbf{aPCF}_{\mathit{por}}$ *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* $\mathbf{aPCF}_{\mathit{por}}$ *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. *CoRR*, abs/1607.03747, 2016.

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.

For proofs about the construction of **CG**, we refer the reader to [7]. From the compact closed structure of **CG**, it is easy to derive the required structure of $\mathbf{CG}_{-}$ and the interpretation of $\mathbf{aPCF}_{\texttt{por}}$. This technical appendix will focus on the construction of **Disj** and **Odet**.

A Construction of Disj

A.1 Configuration families

▶ **Lemma 34.** *Let $\mathcal{F}$ be a set of configurations on underlying set of events F, with $\emptyset \in \mathcal{F}$ and satisfying the completeness axiom. Then, the following are equivalent.*

(i) *$\mathcal{F}$ satisfies coincidence-freeness.*

(ii) *For any $x, y \in \mathcal{F}$ such that $x \subseteq y$, there is a covering chain in $\mathcal{F}$:*

$$x \mathrel{-\!\!\!\subset} \ldots \mathrel{-\!\!\!\subset} y$$

(iii) *Any $x \in \mathcal{F}$ has a covering chain.*

(iv) *Any $x \in \mathcal{F}$ has a last element: for any $x \in \mathcal{F}$, there exists $y \in \mathcal{F}$ such that $y \mathrel{-\!\!\!\subset} x$.*

Proof. *(i)* ⇒ *(ii)*. If $x = y$ or $x \mathrel{-\!\!\!\subset} y$, there is nothing to prove. Otherwise, there are $e_1, e_2 \in y$ such that $e_1, e_2 \notin x$. By coincidence-freeness, there is $z \in \mathcal{F}$ such that $z \subseteq y$ and $e_1 \in z \Leftrightarrow e_2 \notin z$. Say *e.g.* $e_1 \in z$ and $e_2 \notin z$. But $z \uparrow x$ since both are included in y, so $x \cup z \in \mathcal{F}$. By construction we have $x \subseteq x \cup z \subseteq y$, with at least $e_2 \in y \setminus (x \cup z)$ and $e_1 \in (x \cup z) \setminus x$. So by induction hypothesis these two inclusions have covering chains, which can be composed to yield a covering chain for $x \subseteq y$.

(ii) ⇒ *(iii)*. Trivial.

(iii) ⇒ *(iv)*. Trivial.

(iv) ⇒ *(i)*. Immediate by induction. ◀

▶ **Proposition 35.** Let $f : \mathcal{A} \to \mathcal{C}$ and $g : \mathcal{B} \to \mathcal{C}$ be two maps in **Fam**. There is a pullback diagram:

$$\begin{array}{ccc} \mathcal{A} \wedge \mathcal{B} & \xrightarrow{\pi_2} & \mathcal{B} \\ {\scriptstyle \pi_1}\downarrow & \lrcorner & \downarrow{\scriptstyle g} \\ \mathcal{A} & \xrightarrow{f} & \mathcal{C} \end{array}$$

where $\mathcal{A} \wedge \mathcal{B}$ is a configuration family on underlying set of events $A \times B$, with configurations the bijections $\varphi : x \simeq y$ between $x \in \mathcal{A}$ and $y \in \mathcal{B}$ such that $(a, b) \in \varphi \implies f\,a = g\,b$, such that for all distinct $(a, b), (a', b') \in \varphi$, there exists $\varphi' \subseteq \varphi$ such that $\varphi' : x' \simeq y'$ with $x' \in \mathcal{A}$ and $y' \in \mathcal{B}$, and $(a, b) \in \varphi' \Leftrightarrow (a', b') \notin \varphi'$.

The maps π_1 and π_2 are then the set-theoretic projections.

Proof. We prove that this is a valid configuration family. Clearly it includes the empty set. Assume that φ, φ' are two compatible configurations of $\mathcal{A} \wedge \mathcal{B}$, *i.e.* there exists φ'' such that $\varphi \subseteq \varphi'', \varphi' \subseteq \varphi''$ (write x, x', x'' the corresponding left projections and y, y', y'' the corresponding left configurations). Then $x \cup x' \in \mathcal{A}$ and $y \cup y' \in \mathcal{B}$ since $\mathcal{A}, \mathcal{B}$ are configuration families. Then, φ'' restricts to a bijection $\psi = \varphi \cup \varphi' : x \cup x' \simeq y \cup y'$. Finally, let $(a, b), (a', b') \in \psi$ be distinct. If they are both in φ or φ', then we can separate them by hypothesis. Otherwise one is in φ and the other in φ', in which case φ separates them. Finally, we have to check coincidence-freeness. Let $(a, b), (a', b') \in \varphi \in \mathcal{A} \wedge \mathcal{B}$. By hypothesis

$(a, b), (a', b')$ are separated by $\varphi' \subseteq \varphi$, such that $\varphi' : x' \simeq y'$ with $x' \in \mathcal{A}$ and $y' \in \mathcal{B}$. But then it follows automatically that $\varphi' \in \mathcal{A} \wedge \mathcal{B}$ as well.

It is easy to check that π_1, π_2 are maps in **Fam**, so we have a commuting square in **Fam**. It remains to prove that it is a pullback. Let $f : \mathcal{X} \to \mathcal{A}$, $g : \mathcal{X} \to \mathcal{B}$ making the outer square commute. On events, we simply define $\langle f, g\rangle\, e = (f\, e, g\, e)$ for $e \in X$. If $x \in \mathcal{X}$, then $\langle f, g\rangle\, x = \{(f\, e, g\, e) \mid e \in x\}$ is a bijection between configurations. For distinct $(f\, e_1, g\, e_1), (f\, e_2, g\, e_2) \in \langle f, g\rangle\, x$, since $\mathcal{X}$ is a configuration family there is some $y \in \mathcal{X}$ with $y \subseteq x$ that separates e_1 and e_2. Then, $\langle f, g\rangle\, y$ separates $(f\, e_1, g\, e_1)$ and $(f\, e_2, g\, e_2)$ as required, concluding the proof that $\langle f, g\rangle\, x \in \mathcal{A} \wedge \mathcal{B}$. Local injectivity follows from that of f or g. Uniqueness is clear from the requirement that $\langle f, g\rangle$ commutes with the projection, so we have indeed a pullback. ◀

A.2 Hiding on configuration families with polarities

In the main text, we were only concerned about hiding *all* neutral events. In this technical appendix (and in particular for the proof of associativity), it will be useful to consider a more partial notion of hiding, where some invisible events may be retained.

Let us fix a dcfpp $\mathcal{A}$, and V a subset of A whose elements are called the *visible* events, such that all events with non-zero polarity of A are in V; but some *neutral events* (*i.e.* events with polarity 0) might be as well. Write

$$A \downarrow V = \{x \cap V \mid x \in \mathcal{A}\}$$

for the candidate projection.

▶ **Lemma 36** (Lemma 13 of the paper). *Let $x, y \in \mathcal{A} \downarrow V$ such that $x \subseteq y$. Then, for any $x' \in \mathcal{A}$ a witness for x (*i.e. *$x' \cap V = x$), there is a witness $y' \in \mathcal{A}$ for y such that $x' \subseteq y'$.*

Proof. Note that for any diagram

$$\begin{array}{ccccc} x_{1,2} & & & & \\ \rotatebox{90}{$\subseteq$}\!\!\!-\!\!\!\!\!\! & & & & \\ x_{1,1} & \longrightarrow\!\!\!\!\!\subset & x_{2,1} & \subseteq & x_{n,1} \end{array}$$

such that every visible event in $x_{1,2}$ is in $x_{n,1}$, then either $x_{1,2} = x_{2,1}$, or there exists $x_{2,2}$ such that the following diagram commutes:

$$\begin{array}{ccccc} x_{1,2} & -\!\!\!\subset & x_{2,2} & & \\ \rotatebox{90}{$-\!\!\!\subset$} & & \rotatebox{90}{$-\!\!\!\subset$} & & \\ x_{1,1} & -\!\!\!\subset & x_{2,1} & \subseteq & x_{n,1} \end{array}$$

Indeed, if one of the coverings is neutral or positive we apply determinism, otherwise both are negative hence visible. In that case we can apply the completeness axiom exploiting that $x_{1,2} \subseteq x_{n,1}$ and $x_{2,1} \subseteq x_{n,1}$.

Take y'' any witness for y, and let us set $x_{1,1}$ to be a subconfiguration of x' still in y'' (it could be the empty configuration). This means (by Lemma 34) that we have two covering chains:

$$\begin{array}{cccccc} x_{1,1} & -\!\!\!\subset & x_{1,2} & -\!\!\!\subset & \dots & -\!\!\!\subset x_{1,p} = x' \\ x_{1,1} & -\!\!\!\subset & x_{2,1} & -\!\!\!\subset & \dots & -\!\!\!\subset x_{n,1} = y'' \end{array}$$

By iterating the reasoning above we can complete the grid $(x_{i,j})$ for $1 \leq i \leq n$ and $1 \leq j \leq p$, where every node relates to its upper and right neighbor via the reflexive closure of the covering relation. The lemma follows, as y' can be picked to be the upper-right corner of the grid. ◀

▶ **Lemma 37.** *Assume $x, y \in \mathcal{A}$ such that $x \cap V \uparrow y \cap V$ in $\mathcal{A} \downarrow V$, then $x \uparrow y \in \mathcal{A}$.*

Proof. Immediate application of Lemma 36. ◀

▶ **Proposition 38** (Proposition 14 of the paper)**.** Let $\mathcal{A}$ be a dcfpp. Then, $\mathcal{A} \downarrow V$ is a dcfpp.

Proof. Let us show first that it is a cfpp. Clearly, $\emptyset \in \mathcal{A} \downarrow V$. We show completeness. Let $x \cap V$ and $y \cap V \in \mathcal{A} \downarrow V$, such that $x \cap V \uparrow y \cap V$ in $\mathcal{A} \downarrow V$. By Lemma 37, we have $x \uparrow y$ in $\mathcal{A}$ as well. By completeness, $x \cup y \in \mathcal{A}$. But $(x \cup y) \cap V = (x \cap V) \cup (y \cap V) \in \mathcal{A} \downarrow V$ by definition. For coincidence-freeness, consider distinct $e_1, e_2 \in x \cap V \in \mathcal{A} \downarrow V$. Necessarily $e_1, e_2 \in x \in \mathcal{A}$ as well, and by coincidence-freeness there us $y \in \mathcal{A}$ such that $y \subseteq x$ and $e_1 \in y \Leftrightarrow e_2 \notin y$. But then $y \cap V \in \mathcal{A} \downarrow V$ separates them as well in $\mathcal{A} \downarrow V$.

Finally, let us prove determinism. Assume $x \cap V \subseteq^p y \cap V$ and $x \cap V \subseteq z \cap V$. *W.l.o.g.* we may assume that $x \subseteq^p y$ and $x \subseteq z$. By determinism of $\mathcal{A}$ it follows that $y \cup z \in \mathcal{A}$, so $(y \cup z) \cap V = (y \cap V) \cup (z \cap V) \in \mathcal{A} \downarrow V$. ◀

A.3 Composition of receptive prestrategies

Let $\sigma : \mathcal{A}^\perp \parallel \mathcal{B}$ and $\tau : \mathcal{B}^\perp \parallel \mathcal{C}$ be prestrategies. Recall that to define the interaction of σ and τ, we consider the pullback of the identity-on-events **Fam**-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 39** (Proposition 17 of the paper)**.** The pullback above is given by the family $\tau \circledast \sigma$ with events $|A| \parallel |B| \parallel |C|$ 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.* $p_1 \in y \Leftrightarrow p_2 \notin y$.

Proof. Unfolding of the definition in Proposition 35. ◀

▶ **Lemma 40.** *Let σ and τ be receptive prestrategies on $\mathcal{A}^\perp \parallel \mathcal{B}$ and $\mathcal{B}^\perp \parallel \mathcal{C}$. Then, $\tau \circledast \sigma$ is a dcfpp.*

Proof. We only need to prove one-step determinism, the general case follows by induction. Consider $x = x_A \parallel x_B \parallel x_C \in \tau \circledast \sigma$, with $x \overset{p_1}{\longrightarrow\!\!\subset}$ and $x \overset{p_2}{\longrightarrow\!\!\subset}$. Assume *w.l.o.g.* that p_1 is positive or neutral. Say *w.l.o.g.* that it is positive for σ. By determinism of σ, $x \cup \{p_1, p_2\} \in \sigma \parallel \mathcal{C}$, it remains to prove that $x \cup \{p_1, p_2\} \in \mathcal{A} \parallel \tau$ as well. If p_1 is in $\mathcal{A}$, this is obvious. If p_1 is in $\mathcal{B}$, then it follows from receptivity of τ. ◀

▶ **Lemma 41.** *Let σ and τ be receptive prestrategies on $\mathcal{A}^\perp \parallel \mathcal{B}$ and $\mathcal{B}^\perp \parallel \mathcal{C}$. Then, $\tau \odot \sigma$ is a receptive prestrategy on $\mathcal{A}^\perp \parallel \mathcal{C}$.*

Proof. We only need to prove receptivity, which is straightforward. ◀

Therefore, composition is a well-defined binary operation on receptive prestrategies.

A.3.0.1 Associativity.

We now prove that composition of receptive prestrategies is associative.

▶ **Lemma 42.** *Let $\sigma : \mathcal{A}^\perp \parallel \mathcal{B}$ and $\tau : \mathcal{B}^\perp \parallel \mathcal{C}$ be receptive prestrategies. Let*

$$x_A^0 \parallel x_C^0 \stackrel{s_1}{\longrightarrow\!\!\!\!\subset} x_A^1 \parallel x_C^1 \stackrel{s_2}{\longrightarrow\!\!\!\!\subset} \dots \stackrel{s_n}{\longrightarrow\!\!\!\!\subset} x_A^n \parallel x_C^n$$

be a covering chain (we write s for the sequence $s_1 \dots s_n$) from $x_A^0 \parallel x_C^0 \in \tau \odot \sigma$ to $x_A \parallel x_C = x_A^n \parallel x_C^n \in \tau \odot \sigma$, and let $x_B^0 \in \mathcal{B}$ be a configuration such that $x_A^0 \parallel x_B^0 \parallel x_C^0 \in \tau \circledast \sigma$. Then, there is a covering chain u

$$x_A^0 \parallel x_B^0 \parallel x_C^0 \longrightarrow\!\!\!\!\subset \dots \longrightarrow\!\!\!\!\subset x_A \parallel x_B \parallel x_C$$

of configurations in $\tau \circledast \sigma$, such that $s' \upharpoonright A, C = s$ (where $\upharpoonright A, C$ keeps only those events in A, C).

Proof. For each i, we construct x_B^i such that $x_A^i \parallel x_B^i \parallel x_C^i \in \tau \circledast \sigma$, and $x_B^i \subseteq x_B^{i+1}$. This is done by an immediate application of Lemma 36. For each inclusion $x_A^i \parallel x_B^i \parallel x_C^i$, there is by Lemma 34 a covering chain. Appending those, we get the required covering chain u. ◀

▶ **Proposition 43.** Composition is associative.

Proof. Let $\sigma : \mathcal{A}^\perp \parallel \mathcal{B}$, $\tau : \mathcal{B}^\perp \parallel \mathcal{C}$ and $\delta : \mathcal{C}^\perp \parallel \mathcal{D}$ be receptive prestrategies. We define the ternary interaction $\delta \circledast \tau \circledast \sigma$ as the sets $x_A \parallel x_B \parallel x_C \parallel x_D$ such that $x_A \parallel x_B \in \sigma$, $x_B \parallel x_C \in \tau$ and $x_C \parallel x_D \in \delta$, which has a covering chain among such sets. This unbiased interaction is crucial (as usual in game semantics) in the associativity proof, because from $x_A \parallel x_B \parallel x_C \parallel x_D \in \delta \circledast \tau \circledast \sigma$ it is clear that both $x_A \parallel x_B \parallel x_D \in (\delta \odot \tau) \circledast \sigma$ and $x_A \parallel x_C \parallel x_D \in \delta \circledast (\tau \odot \sigma)$ (by restricting the covering chain), so $x_A \parallel x_D$ belongs to both choices of association. So we only have to prove that reciprocally, any $x_A \parallel x_D$ in some choice of association has a witness in the unbiased interaction. This is proved using the lemma above.

We prove one case, the other being symmetric. Take $x = x_A \parallel x_D \in (\delta \odot \tau) \odot \sigma$. Then by definition, it has a witness $x_A \parallel x_B \parallel x_D \in (\delta \odot \tau) \circledast \sigma$, with a covering chain u (using Lemma 34). Then, $u \upharpoonright B, D$ is a covering chain for $x_B \parallel x_D \in \delta \odot \tau$. By Lemma 42, this covering chain has a witness v of some $x_B \parallel x_C \parallel x_D$ such that $v \upharpoonright B, D = u \upharpoonright B, D$. Then, take w any total ordering of $x_A \parallel x_B \parallel x_C \parallel x_D$ compatible with both u and v – it is easy to show from the constraint that $v \upharpoonright B, D = s \upharpoonright B, D$ that such a total ordering exists. Then, by construction $x_A \parallel x_B \parallel x_C \parallel x_D$ is in the ternary interaction $\delta \circledast \tau \circledast \sigma$. ◀

A.4 Copycat

▶ **Proposition 44.** Let $\mathcal{A}$ be a cfp. Then, defining $\text{\ae}_\mathcal{A}$ as having

- *Events, polarity.* Those of $\mathcal{A}^\perp \parallel \mathcal{A}$.
- *Configurations.* Those of the form $x \parallel y$ with $x, y \in \mathcal{A}$ and $y \sqsubseteq_\mathcal{A} x$.

we get that $\text{\ae}_\mathcal{A}$ is a cfp on events $A \parallel A$, with configurations included in those of $\mathcal{A}^\perp \parallel \mathcal{A}$.

Proof. Clearly $\emptyset \in \text{\ae}_\mathcal{A}$. We check completeness, and take $x = x_1 \parallel x_2, y = y_1 \parallel y_2 \in \text{\ae}_\mathcal{A}$ such that $x \uparrow y$. This means that there is $z = z_1 \parallel z_2 \in \text{\ae}_\mathcal{A}$ such that $x_1 \subseteq z_1, x_2 \subseteq z_2, y_1 \subseteq z_1$ and $y_2 \subseteq z_2$. By completeness, $x_1 \cup y_1 \in \mathcal{A}$ and $x_2 \cup y_2 \in \mathcal{A}$. Since $x \in \text{\ae}_\mathcal{A}$, there is $u \in \mathcal{A}$ such that:

$$x_2 \supseteq^- u \subseteq^+ x_1$$

Likewise, since $y \in \mathfrak{C}_\mathcal{A}$, there is $v \in \mathcal{A}$ such that:

$y_2 \supseteq^- v \subseteq^+ y_1$

So, we conclude that

$x_2 \cup y_2 \supseteq^- u \cup v \subseteq^+ x_1 \cup y_1$

which means that $x_1 \cup y_1 \sqsubseteq_\mathcal{A} x_2 \cup y_2$, so $x \cup y \in \mathfrak{C}_\mathcal{A}$.

For coincidence-freeness, take $x = x_1 \parallel x_2 \in \mathfrak{C}_\mathcal{A}$. We have $x_2 \sqsubseteq_\mathcal{A} x_1$. Two cases arise. Firstly, if $x_1 \neq x_2$, then we have

$x_2 \supseteq^- x_1 \cap x_2 \subseteq^+ x_1$

with, *e.g.* $x_1 \cap x_2 \subset^+ x_1$ (meaning the inclusion is strict). But then, by Lemma 34, there is a covering chain

$x_1 \cap x_2 {-\!\!\subset^+} \ldots x_1' {-\!\!\subset^+} x_1$

and $x' = x_1' \parallel x_2 \in \mathfrak{C}_\mathcal{A}$, with $x' {-\!\!\subset} x$.

If $x_1 = x_2$, then by coincidence-freeness of $\mathcal{A}$ (and Lemma 34), there is $x_2' {-\!\!\subset} x_1 = x_2$. Let us assume *w.l.o.g.* that the additional move is positive, the other case is symmetric. But then, we have $x_2', x_1 \in \mathcal{A}$ and:

$x_2' \sqsubseteq_\mathcal{A} x_1$

so by definition $x' = x_1 \parallel x_2' \in \mathbb{C}_\mathcal{A}$ and $x' {-\!\!\subset} x$. We have proved than in all cases, one could remove an event from $x \in \mathbb{C}_\mathcal{A}$, therefore by Lemma 34 it is coincidence-free. ◀

▶ **Proposition 45** (Proposition 19 of the paper)**.** If $\mathcal{A}$ is a cfp, then $\mathfrak{C}_\mathcal{A} : \mathbb{C}_\mathcal{A} \to \mathcal{A}^\perp \parallel \mathcal{A}$ is a prestrategy (in particular, is deterministic) iff $\mathcal{A}$ is race-free.

Proof. It is clear by construction that $\mathfrak{C}_\mathcal{A}$ is always receptive. We show that it is deterministic iff $\mathcal{A}$ is race-free.

If. Assume $\mathcal{A}$ is race-free. Take $x = x_l \parallel x_r \in \mathfrak{C}_\mathcal{A}$. We prove one-step determinism, the general case follows. Assume we have $x \overset{a_1}{-\!\!\subset}$ and $x \overset{a_2}{-\!\!\subset}$, with $\mathsf{pol}_{\mathcal{A}^\perp \parallel \mathcal{A}}(a_1) = +$. If they are both on the same side (say $\mathcal{A}^\perp$, and abusing notations slightly write $x_l \overset{a_1}{-\!\!\subset}$ and $x_l \overset{a_2}{-\!\!\subset}$), then either $\mathsf{pol}(a_2) = -$, in which case they are compatible in x_l by race-freeness, or $\mathsf{pol}(a_2) = +$, in which case it follows that $a_1, a_2 \in x_r$ as well. By race-freeness again x_l and x_r are compatible, hence $x_l \cup x_r \in \mathcal{A}$. But $x_l \cup \{a_1\} \subseteq x_l \cup x_r$ and $x_l \cup \{a_2\} \in x_l \cup x_r$, so $x_l \cup \{a_1, a_2\} \in \mathcal{A}$ by completeness. The condition with respect to the Scott order is obvious. If a_1 and a_2 are on different sides, the proof is obvious as well.

Only if. Assume that $\mathfrak{C}_\mathcal{A}$ is deterministic. Assume, looking for a contradiction, that $\mathcal{A}$ is not race-free. That means that there is $x \in \mathcal{A}$ with $x \overset{a_1^-}{-\!\!\subset}, x \overset{a_2^+}{-\!\!\subset}$, but $x \cup \{a_1, a_2\} \notin \mathcal{A}$. But then $x \parallel x \cup \{a_1\} \in \mathfrak{C}_\mathcal{A}$ with $x \parallel x \cup \{a_1\} {-\!\!\subset} x \cup \{a_2\} \parallel x \cup \{a_1\}, x \parallel x \cup \{a_1\} {-\!\!\subset} x \cup \{a_1\} \parallel x \cup \{a_1\}$, but their union is not a valid configuration, so $\mathfrak{C}_\mathcal{A}$ is not deterministic. ◀

▶ **Proposition 46** (Proposition 21 of the paper)**.** Let $\mathcal{A}$ be a race-free cfp. Then, the following are equivalent.

(i) The cfp $\mathcal{A}$ is co-race-free,
(ii) The relation $\sqsubseteq_\mathcal{A}$ is a partial order,

(iii) The prestrategy $\text{æ}_{\mathcal{A}}$ is idempotent, *i.e.* $\text{æ}_{\mathcal{A}} \odot \text{æ}_{\mathcal{A}} = \text{æ}_{\mathcal{A}}$.

Proof. *(i) ⇒ (ii).* Straightforward, by iterating co-race-freeness.

(ii) ⇒ (i). Assume that $\mathcal{A}$ is not co-race-free, meaning that there are

$$x_1 \stackrel{+}{\longrightarrow\!\!\!\!\subset} x \stackrel{-}{\supset\!\!\!\!\longrightarrow} x_2$$

such that $x_1 \cap x_2 \notin \mathcal{A}$. Then, we have $x_1 \sqsubseteq_{\mathcal{A}} x$ and $x \sqsubseteq_{\mathcal{A}} x_2$. Since $\sqsubseteq_{\mathcal{A}}$ is transitive, $x_1 \sqsubseteq_{\mathcal{A}} x_2$ as well, but then $x_1 \cap x_2 \in \mathcal{A}$, contradiction.

(ii) ⇔ (iii). Both configuration families $\text{æ}_{\mathcal{A}} \odot \text{æ}_{\mathcal{A}}$ and $\text{æ}_{\mathcal{A}}$ operate on events $A \parallel A$, therefore the isomorphism boils down to checking that $\text{æ}_{\mathcal{A}} \odot \text{æ}_{\mathcal{A}}$ and $\text{æ}_{\mathcal{A}}$ have the same configurations. We characterise the configurations of $\text{æ}_{\mathcal{A}} \odot \text{æ}_{\mathcal{A}}$. They have as witnesses configurations of $\text{æ}_{\mathcal{A}} \circledast \text{æ}_{\mathcal{A}}$, *i.e.* triples:

$$x_1 \parallel x_2 \parallel x_3 \in \mathcal{A} \parallel \mathcal{A} \parallel \mathcal{A}$$

such that $x_1 \parallel x_2 \in \text{æ}_{\mathcal{A}}$ and $x_2 \parallel x_3 \in \text{æ}_{\mathcal{A}}$, and that satisfy the coincidence-freeness condition. But in fact, the coincidence-freeness is redundant: it is straightforward to show that the projections conditions suffices to ensure that the triple has a covering chain.

Therefore, configurations of $\text{æ}_{\mathcal{A}} \circledast \text{æ}_{\mathcal{A}}$ just correspond to triples

$$x_1 \parallel x_2 \parallel x_3 \in \mathcal{A} \parallel \mathcal{A} \parallel \mathcal{A}$$

such that $x_3 \sqsubseteq_{\mathcal{A}} x_2$ and $x_2 \sqsubseteq_{\mathcal{A}} x_1$, so configurations $x_1 \parallel x_3 \in \text{æ}_{\mathcal{A}} \odot \text{æ}_{\mathcal{A}}$ correspond to triples such that there is some x_2 with $x_3 \sqsubseteq_{\mathcal{A}} x_2$ and $x_2 \sqsubseteq_{\mathcal{A}} x_1$, *i.e.* x_1 and x_3 are related by the relational composition of $\sqsubseteq_{\mathcal{A}}$ with itself. From there the equivalence is clear. ◀

A.5 Characterisation of strategies

Strategies $\sigma : \mathcal{A}^\perp \parallel \mathcal{B}$ are those receptive prestrategies such that $\text{æ}_{\mathcal{B}} \odot \sigma \odot \text{æ}_{\mathcal{A}} = \sigma$. Because copycat is idempotent, it is clear that copycat is a strategy, and that those are stable under composition; they form a category **Disj**. We wish however to have a more concrete characterization.

▶ **Definition 47.** Let σ be a prestrategy on $\mathcal{A}$. A **Scott pair** is a pair $x \parallel y \in \sigma \parallel \mathcal{A}$ such that $y \sqsubseteq_{\mathcal{A}} x$. We say that a Scott pair is **secured** if it has a covering chain amongst Scott pairs.

▶ **Definition 48.** Let σ be a prestrategy on $\mathcal{A}$. We say that σ satisfies the **secured discrete fibration property** iff for all secured Scott pair $x \parallel y$, we have $y \in \sigma$ as well.

The name comes from the discrete fibration property of strategies with respect to the Scott order in **CG** (see *e.g.* [7]). The condition above amounts to the fact that the identity-on-events **Fam**-map $\sigma \to \mathcal{A}$ has the discrete fibration property, restricted to those Scott-order related pairs that satisfy the additional *securedness* requirement.

▶ **Proposition 49.** Let σ be a receptive prestrategy on $\mathcal{A}$. Then, we have $\text{æ}_{\mathcal{A}} \odot \sigma = \sigma$ iff σ has the secured discrete fibration property.

Proof. *If.* By definition, $\text{æ}_{\mathcal{A}} \odot \sigma$ has for configurations those $x \in \mathcal{A}$ such that there exists $y \in \sigma$ with $x \sqsubseteq_{\mathcal{A}} y$, and $y \parallel x$ satisfies coincidence-freeness (*is secured*). This exactly means that $y \parallel x$ is a secured Scott pair, so $x \in \sigma$.

Only if. Take $y \in \sigma$, and $y \parallel x$ a secured Scott pair. But then as above $y \parallel x \in \text{æ}_{\mathcal{A}} \circledast \sigma$, so $x \in \text{æ}_{\mathcal{A}} \odot \sigma$, so $x \in \sigma$. ◀

The more subtle part of the proof is to prove the equivalence with the more concrete condition of *courtesy*. The formulation of courtesy in the paper may seem more restricted than usual, since it only mention commutations between positive events. In fact, in conjunction with receptivity and the fact that the strategy shares the events of the game it follows than the more general condition can be derived.

▶ **Lemma 50.** *Take a receptive courteous prestrategy σ on $\mathcal{A}$. Then, for all diagram in σ of the form*

$$\begin{array}{ccc} & x \cup \{a_1, a_2\} & \\ & \nearrow & \\ x \cup \{a_1\} & & \\ & \searrow & \\ & & x \end{array}$$

where $\mathsf{pol}(a_1) = +$ *or* $\mathsf{pol}(a_2) = -$*, then if* $\sigma\, x \cup \{\sigma\, a_2\} \in \mathcal{A}$*,* $x \cup \{a_2\} \in \sigma$ *as well.*

Proof. Straightforward. ◀

We first note that a strategy is always receptive and courteous. We will use the following lemma.

▶ **Lemma 51.** *Let σ be a receptive prestrategy on $\mathcal{A}$, and $x \in \sigma$. Then, $x \parallel \sigma\, x$ is a secured Scott pair.*

Proof. Immediate. ◀

▶ **Lemma 52.** *Let $\sigma : \mathcal{A}$ be a strategy (*i.e. $\text{æ}_{\mathcal{A}} \odot \sigma = \sigma$*), then it is courteous.*

Proof. By Proposition 49, we know that σ satisfies the secured discrete fibration property.

Take a diagram in σ:

$$\begin{array}{ccc} & x \cup \{a_1, a_2\} & \\ & \nearrow & \\ x \cup \{a_1\} & & \\ & \searrow & \\ & & x \end{array}$$

such that $\mathsf{pol}(a_1) = \mathsf{pol}(a_2) = +$ and $x \cup \{a_2\} \in \mathcal{A}$. But then, $x \cup \{a_1, a_2\} \parallel x \cup \{a_2\}$ is a Scott pair. It is a *secured* Scott pair, as one can remove $(2, a_2), (1, a_2), (1, a_1)$ (in that order) while remaining within Scott pairs, and $x \parallel x$ is secured by Lemma 51. Therefore, by the secured discrete factorization property, we have $x \cup \{a_2\} \in \sigma$ as required. ◀

We have established that strategies are receptive and courteous; we will now prove the converse: that receptive and courteous prestrategies are indeed strategies. We aim to prove that they have the secured discrete fibration property. For that, we need to establish first a few lemmas concerning covering chains of secured Scott pairs for receptive and courteous prestrategies.

▶ **Lemma 53.** *Let σ be a receptive and courteous prestrategy on $\mathcal{A}$. Assume have the following coverings between secured Scott pairs:*

$$\begin{array}{ccc} x_2 & \parallel & y \\ {\scriptstyle a_2^-}\Upsilon & & \\ x_1 & \parallel & y \\ {\scriptstyle a_1^+}\Upsilon & & \\ x_0 & \parallel & y \end{array}$$

Then, we have the following coverings between secured Scott pairs:

$$\begin{array}{ccc} x_2 & \parallel & y \\ {\scriptstyle a_1^+}\Upsilon & & \\ x_1' & \parallel & y \\ {\scriptstyle a_2^-}\Upsilon & & \\ x_0 & \parallel & y \end{array}$$

Proof. We have to prove that $x_1' \parallel y$ is a Scott pair. First, we need to prove that $x_1' \in \sigma$, but by courtesy (and Lemma 50) it suffices to show that $x_1' \in \mathcal{A}$. But we have $y \supseteq^- \subseteq^+ x_2$, so since $\mathcal{A}$ is co-race-free we have $y \cap x_2 \in \mathcal{A}$. But $a_1 \notin y$, as that would imply that $a_1 \in x_0$, absurd. However, we have $a_2 \in y \cap x_2 \in \mathcal{A}$ and

$$y \cap x_2 \subseteq x_0 \cup \{a_2\}$$

Since $\mathcal{A}$ is race-free, we also have that $x_2 \cup y \in \mathcal{A}$. Therefore, $x_0 \uparrow_{\mathcal{A}} (y \cap x_2)$, hence $x_0 \cup (y \cap x_2) \in \mathcal{A}$. But from the above it is clear that $x_0 \cup (y \cap x_2) = x_1'$. Finally, it is clear that all pairs involved are Scott pairs. Securedness is preserved by construction. ◀

▶ **Lemma 54.** *Let σ be a receptive and courteous prestrategy on $\mathcal{A}$. Assume we have the following coverings between secured Scott pairs:*

$$\begin{array}{ccc} x' & \parallel & y' \\ & & {\scriptstyle a_2}\Upsilon \\ x' & \parallel & y \\ {\scriptstyle a_1^+}\Upsilon & & \\ x & \parallel & y \end{array}$$

where $\sigma\, s \neq a$. Then, we have the following coverings between secured Scott pairs:

$$\begin{array}{ccc} x' & \parallel & y' \\ {\scriptstyle a_1^+}\Upsilon & & \\ x & \parallel & y' \\ & & {\scriptstyle a}\Upsilon{\scriptstyle 2} \\ x & \parallel & y \end{array}$$

Proof. Straightforward. ◀

We would also like to have a +/+ commutation on the left; however that is not true in general. What is true is the following lemma, which will serve the same purpose.

▶ **Lemma 55.** *Let σ be a receptive and courteous prestrategy on $\mathcal{A}$. Assume we have the following coverings between secured Scott pairs:*

$$\begin{array}{rcl} x_2 & \parallel & y' \\ & & {\scriptstyle a_2^+}\cup \\ x_2 & \parallel & y \\ {\scriptstyle a_2^+}\cup & & \\ x_1 & \parallel & y \\ {\scriptstyle a_1^+}\cup & & \\ x_0 & \parallel & y \end{array}$$

Then, we have the following coverings between secured Scott pairs:

$$\begin{array}{rcl} x_2 & \parallel & y' \\ & & {\scriptstyle a_2^+}\cup \\ x_2 & \parallel & y \\ {\scriptstyle a_1^+}\cup & & \\ x_1' & \parallel & y \\ {\scriptstyle a_2^+}\cup & & \\ x_0 & \parallel & y \end{array}$$

Proof. By courtesy, this amounts to proving that $x_1' \in \mathcal{A}$. It is immediate to check that

$$x_0 \cap y \stackrel{a_2}{-\!\!\!\subset} x_2 \cap y'$$

which are in $\mathcal{A}$ since $\mathcal{A}$ is co-race-free.

Moreover $x_2 \cup y' \in \mathcal{A}$ since $\mathcal{A}$ is race-free, so x_0 is compatible with the two configurations above. So by completeness we can take the unions, and obtain in $\mathcal{A}$:

$$x_0 \stackrel{a_2}{-\!\!\!\subset} x_0 \cup (x_2 \cap y')$$

from which it follows that $x_1' = x_0 \cup (x_2 \cap y') \in \mathcal{A}$. Hence by courtesy, $x_1' \in \sigma$. The fact that these are Scott pairs is straightforward, and securedness is preserved by construction. ◀

▶ **Lemma 56.** *Let σ a receptive and courteous prestrategy on $\mathcal{A}$. For all secured Scott pair $x \parallel y$, there exists a covering chain (within Scott pairs) such that:*

- *Every positive event a^+ on the right occurs immediately after a on the left,*
- *Every positive events in x that do not occur in y happen after the last event of y, and after all negative events in x.*

We say that such a covering chain is in ***normal form****.*

Proof. Consider a covering chain of $x \parallel y$. Call a positive event of x *postponed* if it occurs after all events in y, and after all negative events in x.

Consider the last positive move in x that is not postponed, and that does not immediately precedes the corresponding event in y. Then it is an immediate verification that with Lemmas 53, 54 and 55 it can be pushed until it hits the corresponding event in y, or until it is postponed. Indeed the only case where it cannot be pushed up is if the previous move is a positive move in x not immediately before the corresponding event in y. But then by hypothesis this event is postponed, so the event we are pushing is postponed as well. This operation is performed until the covering chain is in normal form. ◀

▶ **Proposition 57.** Let σ be a receptive and courteous prestrategy on $\mathcal{A}$. Then, σ satisfies the secured discrete fibration property.

Proof. We prove by induction on $|x \setminus (x \cap y)|$, *i.e.* on the cardinality of the positive inclusion $x \cap y \subseteq^+ x$, that for all secured Scott pair $x \parallel y$, we have $y \in \sigma$. By Lemma 56, consider a covering chain of $x \parallel y$ in normal form. We distinguish cases, depending on its last move.

If the last move is $x' \stackrel{s^+}{\longrightarrow\!\!\subset} x$, we still have a secured Scott pair $x' \parallel y$. By induction hypothesis, $y \in \sigma$.

If the last move is something else, then by definition of the normal form it follows that $x \subseteq^- y \in \mathcal{A}$. Then, $y \in \sigma$ as well by receptivity. ◀

All in all, we have proved:

▶ **Theorem 58** (Theorem 22 of the paper). *Let σ be a receptive prestrategy on $\mathcal{A}^\perp \parallel \mathcal{B}$. Then, $\text{cc}_\mathcal{B} \odot \sigma \odot \text{cc}_\mathcal{A} = \sigma$ iff σ is courteous.*

Proof. For σ a receptive prestrategy on $\mathcal{A}$, it follows from Propositions 49 and 57 that $\text{cc}_\mathcal{A} \odot \sigma = \sigma$ iff σ is courteous. It is immediate to check that $\text{cc}_{\mathcal{A}^\perp \parallel \mathcal{B}} \odot \sigma = \text{cc}_\mathcal{B} \odot \sigma \odot \text{cc}_\mathcal{A}$, so the statement above follows. ◀

A.6 Compact closed and SMCC structures

On objects, the tensor $\mathcal{A} \otimes \mathcal{B}$ is simply the parallel composition $\mathcal{A} \parallel \mathcal{B}$. On strategies, given $\sigma_1 : \mathcal{A}_1^\perp \parallel \mathcal{B}_1$ and $\sigma_2 : \mathcal{A}_2^\perp \parallel \mathcal{B}_2$, we define $\sigma_1 \otimes \sigma_2$ to comprise configurations $(x_{A_1} \parallel x_{A_2}) \parallel (x_{B_1} \parallel x_{B_2})$ such that $x_{A_1} \parallel x_{B_1} \in \sigma_1$ and $x_{A_2} \parallel x_{B_2} \in \sigma_2$. All the conditions are easily verified.

▶ **Lemma 59.** *The definitions above extend $\otimes$ to a bifunctor*

$$- \otimes - : \mathbf{Disj} \times \mathbf{Disj} \to \mathbf{Disj}$$

Proof. Straightforward verifications. ◀

A.6.0.1 Lifting.

Take $\mathcal{A}$, $\mathcal{B}$ games, and $f : \mathcal{A} \to \mathcal{B}$ a **Fam**-isomorphism that preserves polarity. Then, consider

$$\overline{f} = \{x_A \parallel x_B \mid x_B \sqsubseteq_B f\, x_A\}$$

The fact that this is a receptive pre-strategy follows exactly the same proof as for copycat. We prove the key *lifting lemma*:

▶ **Lemma 60.** *Let* $\sigma : \mathcal{A}^\perp \parallel \mathcal{B}$ *and* $f : \mathcal{B} \to \mathcal{C}$ *be a polarity-preserving* **Fam**-*isomorphism. Then,*

$$\overline{f} \odot \sigma = \{x_A \parallel f\, x_B \mid x_A \parallel x_B \in \sigma\}$$

Proof. Direct consequence of the neutrality of copycat by composition. ◀

From that it is easy to prove that lifting is functorial. Using it we construct the compact closed structure as in [7]: from *e.g.* the symmetry iso $s_{\mathcal{A},\mathcal{B}} : \mathcal{A} \otimes \mathcal{B} \to \mathcal{B} \otimes \mathcal{A}$ in **Fam** we get $\overline{s_{\mathcal{A},\mathcal{B}}} : \mathcal{A} \otimes \mathcal{B} \overset{\mathbf{Disj}}{\twoheadrightarrow} \mathcal{B} \otimes \mathcal{B}$. Same thing for the associativity, the unit, *etc.* The coherence laws follow by functoriality, and naturality by Lemma 60. Writing 1 for the empty game, copycat yields strategies

$$\eta_{\mathcal{A}} : 1 \overset{\mathbf{Disj}}{\twoheadrightarrow} (\mathcal{A}^\perp \otimes \mathcal{A}) \qquad\qquad \epsilon_{\mathcal{A}} : (\mathcal{A} \otimes \mathcal{A}^\perp) \overset{\mathbf{Disj}}{\twoheadrightarrow} 1$$

and the coherence law for compact closed categories is a direct verification, amounting again to the idempotence of copycat.

A.6.0.2 Disj_.

We start by proving:

▶ **Lemma 61.** *The empty game* 1 *is terminal in* **Disj_**.

Proof. From any negative game $\mathcal{A}$, the empty strategy $\emptyset : \mathcal{A}^\perp \parallel 1$ is receptive, and is obviously courteous. It is also the unique negative strategy on $\mathcal{A}^\perp \parallel 1$. Indeed, assume negative $\sigma : \mathcal{A}^\perp \parallel 1$ has a non-empty configuration. Necessarily it has a covering chain. Take the first step of this covering chain: it is a singleton configuration, and necessarily the single event must be negative as σ is negative. But then it has to map to $\mathcal{A}^\perp$, but since $\mathcal{A}$ is negative, singleton configurations in $\mathcal{A}^\perp$ can only contain a positive event, contradiction. ◀

▶ **Lemma 62.** *For any* $\mathcal{A}$, $\text{æ}_{\mathcal{A}}$ *is negative.*

Proof. Any singleton configuration can only contain a negative event by definition of copycat. But any $x \in \text{æ}_{\mathcal{A}}$ must have a covering chain, therefore must contain a negative event. ◀

▶ **Lemma 63.** *The composition of negative* $\sigma : \mathcal{A}^\perp \parallel \mathcal{B}$ *and* $\tau : \mathcal{B}^\perp \parallel \mathcal{C}$ *is negative.*

Proof. Take a singleton $x = x_A \parallel x_C \in \tau \odot \sigma$. Say *e.g.* that $x_A = \emptyset$ and $x_C = \{c\}$. Take its witness $\emptyset \parallel x_B \parallel \{c\} \in \tau \circledast \sigma$. Take a covering chain for it. If it starts in B, then the projection to either σ or τ is positive, absurd. Otherwise the first move of the covering chain is c, and $\emptyset \parallel \{c\} \in \tau$. But τ is negative, so c is negative. The other case is dual. ◀

Therefore there is a subcategory **Disj_** of **Disj** with negative games as objects, and negative strategies as morphisms. This subcategory has a terminal object 1. Just like the fact that copycat is negative, lifted strategies are negative as well, and so the symmetric monoidal structure of **Disj** transfers to **Disj_**. However, the compact closed structure does not: indeed, if $\mathcal{A}$ and $\mathcal{B}$ are negative, $\mathcal{A}^\perp \parallel \mathcal{B}$ is not necessarily. Hence, in the paper we introduce $\mathcal{A} \multimap \mathcal{B}$ as a negative counterpart to $\mathcal{A}^\perp \parallel \mathcal{B}$. We prove:

▶ **Lemma 64.** *For negative* $\mathcal{A}$ *and* $\mathcal{B}$, $\mathcal{A} \multimap \mathcal{B}$ *is a negative game.*

Proof. All verifications are direct. ◀

Finally, we have:

▶ **Proposition 65.** Let $\mathcal{A}, \mathcal{B}, \mathcal{C}$ be negative game. Let σ be a configuration family on events $A \parallel B \parallel C$. Then, σ is a negative strategy on $\mathcal{A}^\perp \parallel \mathcal{B}^\perp \parallel \mathcal{C}$ iff it is a negative strategy on $\mathcal{A}^\perp \parallel \mathcal{B} \multimap \mathcal{C}$.

Proof. By negativity of $\mathcal{A}, \mathcal{B}$ and σ, any non-empty $x_A \parallel x_B \parallel x_C \in \sigma$ must be such that x_C is non-empty, so $x_B \parallel x_C \in \mathcal{B} \multimap \mathcal{C}$. ◀

▶ **Theorem 66.** **Disj**$_-$ *is a symmetric monoidal closed category with a terminal object.*

Proof. Follows directly from Proposition 65 and the compact closed structure of **Disj**. ◀

A.6.0.3 Products.

It is straightforward that if $\mathcal{A}$ and $\mathcal{B}$ are negative games, then so is $\mathcal{A} \,\&\, \mathcal{B}$. Negativity is important here, to ensure that we keep race-freeness. The first projection $\varpi_\mathcal{A} : (\mathcal{A} \,\&\, \mathcal{B})^\perp \parallel \mathcal{A}$ comprises the configurations

$$(x^l_A \parallel \emptyset) \parallel x^r_A$$

for $x^r_A \sqsubseteq_\mathcal{A} x^l_A$. The second projection is defined likewise. From negative $\sigma : \mathcal{A}^\perp \parallel \mathcal{B}$ and $\tau : \mathcal{A}^\perp \parallel \mathcal{C}$, $\langle \sigma, \tau \rangle$ comprises:

$$\{x_A \parallel (x_B \parallel \emptyset) \mid x_A \parallel x_B \in \sigma\} \cup \{x_A \parallel (\emptyset \parallel x_C) \mid x_A \parallel x_C \in \tau\}$$

Interactions $\varpi_B \circledast \langle \sigma, \tau \rangle$ are in one-to-one correspondence with those in $æ_B \circledast \sigma$, from which follows that $\varpi_B \odot \langle \sigma, \tau \rangle = \sigma$. Sujective pairing is straightforward.

B Construction of Odet

First, we prove the following.

▶ **Lemma 67.** *Let $\sigma : S \to A$ be deterministic, then it is also odet.*

Proof. It is straightforward that $\mathcal{O}(\sigma)$ is a disjunctive deterministic strategy. The extension property follows directly from the fact that as a deterministic strategy, the map σ is injective on configurations (Lemma 5 of [22]). ◀

As a result, all structural morphisms in **CG**$_-$, being deterministic (since arenas are race-free [22]), are odet. The tensor and pairing of strategies clearly preserve odet as well. The key point to check is that odet strategies are preserved under composition as well, which we focus on now.

▶ **Lemma 68.** *Let $\sigma : S \to A^\perp \parallel B$ and $\tau : T \to B^\perp \parallel C$ be odet. Then, $\tau \circledast \sigma : T \circledast S \to A \parallel B \parallel C$ is also a* **Fam***-morphism:*

$$\tau \circledast \sigma : \mathscr{C}(T \circledast S) \to \mathcal{O}(\tau) \circledast \mathcal{O}(\sigma)$$

with the configuration extension property.

As a corollary, $\mathcal{O}(\tau \circledast \sigma) = \mathcal{O}(\tau) \circledast \mathcal{O}(\sigma)$.

Proof. We check first that it preserves configurations. Let $x \in \mathscr{C}(T \circledast S)$. From [7], we know that x corresponds to a secured bijection

$$\varphi = \cup x : x_S \parallel x_C \simeq x_A \parallel x_T$$

where $\sigma\, x_S = x_A \parallel x_B$, $\tau\, x_T = x_B \parallel x_C$. Then, $(\tau \circledast \sigma)\, x = x_A \parallel x_B \parallel x_C \in \mathcal{O}(\tau) \circledast \mathcal{O}(\sigma)$ as by construction $x_A \parallel x_B \in \mathcal{O}(\sigma)$, $x_B \parallel x_C \in \mathcal{O}(\tau)$, and it has a covering chain, obtained by transporting any covering chain of (secured) x. Local injectivity is clear from local injectivity of σ and τ.

We prove the configuration extension property. It is sufficient to prove the one-step version, the general one follows immediately by induction. Let $x \in \mathscr{C}(T \circledast S)$, and $(\tau \circledast \sigma)\, x = x_A \parallel x_B \parallel x_C \in \mathcal{O}(\tau) \circledast \mathcal{O}(\sigma)$. We know (from [7]) that x canonically corresponds to a secured bijection

$$\varphi : x_S \parallel x_C \simeq x_A \parallel x_T$$

with $\sigma\, x_S = x_A \parallel x_B$ and $\tau\, x_T = x_B \parallel x_C$. Assume that

$$x_A \parallel x_B \parallel x_C {-\!\!\subset} x'_A \parallel x'_B \parallel x'_C \in \mathcal{O}(\tau) \circledast \mathcal{O}(\sigma)$$

If the extension is negative in $A^\perp \parallel C$, then we apply receptivity of σ or τ, and obtain the extension of φ (trivially still secured) in the obvious way. Otherwise, the extension is positive for σ or τ. Assume *w.l.o.g.* that it is for σ. It it is positive in B, then

$$\sigma\, x_S = x_A \parallel x_B \overset{(2,b)}{-\!\!\subset} x_A \parallel x'_B \in \mathcal{O}(\sigma)$$

By the extension property for σ, there is $x_S \overset{s^+}{-\!\!\subset} x'_S$ such that $\sigma\, x'_S = x_A \parallel x'_B$. By receptivity for τ, there is a (unique) $x_T \overset{t^-}{-\!\!\subset} x'_T$ such that $\tau\, x'_T = x'_B \parallel x_C$. The other cases are symmetric or simpler.

By applying the extension property, we deduce that indeed $\mathcal{O}(\tau \circledast \sigma) = \mathcal{O}(\tau) \circledast \mathcal{O}(\sigma)$. ◀

We deduce:

▶ **Proposition 69.** The map $\tau \odot \sigma : T \odot S \to A^\perp \parallel B$ is also a **Fam**-morphism

$$\tau \odot \sigma : \mathscr{C}(T \odot S) \to \mathcal{O}(\tau) \odot \mathcal{O}(\sigma)$$

with the extension property. Therefore, odet strategies are stable under composition.

Proof. The fact that it is a **Fam**-morphism with the extension property follows direct from Lemma 68. As a direct consequence, by the extension property, $\mathcal{O}(\tau \odot \sigma) = \mathcal{O}(\tau) \odot \mathcal{O}(\sigma)$. As **Disj**-strategies are stable under composition, $\mathcal{O}(\tau \odot \sigma)$ is a **Disj**-strategy as required, so $\tau \odot \sigma$ is odet. ◀

Putting everything together, along with direct verifications, we obtain:

▶ **Theorem 70** (Theorem 26 of the paper). *There is a compact closed category* **Odet** *with arenas as objects and odet strategies up to* $\approx$ *as morphisms. Moreover,* **Odet** *has a symmetric monoidal subcategory* $\mathbf{Odet}_-$ *of negative arenas and negative strategies; with products and* 1 *terminal. It admits well-opened arenas as an exponential ideal.*

C Intensional full abstraction

The construction of the category **PorStrat** relies on the proof of stability by composition of [8] which are not reproduced here. We focus now on the proof of intensional full abstraction. In the following, "strategy" will mean $\mathbf{aPCF}_{\mathtt{por}}$-strategy.

▶ **Lemma 71.** *Let $\sigma : S \to A$ be a strategy. Any $x \in \mathscr{C}(S)$ can be extended to $\overline{x}$ such that the restriction of σ to $\overline{x}$ defines a strategy written $\overline{x} : A$.*

Proof. $\overline{x}$ is obtained by closing under receptivity (adding the negative events enabled from a subset of x). ◀

Such strategies are instances of path-strategies. A strategy $\sigma : [\![A]\!]$ is a **path-strategy** if there exists a configuration $x \in \mathscr{C}(S)$ containing all positive moves of σ (in that case, $\sigma \cong \overline{x}$). It turns out that path-strategies have all the expressive power needed to distinguish strategies.

▶ **Lemma 72.** *Let σ and τ be strategies on A. If there exists a strategy $\alpha : A^\perp \parallel \mathbb{B}$ that distinguishes them, then there exists a path-strategy distinguishing them.*

Proof. Without loss of generality we can assume that $\alpha \odot \sigma$ converges (ie. contains a positive event) whereas $\alpha \odot \tau$ does not. Let $p \in \alpha \odot \sigma$ be such a positive event. Closing down inside the interaction $\alpha \circledast \sigma$ gives us a configuration z of the pullback. Since $\Pi_2 z \in \mathscr{C}(\alpha)$, it induces a strategy $\overline{\Pi_2 z}$ on $A^\perp \parallel \mathbb{B}$. By construction $\overline{\Pi_2 z} \odot \sigma$ converges. If $\overline{\Pi_2 z} \odot \tau$ converges, then the corresponding convergence witness would be a convergence witness for $\alpha \odot \tau$, absurd. ◀

We now show that path strategies enjoy finite definability – which is then enough to conclude for full abstraction by standard semantic means:

▶ **Lemma 73.** *Assuming that for every path-strategy $\sigma : A$ there exists a term t such that $[\![t]\!] \simeq_{\mathrm{obs}} \sigma$, then* **PorStrat** *is fully-abstract: for closed terms $\vdash t, t' : A$, $t \simeq_{\mathrm{obs}} t'$ if and only if $[\![t]\!] \simeq_{\mathrm{obs}} [\![t']\!]$.*

Proof. Assume $[\![t]\!] \simeq_{\mathrm{obs}} [\![t']\!]$. Let $\mathcal{C}$ be a context such that $\mathcal{C}[t] \Downarrow$. Then $[\![\mathcal{C}]\!] \odot [\![t]\!]$ converges and by assumption so does $[\![\mathcal{C}]\!] \odot [\![t']\!]$. By adequacy, this means that $\mathcal{C}[t']$ converges as well.

Conversely, if $t \simeq_{\mathrm{obs}} t'$, assume that $[\![t]\!]$ and $[\![t']\!]$ are not observationally equivalent. Then there exists a test $\alpha : A^\perp \parallel \mathbb{B}$ such that for instance $\alpha \odot [\![t]\!]$ converges but not $\alpha \odot [\![t']\!]$. By Lemma 72, we can assume that α is a path-strategy. Hence there exists a term c such that $[\![c]\!] \simeq_{\mathrm{obs}} \alpha$. By adequacy $c\,t$ converges, and so does $c\,t'$ by assumption. Hence by soundness $[\![c\,t']\!] \simeq_{\mathrm{obs}} \alpha \odot [\![t']\!]$. ◀

The rest of the section is dedicated to prove the assumption, by induction on the size of path-strategies (which are all finite because types have finite denotations).

C.1 Ground and first-order types

We use the notation $X^n \multimap Y$ to denote $X \multimap \ldots \multimap X \multimap Y$ with n occurrences of X.

▶ **Lemma 74.** *Any path-strategy on $\mathbb{B}$ is display-equivalent to either $[\![\bot]\!], [\![\mathtt{tt}]\!], [\![\mathtt{ff}]\!]$.*

Proof. Obvious, by case analysis on configurations of $\mathbb{B}$. ◀

We show that observational equivalence at higher-order types boils down to extensional equivalence:

▶ **Lemma 75.** *Two* $\mathbf{aPCF}_{por}$*-strategies* $\sigma, \tau : [\![A_1 \Rightarrow \ldots \Rightarrow A_i \Rightarrow \mathbb{B}]\!]$ *are observationally equivalent if and only if for all* $\upsilon_i : [\![A_i]\!]$*, then* $\sigma \odot \langle \upsilon_1, \ldots, \upsilon_n \rangle \simeq_{\mathrm{obs}} \tau \odot \langle \upsilon_1, \ldots, \upsilon_n \rangle$*.*

Proof. Clearly, if they are observationally equivalent, they are extensionally equivalent: the argument list $(\upsilon_1, \ldots, \upsilon_n)$ can be tested via $\lambda f.\, f\, \upsilon_1\, \ldots\, \upsilon_n$.

Conversely assume that σ and τ are extensionally equivalent. Let α be a strategy on $A^\perp \parallel \mathbb{B}$. By Lemma 72, α can be assumed to be a path-strategy. If α does not play in $A^\perp$, $\alpha \odot \sigma$ and $\alpha \odot \tau$ have the same outcome. Otherwise, if it converges, then in a causal history of one of the answers in $\mathbb{B}$, α plays the initial question and by receptivity, the minimal negative questions for each of the A_i. These questions enable strategies υ_i on $[\![A_i]\!]$ that, by innocence are disjoint. By well-bracketing, the result of $\alpha \odot \sigma$ can only depend on the result of σ on those υ_i, so by assumption it will be the same outcome as for $\alpha \odot \tau$. ◀

▶ **Lemma 76.** *For any path-strategy* $\sigma : [\![\mathbb{B}^n \multimap \mathbb{B}]\!]$*, there exists a term* $t : \mathbb{B}^n \multimap \mathbb{B}$ *such that* $[\![t]\!] \simeq_{\mathrm{obs}} \sigma$*.*

Proof. We proceed by induction on n just like in [8]. If $n = 0$, then σ is simply a boolean and this covered by Lemma 74. Otherwise, write x for the configuration of σ such that $\overline{x} \cong \sigma$. there are two cases:

- Either there are no positive answers in x, then $\overline{x} \simeq_{\mathrm{obs}} [\![\perp]\!]$
- Or there is a (unique) positive answer $a \in x$. Then, the first remark is that $\overline{[a]} \simeq_{\mathrm{obs}} \sigma$ so we can assume $x \cong [a]$. Take $s^+ \in [a]$ to be a positive move, minimal among positive move. If $s = a$, then $\overline{x} \simeq_{\mathrm{obs}} [\![a]\!]$ (where $a \in \{\mathtt{tt}, \mathtt{ff}\}$). Otherwise, s is a positive question, assume without loss of generality that it is the initial question of the first argument. In $[a]$, there must be exactly one answer to q, written a_0^-. In that case $\overline{x} \downarrow (\overline{x} \setminus \{\mathrm{q}, a_0\})$ is a strategy on $\mathbb{B}^{n-1} \multimap \mathbb{B}$. By induction, there exists a term t_0 whose interpretation is this strategy. If for instance $a_0 = \mathtt{tt}$, then the following term

$$\lambda x_1 \ldots x_n.\, \mathbf{if}\, x_1\, (t_0\, x_2\, \ldots\, x_n) \perp$$

  has an interpretation observationally equivalent to $\overline{x}$ (by Lemma 75).

◀

C.2 Higher-order types

Let $A = A_1 \Rightarrow \ldots \Rightarrow A_n \Rightarrow \mathbb{B}$ be a higher-order type and $\sigma : [\![A]\!]$. Following [8], a path-strategy $\sigma : [\![A]\!]$ splits into a *flow substrategy* which has a first-order type and *arguments substrategies* that are smaller. To proceed we thus do an induction on the size of S.

C.2.1 Flow substrategy.

The flow substrategy of σ describes which arguments it evaluates, and which order. It will play on the first-order type $\mathbb{B}^n \multimap \mathbb{B}$. Note that this is a subarena of $[\![A]\!]$. Its underlying event structure is defined as: $S_{\mathrm{flow}} = S \downarrow \{s \in S \mid \sigma[s] \subseteq \mathbb{B}^n \multimap \mathbb{B}\}$.

There is a natural map $\sigma_{\mathrm{flow}} : S_{\mathrm{flow}} \to \mathbb{B}^n \multimap \mathbb{B}$ that is a path-strategy. A positive question $\mathrm{q}^+ \in S_{\mathrm{flow}}$ is a **primary question**. By linearity, and the fact that σ_{flow} is a path-strategy, there is at most n primary questions. If q calls the ith argument we say its **index** is i.

C.2.2 Argument substrategies.

Write $\Gamma = A_1, \dots, A_n$. Let $1 \leq i \leq n$. Let q be a primary question of index i. Write $A_i = A_{i,1} \multimap \dots \multimap A_{i,n_i} \multimap \mathbb{B}$. For each each $1 \leq j \leq n_i$, there is a negative question $\mathrm{q}_j^- \in S$ (by receptivity) corresponding to the jth argument of the call initiated by q. Define $S_{\mathrm{q},j}$ to be the event structure $S \downarrow \{s \in S \mid s \geq \mathrm{q}_j\}$. The image of $S_{\mathrm{q},j}$ does not quite live in $[\![A_k]\!]$ since a call to another $[\![A_i]\!]$ is necessary to compute the arguments – eg. in $\lambda f x.\, f\, x$ where a call to x is necessary to compute the argument to f. However, up to reindexing they is a path-strategy

$$\sigma_{\mathrm{q},j} : S_{\mathrm{q},j} \to (\Gamma_{\mathrm{q},j} \multimap A_{i,j})$$

where $\Gamma_{\mathrm{q},j}$ is the subset of the context used by the argument $\sigma_{\mathrm{q},j}$ (A_k belongs to $\Gamma_{\mathrm{q},j}$ if the initial question of A_k belongs to $S_{\mathrm{q},j}$).

By local injectivity, all $\Gamma_{\mathrm{q},j}$ are all disjoint and cannot contain A_i, so we build

$$\sigma_q : [\![\Gamma_q]\!] \xrightarrow{[\![A_i]\!] \parallel \sigma_{\mathrm{q},1} \parallel \dots \parallel \sigma_{\mathrm{q},n_i}} [\![A_i]\!] \parallel [\![A_{i,1}]\!] \parallel \dots \parallel [\![A_{i,n_i}]\!] \xrightarrow{ev} \mathbb{B}$$

where $\Gamma_q = A_i, \Gamma_{\mathrm{q},0}, \dots, \Gamma_{\mathrm{q},j}$.

Because σ is a path-strategy, all primary questions are consistent and have different index. As a consequence, all the Γ_{q} are disjoint. For $1 \leq i \leq n$ define σ_i to be σ_{q} if q is the primary question of index i, or $\sigma_i = \bot$ if there are no primary question of index i (σ does not use its ith argument).

▶ **Lemma 77.** *We have the following:*

$$\sigma \simeq_{\mathrm{obs}} \sigma_{\mathit{flow}} \odot (\sigma_1 \parallel \dots \parallel \sigma_{n_i})$$

Proof. First, as a consequence of well-bracketing and innocence, the flow and the argument strategies form a partition of S.

Take $x \in \mathscr{C}(S)$. Consider $x_{\mathrm{flow}} = x \cap S_{\mathrm{flow}}$, and for $1 \leq i \leq n$ and q the corresponding question

$$x_{A_i} = x \cap A_i \quad x_{\mathbb{B}_i} = x \cap \mathbb{B}_i \quad x_{\mathrm{q},j} = x \cap S_{\mathrm{q},j}.$$

Write $\sigma^1_{\mathrm{q},j}$ and $\sigma_{\mathrm{q},j}$ for the obvious partial projections, to $\Gamma^\perp_{\mathrm{q},j}$ and $A_{i,j}$ respectively. For a given q of index i, it is easy to see that the tuple $(x_{A_i}, x_{\mathrm{q},j}, x_{\mathbb{B}})$ corresponds to a configuration of σ_q. Hence the whole decomposition is a configuration of the composition $\sigma_{\mathrm{flow}} \odot (\sigma_1 \parallel \dots \parallel \sigma_n)$ as desired. This map defines an order-isomorphism at the level of configurations, hence induces an isomorphism of event structures. ◀

As a result, since σ_{flow} is definable (Lemma 76) and the σ_i are smaller, by induction hypothesis they are definable, hence σ is definable.