Symmetry in Concurrent Games
Simon Castellan, Pierre Clairambault, Glynn Winskel

To cite this version:
Simon Castellan, Pierre Clairambault, Glynn Winskel. Symmetry in Concurrent Games. Logic in Computer Science, Jul 2014, Vienna, Austria. 10 p. hal-00990020

HAL Id: hal-00990020
https://hal.archives-ouvertes.fr/hal-00990020
Submitted on 24 Jun 2014

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

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

Simon Castellan    Pierre Clairambault
Ecole Normale Supérieure de Lyon, France

Glynn Winskel
Computer Laboratory, University of Cambridge, UK

Abstract—Behavioural symmetry is introduced into concurrent games. It expresses when plays are essentially the same. A characterization of strategies on games with symmetry is provided. This leads to a bicategory of strategies on games with symmetry. Symmetry helps allay the perhaps overly-concrete nature of games and strategies, and shares many mathematical features with homotopy. In the presence of symmetry we can consider monads for which the monad laws do not hold on the nose but do hold up to symmetry. This broadening of the concept of monad has a dramatic effect on the types concurrent games can support and allows us, for example, to recover the replication needed to express and extend traditional game semantics.

I. INTRODUCTION

In game semantics of programming languages a type of a program is represented by a game, and the program itself by a strategy in the game. The approach is very flexible: game semantics has managed to capture accurately a wide variety of features of higher-order programming languages, including state, control, exceptions, and many others. Game semantics follows the methodology of denotational semantics, and interprets complex programs compositionally in terms of interpretations of their components. Game semantics is also operational: moves in a game correspond to computation steps. As such it is increasingly exploited to provide a syntax-free operational semantics for programming languages, a much needed tool for the analysis and verification of programs.

Concurrency is a central concern in computer science. Reflecting the traditions of concurrency theory, game semantics for concurrent programs come in two styles: based on interleavings and on partial-orders. Of these, the historical focus has been on interleaving, giving models of various programming languages [1], [2], even a model-checking tool [3]. Partial-order methods have the strength of supporting reasoning about dependency directly, but are at a more preliminary stage. Abramsky and Melliès proposed in [4] a framework for concurrent games based on closure operators, re-understood later by Melliès and Mimram in terms of asynchronous transition systems [5], [6]; Faggian and Piccolo have also presented strategies as partial orders [7]. In [8], Rideau and Winskel gave a new foundation based on event structures, generalizing all previous approaches and allowing nondeterminism. It is this framework we refer to as concurrent games.

When developing a game semantics, one has to deal with the low-level aspect of games—some identities that hold operationally do not hold automatically in games. An important example of such phenomena occurs in the replication of resources in programming languages: whereas two accesses to the same resource might be indistinguishable operationally, they can correspond to different and unrelated events in the game. In this sense, games are overly-concrete. In the history of game semantics [9], this has been alleviated by introducing symmetry into games. Informally, symmetry in a game concerns when one play of a game is essentially the same as another. Our treatment of symmetry in concurrent games, where plays can be highly-distributed, stems from earlier work on symmetry in event structures [10] and makes use of a general method of open maps for defining bisimulation in a variety of models [11]. Briefly, a symmetry in a game is expressed as a bisimulation equivalence (given as a span of open maps) that says when two plays are similar according to the symmetry. This feature considerably enhances the mathematical theory of concurrent games. Symmetry comes to share many features with homotopy—symmetric plays are like homotopic paths—which plays a role in its mathematical development.

a) Contributions. Firstly, we introduce concurrent games with symmetry. This involves a new definition of the copycat strategy and of composition, which now have to respect symmetry. As in [8], we characterize strategies, for which copycat behaves like an identity w.r.t. composition. This leads to the construction of a bicategory (up to symmetry) of concurrent games with symmetry and symmetry-respecting strategies. Secondly, we give two illustrations of how this framework can be used to model logics and programming languages. The first is a presentation within concurrent games of the construction of [12], using an adaptation of AJM games [9] to model classical linear logic (CLL). The second is a concurrent games presentation of HO games [13], giving a concurrent and non-deterministic notion of innocent strategies (in the sense of Hyland and Ong), and a new proof that standard innocent strategies are stable under composition.

b) Related work. In sequential games, the notion of symmetry that is closest to ours is that of AJM games [9], and in particular its variant in [12]. In asynchronous games, Melliès expressed symmetry by giving groups acting on the game [14], reindexing the events. Restricting to a polarized deterministic setting, it should be possible to reformulate Melliès’ approach in terms of concurrent games with symmetry—left however for future work.

c) Outline. In Section II, we present event structures, and their extension with symmetry and polarity. In Section III, we give the main contribution of this paper, the bicategory of concurrent games with symmetry and concurrent strategies. In Section IV we give two important applications: concurrent
generalizations of AJM games and HO games, showing how concurrent games with symmetry extend the games of traditional game semantics.

II. PRELIMINARIES

A. Event structures

An event structure comprises \((E, \leq, \text{Con})\), consisting of a set \(E\), of events which are partially ordered by \(\leq\), the causal dependency relation, and a nonempty consistency relation \(\text{Con}\) consisting of finite subsets of \(E\), which satisfy

\[
\begin{align*}
\{e' \mid e' \leq e\} & \text{ is finite for all } e \in E, \\
\{e\} & \in \text{Con} \text{ for all } e \in E, \\
Y \subseteq X \in \text{Con} & \implies Y \in \text{Con}, \text{ and} \\
x \in \text{Con} \& e \leq e' \in X & \implies X \cup \{e\} \in \text{Con}.
\end{align*}
\]

The (finite) configurations, \(C(E)\), of an event structure \(E\) consist of those finite subsets \(x \subseteq E\) which are

Consistent: \(x \in \text{Con}\), and

Down-closed: \(\forall e, e', e' \leq e \in x \implies e' \in x\).

We say an event structure is elementary when the consistency relation consists of all finite subsets of events. For \(X \subseteq E\) we write \([X]_E\) for \(\{e \in E \mid \exists e' \in X. e \leq e'\}\), the down-closure of \(X\); note if \(X \in \text{Con}\), then \([X]_E \in \text{Con}\) is a configuration. In games the relation of immediate dependency \(e \rightarrow e'\), meaning \(e\) and \(e'\) are distinct with \(e \leq e'\) and no event in between, will play a very important role. For configurations \(x, y\), we use \(x \rightarrow_e y\) to mean \(y\) covers \(x\), i.e. \(x \subseteq y\) with nothing in between, and \(x \rightarrow_{-e} y\) to mean \(x \cup \{e\} = y\) for an event \(e \notin x\).

We sometimes use \(x \rightarrow_{-c} y\), expressing that event \(e\) is enabled at configuration \(x\), when \(x \rightarrow_{-c} y\) for some \(y\).

A (partial) map of event structures \(f : E \rightarrow E'\) is a partial function \(f : E \rightarrow E'\) from the events of \(E\) to the events of \(E'\) such that for all configurations \(x \in C(E)\),

\[
f x \in C(E') \& (\forall e_1, e_2 \in x. f(e_1) = f(e_2) \implies e_1 = e_2).
\]

Maps compose as functions. We say a map is total when it is total as a function; then \(f\) restricts to a bijection \(x \cong f x\) on \(x \in \text{Con}(E)\). Say a total map of event structures is rigid when it preserves causal dependency. We write \(E\) (resp. \(E_p\)) for the category of event structures with total (resp. partial). These two categories have coproducts, binary products and pullbacks.

**Proposition 1.** Finite configurations of a product \(A \times B\) in \(E\) correspond to secured bijections \(\theta : x \cong y\) between configurations \(x \in C(A)\) and \(y \in C(B)\), such that the order generated on \(\theta\) by taking \((a, b) \leq (a', b')\) if \(a \leq_A a'\) or \(b \leq_B b'\) is a partial order. The correspondence respects inclusion.

Individual configurations inherit an order from the ambient event structure and can themselves be regarded as finite elementary event structures. Viewed this way, an inclusion \(x \subseteq y\) between configurations induces a rigid map \(x \rightarrow y\) between the configurations regarded as event structures. The configurations of an event structure form a non-empty family of finite partial orders closed under rigid inclusions. Conversely, given such a rigid family, we can build an event structure:

**Proposition 2.** Let \(Q\) be a rigid family, a non-empty family of finite partial orders closed under rigid inclusions, i.e. if \(q \in Q\) and \(q' \rightarrow q\) is a rigid inclusion (regarded as a map of event structures) then \(q' \in Q\). The family \(Q\) determines an event structure \(\text{Pr}(Q) := (P, \leq, \text{Con})\):

- the events \(P\) are the finite partial orders in \(Q\) with a top element (the primes),
- the causal dependency relation \(p' \leq p\) holds when there is a rigid inclusion from \(p' \rightarrow p\);
- for any finite \(X \subseteq P\), \(X \in \text{Con}\) iff there is \(q \in Q\) and rigid inclusions \(p \rightarrow q\) for all \(p \in X\).

If \(x \in \text{C}(P)\) then \(\cup x\), the union of the partial orders in \(x\), is in \(Q\). The function \(x \mapsto \cup x\) is an order-isomorphism from \(\text{C}(P)\), ordered by inclusion, to \(Q\), ordered by rigid inclusions.

B. Symmetry

We endow event structures with symmetry. A relation of symmetry on an event structure expresses when two configurations are essentially the same and is expressed as a form of bisimulation equivalence, based on open maps [11].

1) Open maps: Open maps are a generalisation of functional bisimulations, known from transition systems. Let \(C\) be a category with a distinguished subcategory \(P\) of path objects with path-extension maps. A map \(f : A \rightarrow B\) in \(C\) is open if it satisfies a path-lifting property:

\[
\begin{array}{c}
p \arrow[L] m \arrow[R] f \\
p \downarrow \downarrow \downarrow \\
q \arrow[L] \downarrow \downarrow \downarrow \\
q \downarrow \downarrow \downarrow \downarrow B.
\end{array}
\]

Any commuting square, with \(m : p \rightarrow q\) in \(P\), factors into two commuting triangles as shown. Bisimulation is then expressed as a span of open maps.

2) Event structures with symmetry: The way we equip event structures with symmetry is an instance of the following general construction. Let \(X\) be an object of a category \(C\). Recall that a relation on \(X\) is an object \(\tilde{X}\) and pair of maps \(l_X, r_X : \tilde{X} \rightarrow X\)—so forming a span—which are jointly-monadic. A map between objects with relations \(f : (X, \tilde{X}) \rightarrow (Y, \tilde{Y})\) is a map \(f : X \rightarrow Y\) in \(C\) for which there is a necessarily unique map \(\tilde{f} : \tilde{X} \rightarrow \tilde{Y}\) in \(C\) such that \(f l_X = l_Y \tilde{f}\) and \(f r_X = r_Y \tilde{f}\). If \(C\) has products, then a relation on \(X\) can equivalently be given by a mono \(\tilde{X} \rightarrow X \times X\). If \(C\) has pullbacks we can formulate diagrammatically the requirement that \(\tilde{X}\) be an equivalence relation—see [10].

If \(C\) is equipped with a class of open maps, we say that a relation \(l_X, r_X : \tilde{X} \rightarrow X\) is a symmetry in \(X\) if it is an equivalence relation with both \(l_X\) and \(r_X\) open; this amounts to \(\tilde{X}\) being a bisimulation equivalence. A map \(f : (X, \tilde{X}) \rightarrow (Y, \tilde{Y})\) necessarily preserves symmetry. We obtain a category \(\mathcal{CS}\) of objects with symmetry and symmetry-preserving maps. Given two maps \(f, g : (X, \tilde{X}) \rightarrow (Y, \tilde{Y})\) they are equal up to symmetry, written \(f \sim g\), if there is a necessarily unique map \(h : X \rightarrow \tilde{Y}\) such that \(f = l_Y h\) and \(g = r_Y h\). The category \(\mathcal{CS}\) is more fully described as enriched in the category of equivalence relations and so, because equivalence relations are a degenerate form of category, as a 2-category in
which the 2-cells are instances of the equivalence. This view
forms the constructions in such categories which are often
very simple examples of the (pseudo- and bi-) constructions
of 2-categories. In particular, objects A and B are equivalent,
written $A \cong B$, if there are maps $f : A \to B$ and $g : B \to A$
such that $f \circ g \sim \text{id}_B$ and $g \circ f \sim \text{id}_A$.

We can take advantage of the concrete nature of event
structures to give an explicit description of symmetries
there, a characterization which is independent of whether
the category of event structures carries rigid, total or partial
maps, though for definiteness assume the category is $\mathcal{E}$. A
symmetry in an event structure $E$ determines a mono
$\bar{E} \to E \times E$, thus a subset of $\mathcal{C}(E \times E)$, and so by Proposition 1 a family
of bijections between finite configurations of $E$. In this
way, a symmetry in an event structure $E$ corresponds to
an isomorphism family comprising a non-empty family of
symmetries $\theta : x \cong_E y$ between pairs of finite configurations
of $E$ such that:

(i) for all identities $\text{id}_x : x \cong_E x$, where $x \in \mathcal{C}(E)$; if $\theta : x \cong_E y$, then
the inverse $\theta^{-1} : y \cong_E x$; and if $\theta : x \cong_E y$ and $\varphi : y \cong_E z$, then
their composition $\varphi \circ \theta : x \cong_E z$.

(ii) for $\theta : x \cong_E y$ whenever $x' \subseteq x$ with $x' \in \mathcal{C}(E)$, then there
is a (necessarily unique) $y' \in \mathcal{C}(E)$ with $y' \subseteq y$ such that the
restriction $\theta' : x' \cong_E y'$.

(iii) for $\theta : x \cong_E y$ whenever $x \subseteq x'$, there is an
extension of $\theta$ to $\theta'$ so $\theta' : x' \cong_E y'$
or (not necessarily unique) $y' \in \mathcal{C}(E)$ with $y \subseteq y'$.

The isomorphism family makes precise the sense in which a
symmetry expresses when two configurations are essentially
the same. Note that (ii) implies that the bijections in the
isomorphism family respect the partial order of causal
dependency on configurations inherited from $E$; the bijections
in an isomorphism family are isomorphisms between the
configurations regarded as elementary event structures.

An event structure with symmetry $A$ corresponds to an
isomorphism family $\cong_A$ of the underlying event structure
of $A$ [10]. There are straightforward reformulations of what
means for a map to preserve symmetry or for two maps to
be equal up to symmetry in terms of isomorphism families.

A total map $f : A \to B$ preserving symmetry amounts to

$$x \cong_A y \Rightarrow f(x) \cong_B f(y),$$

where $\cong$ is the composite

$$f \circ \cong_A \circ f^{-1}.$$

while $f \sim g$, for two total maps $f, g : A \to B$ preserving
symmetry, iff $f(x) \cong_B g(x)$ for all $x \in \mathcal{C}(A)$, where $\cong$ is the composite

$$f \circ \cong_A \circ f^{-1}.$$

We define the category $\mathcal{E}S$ (resp. $\mathcal{E}S_p$) to consist of event
structures with symmetry, with total (resp. partial) maps.

**Proposition 3.** Any map $f : A \to B$ in $\mathcal{E}S_p$ has a partial-total
factorization as a composite

$$A \xrightarrow{PV} (A \downarrow V) \xrightarrow{f_1} B$$

where: $V : \text{def} \{a \in A \mid f(a) \text{ is defined}\}$ is the domain of definition
of $f$; $A \downarrow V : \text{def} \{v \in V \mid v \subseteq v' \text{ iff } v \subseteq_A v' \}$

$v' \in V$ and $X \in \text{Con}_X$ iff $X \in \text{Con}_X$ & $X \subseteq V$; its
isomorphism family is given by $\theta : x \cong_A y$ iff $\theta \text{ extends to } \theta' : [x]_A \cong_A [y]_A$; the map $p_V : A \to A \uparrow V$ is the partial
map acting as identity on $V$ and undefined elsewhere; and $f_1$
is a total map acting as $f$ on $V$. If $f \sim g : A \to B$ in $\mathcal{E}S_p$, then the
domains of definition of $f$ and $g$ are the same, $V$ say,
and $f_1 \sim g_1$ in their partial-total factorizations $f = f_1p_V$ and
$g = g_1p_V$.

Through the addition of symmetry event structures can
represent a much richer class of categories than mere partial
orders. The finite configurations of an event structure with
symmetry can be extended by inclusion or rearranged bijectively
under an isomorphism family allowed by the symmetry. In
this way an event structure with symmetry determines, in
general, a category of finite configurations with maps those
inclusions obtained by repeatedly composing the inclusions
and allowed isomorphisms. By property (ii) in the definition
of isomorphism family any such map factors uniquely as an
isomorphism of the symmetry followed by an inclusion.

3) **Constructions:** We review from [10] important
constructions in $\mathcal{E}S$.

**a) Products:** First, $\mathcal{E}S$ has binary products.

**Theorem 4.** Let $A$ and $B$ be objects in $\mathcal{E}S$. Their product
in $\mathcal{E}S$ is given by $(A \times_B; \lambda_A \times_{B}, \rho_A \times_{B}, \lambda_B \times_{B}) : \mathcal{A} \times \mathcal{B} \to \mathcal{A} \times \mathcal{B}$, based
on the product $A \times B$ in $\mathcal{E}$, and sharing the same projections,$\pi_1 : A \times B \to A$ and $\pi_2 : A \times B \to B$. Let $f, g : C \to A$ and
$g', f' : C \to B$. If $f \sim f'$ and $g \sim g'$, then $(f, g) \sim (f', g')$.

**b) Simple parallel composition:** The operation of simple
parallel composition of event structures $A || B$ which juxtaposes
the two event structures $A$ and $B$—with a finite set of events
consistent if its overlaps with $A$ and $B$ are consistent—and
acts similarly on maps, extends to a functor on $\mathcal{E}S$. The
symmetry on $A || B$ is $\bar{A} || \bar{B}$ with left and right maps $\lambda_A || \lambda_B$ and
$\rho_A || \rho_B$.

**c) Pseudo pullbacks:** It will have great bearing on this
paper that, while $\mathcal{E}S$ does not have equalizers or pullbacks in
general, it does have pseudo equalizers and pseudo pullbacks.
We give their definition and refer the reader to [10] for proofs.

Maps $f, g : B \to C$ have a pseudo equalizer $i : E \to B$, i.e. $f \sim g$ and for any map $i' : E' \to B$ such that $f_{i'} \sim g_{i'}$, there
is a unique map $h : E' \to E$ such that $i' = ih$. The object
$E$ can be described as having configurations those $x \in \mathcal{C}(B)$
for which $x \cong_C \exists x'$, where $\exists x'$ is the bijection induced by $x$:
its isomorphism family is the restriction of that on $B$.

Maps $f : A \to C$, $g : B \to C$ have a pseudo pullback
given as the pseudo equalizer of $f_{\pi_1} \times g_{\pi_2} : A \times B \to C$. We
summarize its properties: The pseudo pullback comprises
an object $D$ and maps $\pi_1 : D \to A$ and $\pi_2 : D \to B$ such that
$f_{\pi_1} \sim g_{\pi_2}$ which satisfies the further property that for any
object $D'$ and maps $p_1 : D' \to A$ and $p_2 : D' \to B$ such that
$f_{p_1} \sim g_{p_2}$, there is a unique map $h : D' \to D$ such that
$p_1 = h_{\pi_1} p_2$. The pseudo pullback is defined up to
isomorphism, and sometimes written $A \times_C B$.

Concretely, a configuration in $\mathcal{C}(A \times_C B)$ corresponds to a
where for \( \theta \), \( \varphi \), \( \tilde{\theta} \) symmetry, the symmetry \( \tilde{\theta} \) strategies. The Scott order \( \sim \) isomorphism family of \( \left( A, \tilde{A} \right) \), as an object in \( \varepsilon A \), stands for (a generalization of) the unit interval: asserting \( \theta \) \( (a \text{ generalization of}) \) the isomorphism family of \( A \), is analogous to specifying a \( \tilde{\theta} \) \( \varepsilon A \) itself equipped with symmetry \( \tilde{\theta} \). Concretely, a configuration of \( \tilde{\theta} \) corresponds to an element of the isomorphism family \( \theta : x \varepsilon A y \) and a configuration of \( \tilde{\theta} \) to a pair \( \varphi_1, \varphi_2 \) in a commuting square mediating between \( \theta_1 \) and \( \theta_2 \) in the isomorphism family of \( A \):

\[
\begin{array}{c}
\varphi_1 \equiv_A \varphi_2 \\
\varphi_1 \equiv_A y_1 \\
x_1 \theta_1 \equiv_A \varphi_1 \\
\varphi_2 \equiv_A y_2 \\
x_2 \theta_2 \equiv_A \tilde{\theta}
\end{array}
\]

Lemma 5. Let \( f : A \to B \) and \( g : B \to C \) be maps in \( \varepsilon A \) with pseudo pullback \( P \) with maps \( \pi_1 : P \to A \) and \( \pi_2 : P \to B \). Then, \( P \), as an object in \( \varepsilon A \), with maps \( \tilde{\pi}_1 \) and \( \tilde{\pi}_2 \) is a pseudo pullback of \( \tilde{f} \) and \( \tilde{g} \) in \( \varepsilon A \).

C. Adding polarity

Games and strategies will be represented in terms of event structures where events are moves of Player or Opponent, signified by events carrying a polarity, + for Player and − for Opponent. Formally, an event structure with polarity comprises an event structure \( A \) with a function \( \text{pol} \) : \( A \to \{+,-\} \) assigning a polarity to each event. Maps are assumed to preserve polarity.

The addition of polarity to an event structure \( A \) means that the \( \varepsilon \)-order on its finite configurations is now obtained as compositions \( (\leq \vee \leq^+) \) of two more fundamental orders, where for \( x, y \in \varepsilon A \),

\[
\begin{align*}
x \leq y & \iff x \leq y \land \text{pol}(y \setminus x) \leq \{-\}, \text{ and} \\
x \leq^+ y & \iff x \leq y \land \text{pol}(y \setminus x) \leq \{+\}.
\end{align*}
\]

More surprisingly there is a new partial order, the Scott order, between configurations which is intimately related to copycat strategies. The Scott order \( \varepsilon A \) is defined to be \( (\leq^- \vee \leq^+) \). (We use \( \leq^- \) for the converse order to \( \leq^- \)). The Scott order possesses a unique factorization: for \( x, y \in \varepsilon A \),

\[
y \varepsilon A x \iff \exists z \in \varepsilon A . y \leq^- z \leq^+ x
\]

— it is an easy exercise to show that \( z \) is necessarily \( x \cap y \). Not only do the configurations of copycat strategies of \( A \) correspond to pairs of configurations in the Scott order, but also strategies in \( A \) correspond to certain, simply-described, discrete fibrations over \( (\varepsilon A, \leq A) \)—see [15] for the full story. Given this we can expect a variation on the Scott order to play a role in strategies in games with symmetry.

An event structure with polarity and symmetry (henceforth an e.p.s.) consists of \( (A, E) \), an event structure with polarity also endowed with a symmetry \( E \). The categories \( \varepsilon A \) and \( \varepsilon A_p \) of such objects have maps preserving both symmetry and polarity, which are respectively total and partial.

The addition of polarity and symmetry brings a new richness to the configurations of an event structure. The Scott order becomes a Scott category, \( \varepsilon A \), of an e.p.s. \( A \), where now maps between configurations are obtained as compositions of (partial injections) \( \leq^- \), \( \leq^+ \) and the isomorphism family \( \varepsilon A \). Maps from \( y \) to \( x \) in \( \varepsilon A \) have a unique factorization

\[
y \leq^- y_0 \varepsilon A x_0 \leq^+ x.
\]

The unique factorization follows from property (ii) of isomorphism families and uniqueness of the factorization of the Scott order. A map \( f : A \to B \) in \( \varepsilon A \) preserves \( \leq^- \), \( \leq^+ \) and isomorphism families so extends directly to a functor \( f : \varepsilon A \to \varepsilon B \) making \( \varepsilon A \) a functor to \( \text{Cat} \).

III. GAMES WITH SYMMETRY

A concurrent game with symmetry is represented by an e.p.s. \( A \). A pre-strategy in \( A \) is a total map \( \sigma : S \to A \) in \( \varepsilon A \). A map between pre-strategies, from \( \sigma : S \to A \) to \( \sigma' : S' \to A \), is a map \( f : S \to S' \) in \( \varepsilon A \) such that

\[
S \xrightarrow{\sigma} A \xleftarrow{\sigma'} S'.
\]

commutes. We say the two pre-strategies \( \sigma \) and \( \sigma' \) are equivalent, and write \( \sigma \equiv \sigma' \), and sometimes \( f : \sigma \equiv \sigma' \), when there are maps \( f \) from \( \sigma \) to \( \sigma' \), and \( g \) from \( \sigma' \) to \( \sigma \) determining an equivalence \( S \equiv S' \), i.e. such that \( gf \equiv id_S \) and \( fg \equiv id_S' \); isomorphism \( \sigma \equiv \sigma' \) occurs when \( gf \equiv id_S \) and \( fg \equiv id_S' \). A weak map from \( \sigma \) to \( \sigma' \) is a map \( f : S \to S' \) such that the triangle above commutes up to \( \sim \) and analogously say \( \sigma \) and \( \sigma' \) are weakly equivalent when there are weak maps \( f \) and \( g \) making \( S \equiv S' \).

Simple parallel composition extends directly to a functor \( A \| B \) on \( A \) and \( B \) in \( \varepsilon A \). The dual of an e.p.s. \( A \), written \( A^d \), has the same underlying event structure with symmetry but with a reversal of polarities.

Following Joyal [16], a pre-strategy from \( A \) to \( B \) is a pre-strategy in the game \( A^d \| B \). In refining the notion of pre-strategy to that of strategy we shall follow the guiding
A. Copycat

Let $A$ be an e.p.s. Configurations of $C_A$ will correspond to maps in $\mathcal{S}(A)$. Recall a map from $y$ to $x$ in $\mathcal{S}(A)$ can be put into a unique form

$$y \sqsupset y_0 \triangleq x_0 \sqsubseteq x.$$ Define $q(x, \theta, y)$ to be the partial order with underlying set $x/y \equiv \{1\} \times x \cup \{2\} \times y$, causal dependency that inherited from $A|A$ with additional causal dependencies

$$\{(a_1, a_2) \in x_0 \times y_0 \mid pol_A(a_2) = + \& \theta(a_1) = a_2\} \cup \{(a_2, a_1) \in y_0 \times x_0 \mid pol_A(a_1) = + \& \theta(a_1) = a_2\}.$$ That $q(x, \theta, y)$ is indeed a partial order follows as in [8].

Define $\mathbf{C}_{C_A}$ to be a functor, the operation $\mathbf{C}_{C_A}$ is functorial in $A$. We define the symmetry on $\mathbf{C}_{C_A}$ to be $Pr(\mathcal{Q})$. Because $\mathbf{S}(A)$ is a functor, the operation $\mathbf{C}_{C_A}$ is functorial in $A$. We define the symmetry on $\mathbf{C}_{C_A}$ as

$$\mathbf{C}_{\mathbf{C}_{C_A}} \dashv \mathbf{C}_{\mathbf{C}_{C_A}}$$

Note, that in the construction of $\mathbf{C}_{C_A}$ we are using the fact that $\mathcal{A}$ possesses a symmetry $\mathcal{A}$—see Section II-B4. The map $\gamma_A : \mathbf{C}_{C_A} \to A|A$ takes a prime order to its top event.

Lemma 6. The construction $(\mathbf{C}_{C_A}, \mathbf{C}_{r_{C_A}}, \mathbf{C}_{r_{C_A}} : \mathbf{C}_{C_A} \to \mathbf{C}_{C_A})$ is an e.p.s. and $\gamma_A$ a pre-strategy.

In future we shall overload $\mathbf{C}_{C_A}$ and write $\mathbf{C}_{C_A}$ for the e.p.s. of $\mathbf{C}_{C_A}$ by Lemma 6. From the definitions, $\mathbf{C}_{C_A} = \mathbf{C}_{\mathbf{C}_{C_A}}$.

B. Composition

Let $\sigma : S \to A|B$ and $\tau : T \to B|C$ be pre-strategies between games $A, B, C$. To define their composition first form the pseudo pullback

$$\pi_1 S \times_A A \quad \pi_2$$

of the maps on the underlying event structures with symmetry, ignoring polarities, viz. $\sigma : S \to A|B$ and $\tau : T \to B|C$. There is an obvious partial map of event structures $A|B|C \to A|C$ undefined on $B$ and acting as identity on $A$ and $C$. The partial maps from $P$ to $A|C$, given by following the diagram either way round the pseudo pullback,

$$\pi_1 S \parallel C \quad \pi_2 A \parallel T$$

are defined on a common subset $V \subseteq P$. Forming the partial-total factorization of either map (for preciseness take the left/lowest) we obtain

$$P \parallel V \to A|C.$$ The resulting total map gives us the composition

$$\tau \circ \sigma : T \circ S \equiv \text{def} P \parallel V \to A|C$$

once we reinstate polarities to make $\tau \circ \sigma$ a map in $\mathcal{EPS}$. The projection operation $P \parallel V$ hides all events not visible in $A$ and $C$.

From Lemma 5, regarding $e.g. \mathcal{T}$ as itself an e.p.s., it follows that $\mathcal{T} \circ S \equiv \mathcal{T} \circ S$ and $\tau \circ \sigma \equiv \tau \circ \sigma$.

C. Strategies

We are interested in necessary and sufficient conditions on a pre-strategy $\sigma : S \to A$ to ensure $\sigma \equiv \gamma_A \circ \sigma$. As we shall see $\sigma$ should be equivalent to a pre-strategy which is strong-receptive, innocent and saturated. (If a pre-strategy $\sigma : S \to A|B$ satisfies these conditions, then $\sigma \equiv \gamma_B \circ \sigma \circ \gamma_A$.)  

1) Necessity: We show that for any pre-strategy $\sigma : S \to A$ the pre-strategy $\gamma_A \circ \sigma$ is necessarily strong-receptive, innocent and saturated.

A pre-strategy $\sigma$ is receptive iff for all $x \in \mathcal{C}(S)$, $\sigma x - \circ \& \gamma_A(a) = -$ implies $\exists! s \in S. x - \circ \& \sigma(s) = a$. The pre-strategy $\sigma : S \to A$, as a map in $\mathcal{EPS}$ preserves symmetry, so is associated with a map $\sigma : S \to A$. Say $\sigma$ is strong-receptive if $\sigma$, and so also $\sigma$, is receptive. A pre-strategy $\sigma$ is innocent when $s \sim s'$ and $(pol(s) = +$ or $(pol(s') = -$ implies $\sigma(s) \sim \sigma(s')$.

To specify when a pre-strategy $\sigma : S \to A$ is saturated we need some background. Form the pseudo pullback

$$\pi_1 S \times_A A \quad \pi_2$$

The operation taking $\sigma$ to its saturation $\pi_2$ is part of a monad on pre-strategies in $A$. Clearly

$$\pi_2 A \parallel T$$

commutes. Hence there is a unique map $\eta : S \to S \times_A A$ such that $\pi_1 \eta = \text{id}_S$ and $\pi_2 \eta = \sigma$—thus $\eta$ is a map from $\sigma$ to $\pi_2$. We say $\sigma$ is saturated when $\eta$ is part of an equivalence $\sigma \equiv \pi_2$, i.e. there is a map $\text{act} : S \times_A A \to S$ such that

$$S \times_A A \quad \text{act}$$

is a pre-strategy.
commutes and \( \text{act } \eta \sim \text{id}_S \) with \( \eta \text{ act } \sim \text{id}_{S \times A A} \). Concretely, a finite configuration of \( S \times A A \) can be identified with a pair \((x, \theta)\) where \( x \in C(S) \) and \( \theta : \sigma \vec{x} \cong y \). The action of \( \text{act} \) is to transport the configuration \( x \) across \( \theta \) to it a configuration \( x' = \text{def } \text{act}(x, \theta) \) with \( x \vec{x} \cong x' \vec{x} \) and \( \vec{\phi} = \theta \).

Certain compositions are automatically saturated:

**Proposition 7.** The composition \( \tau \circ \sigma \) of pre-strategies \( \tau : \varnothing \rightarrow B \) and \( \tau : B \rightarrow C \) is saturated.

**Proof.** Adopt the notation of the diagram defining the composition \( \tau \circ \sigma \) in Section III-B, with \( A = \varnothing \). Let \( u \in C(T \circ S) \). Its down-closure \([u]\) is a configuration in the pseudo pullback \( P \), and so corresponds to a secured bijection \( x \vec{x} \cong y \) where \( x \in C(S) \), \( v \in C(C) \) and \( y \in C(T) \). Because we define \( \tau \circ \sigma \) to be got via the left way round the pseudo pullback square the configuration \( u \) is sent to \( v \) via \( \tau \circ \sigma \). Consequently, given \( \theta : v \cong C \) \( v' \) we can define \( \text{act}(u, \theta) \) to be that configuration \( x \vec{x} \cong x \vec{x} \) with down-closure \( x \vec{x} \cong x \vec{x} \) in the pseudo pullback. \( \square \)

**Lemma 8.** \( \gamma_A \) is strong-receptive, innocent and saturated.

**Proof.** The construction of \( C A \) directly ensures the innocence and receptivity of \( \gamma_A \). The way symmetry of copycat is obtained from \( C \) makes \( \gamma_A \) equal \( \gamma_A \) so receptive, guaranteeing strong-receptivity of \( \gamma_A \). To see \( \gamma_A \) is saturated we require a map \( \text{act} : C A \times A A A \rightarrow C A \). A configuration of \( C A \times A A A \) corresponds to a configuration of \( C A \), so a map in \( \text{Scott}(A) \)

\[
x \vec{x} \cong x_0 \vec{x}_0 ^{\theta} y_0 \leq y,
\]

and a configuration of \( A \rightarrow C A \), for a pair \( v, w \), for which \( \varphi_1 : x \cong A v \) and \( \varphi_2 : y \cong A w \). This data and the factorization properties in \( \text{Scott}(A) \) yield:

\[
\begin{align*}
x & \vec{x} \cong x_0 & \vec{x}_0 ^{\theta} y_0 \leq y \\
& \varphi_1 = A \vec{\varphi}_1 ^{\theta} & \vec{\varphi}_2
\end{align*}
\]

We take the configuration of \( C A \) got via \( \text{act} \) to be that corresponding to

\[
v \vec{x} \cong v_0 \vec{x}_0 ^{\theta} w_0 \leq w.
\]

The map \( \text{act} \) together with \( \eta : C A \rightarrow C A \times A A A \rightarrow C A \) establishes the equivalence needed for \( \gamma_A \) to be saturated. \( \square \)

**Lemma 9 (Necessity).** \( \gamma_A \circ \sigma \) is strong-receptive, innocent and saturated for any pre-strategy \( \sigma \) in \( A \).

**Proof.** The composition \( \gamma_A \circ \sigma \) inherits innocence and receptivity directly from that of \( \gamma_A \). Now \( \gamma_A \circ \sigma \cong \gamma_A \circ \sigma \) whence \( \gamma_A \circ \sigma \) inherits receptivity from that of \( \gamma_A \), making \( \gamma_A \circ \sigma \) strong-receptive. The composition is saturated for general reasons—Proposition 7.

2) **Sufficiency:** We show the conditions strong-receptive, innocent and saturated are sufficient to ensure that a pre-strategy \( \sigma \) is equivalent to its composition with copycat \( \gamma_A \circ \sigma \).

**Lemma 10.** Let \( \sigma : S \rightarrow A \) be a pre-strategy. There is a map \( I : S \rightarrow C C A \circ S \in \text{EPS} \), unique up to symmetry, such that \( \sigma = (\gamma_A \circ \sigma) I \).

**Proof.** We sketch the existence part of the proof by describing how \( I \) acts on configurations. Given \( x \in C(S) \), there is a secured bijection \( x \vec{x} \cong q(\sigma x, \text{id}_x, \sigma x) \)—the bijection is that given by \( x \vec{x} \cong \sigma x \) between left components and \( \text{id}_x \) between the right. The secured bijection corresponds to a configuration \( z \) of the pseudo pullback \( (S \parallel A) \times A A A C A \). The result of the projection operation is to hide all those events not above the right component in \( A \), so from \( z \) yields a configuration of \( C C A \circ S \) with image \( \sigma x \) under \( \gamma_A \circ \sigma \). \( \square \)

**Lemma 11.** If a pre-strategy \( \sigma \) is strong-receptive, innocent and saturated, then \( I : \sigma \equiv \gamma_A \circ \sigma \) is an equivalence.

**Proof.** We require a map \( K : C C A \circ S \rightarrow S \) that with \( I \) establishes an equivalence \( \gamma_A \circ \sigma \equiv \sigma \). Let \( u \in C(C C A \circ S) \). Its down-closure \([u]\) is a configuration in the pseudo pullback \( (S \parallel A) \times A A A C A \):

\[
\begin{align*}
S \parallel A & \rightarrow (S \parallel A) \times A A A C A \rightarrow C C A \\
\sigma & \rightarrow \gamma_A \rightarrow \gamma_A \circ \sigma \\
A & \rightarrow A A A \rightarrow C C A
\end{align*}
\]

As a configuration of the pseudo pullback, \([u]\) corresponds to a triple, a configuration in \( S \parallel A \) and a configuration in \( C C A \), mediated by an element of the isomorphism family of \( A \parallel A \)—see Section II-B3c. A configuration of \( S \parallel A \) corresponds to a pair of configurations \( x \) of \( S \) and \( w \) of \( A \); a configuration of \( C C A \) to configurations of \( A \) in the relations \( z_1 \vec{z}_1 \cong z_2 \vec{z}_2 \); and the mediating element of the isomorphism family to a pair \( \sigma x \cong z_1 \) and \( w \cong z_2 \). From this data we obtain the composite map

\[
\sigma x \cong z_1 \vec{z}_1 \cong z_2 \vec{z}_2 \cong z_2 \cong A w
\]

in \( \text{Scott}(A) \). This factors uniquely into

\[
\sigma x \vec{x} \cong y_1 \vec{y}_1 \cong y_2 \vec{y}_2 \cong w.
\]

From innocence it follows (see Lemma 1 of [15]) that there is a unique \( x_1 \in C(S) \) for which \( x \vec{x} \cong x_1 \) and \( \sigma x_1 = y_1 \). Now, the pair \( x_1 \) and \( \theta : y_1 \cong y_2 \) can be identified with a configuration of \( S \times A A \). Hence we can apply \( \text{act} \) to obtain \( \text{act}(x_1, \theta) \in C(S) \) with \( \sigma \text{act}(x_1, \theta) = y_2 \).

Define \( p(u) = \text{def } \text{act}(x_1, \theta) \). By considering how it acts on isomorphism families, \( p \) extends to a monotonic function

\[
\vec{p} : C(C C A \circ S) \rightarrow C(S)
\]

such that

\[
\forall u \in C(C C A \circ S), \vec{p}(u) \leq \gamma_A \circ \sigma u.
\]
Using Lemma 23 of [8] and the strong-receptivity of \( \sigma \), we obtain a unique total map \( K : C_S \circ S \to S \) such that \( \forall u \in C(C_A \circ S) \), \( p(u) \subseteq K u \) and \( \gamma_A \circ \sigma = \sigma K \). On checking \( K I \sim id_S \) and \( K K \sim id_{C_A \circ S} \) we have the desired equivalence. 

Define a strategy to be a pre-strategy which is strong-receptive, innocent and saturated.

A strategy \( \sigma : S \to A \) induces a fibration \( \sigma : Scott(S) \to Scott(A) \). In fact, \( \supseteq \) and \( \subseteq \) maps in \( Scott(A) \) have cartesian liftings again as \( \supseteq \) and \( \subseteq \) maps, respectively, in \( Scott(S) \) because \( \sigma \) is receptive and innocent [15]—with strong-receptivity ensuring the appropriate uniqueness—while \( \supseteq_A \) maps have cartesian liftings in \( \supseteq_S \) because \( \sigma \) is saturated.

**Lemma 12.** If \( \sigma \) and \( \tau \) are strategies, so is \( \tau \circ \sigma \).

**Proof.** The composition inherits innocence and receptivity from \( \sigma \) and \( \tau \). Because \( \tau \circ \sigma \equiv \tau \circ \sigma \) it also inherits strong receptivity. Its saturation obtains via Proposition 7 and the saturation of \( \tau \).

3) A \( \sim \)-bicategory of games with symmetry: Combining the above results, we do not quite obtain a bicategory but rather Strat, a “bicategory up to symmetry” in the following sense:

- its objects are e.p.s.’s—the games;
- its arrows from \( A \) to \( B \) are strategies \( \sigma : A \rightsquigarrow B \) related by maps of pre-strategies—\( Strat(A, B) \) is thus a category enriched with equivalence relations \( \sim \);
- horizontal composition is given by composition of strategies \( \sigma \circ \tau \), which extends to functors \( Strat(B, C) \times Strat(A, B) \to Strat(A, C) \) via the universality of pseudo pullback;
- there is a natural isomorphism (derived from the universality of pseudo pullback) to express the associativity of composition, but only natural equivalences (derived from the equivalence of Lemma 11) for left and right identity laws;
- of the usual coherence axioms for bicategories, that for identity only commutes up to \( \sim \).

Because categories of event structures with symmetry are degenerate 2-categories, the above describes a special case of weak 3-category, which we call a \( \sim \)-bicategory as it is morally a bicategory but where the axioms hold up to \( \sim \).

Strat is rich in structure. Observe the duality: a strategy \( \sigma : A \rightsquigarrow B \) corresponds to a strategy \( \sigma^\perp : B^\perp \rightsquigarrow A^\perp \). There is a bijective correspondence between strategies \( A \parallel B \rightsquigarrow C \) and strategies \( A \rightsquigarrow (B^\perp \parallel C) \), making Strat monoidal closed, and in fact compact closed, in an extended sense.

To be more precise, and to relate to standard game semantics, we can quotient out the higher-dimensional structure to obtain a category. The category Strat\( /\perp \) has e.p.s.’s as objects, and \( \perp \)-equivalence classes of strategies \( \sigma : S \to A^\perp \parallel B \) as morphisms from \( A \) to \( B \). The \( \sim \)-bicategorical structure ensures that equivalence of strategies is preserved by composition, so we get a category Strat\( /\perp \).

**Proposition 13.** The category Strat\( /\perp \) is compact closed, with tensor product \( \parallel \) and dual \( (-)^\perp \).

The compact closed structure of Strat\( /\perp \) is not so surprising: the category is defined in a similar fashion to Joyal’s category of Conway games [16], which is compact closed as well. As compact closed categories, they are *-autonomous and hence models of Multiplicative Linear Logic (MLL) [17].

4) Weak strategies: A weak strategy is a pre-strategy which is weakly equivalent to \( \gamma_A \circ \sigma \); so, directly from this definition, any weak strategy is weakly equivalent to a strategy. (We do not have a direct characterization of weak strategies.)

As we shall see, it is sometimes convenient to work with weak strategies (which need not be saturated) and then compose with copycat to obtain the strategies they represent.

Another potential advantage of weak strategies is that they are closed under a more general composition than that of strategies. We can build a \( \sim \)-bicategory of weak strategies—biequivalent to the \( \sim \)-bicategory of strategies—in which instead composition is based on a choice of bipullbacks rather than pseudo-pullbacks. (See [10] for the definition of bipullbacks of event structures with symmetry.) This extra latitude in the choice of definition of composition is likely to have technical advantages when working with sub \( \sim \)-bicategories of games, for which the saturation of strategies seems unnecessary or unnatural.

**Lemma 14.** If two saturated pre-strategies are weakly equivalent they are equivalent. A fortiori, if two strategies are weakly equivalent they are equivalent.

IV. APPLICATIONS

Once we have symmetry in games we can support a rich repertoire of (pseudo) monads on e.p.s.’s, and e.g. all the monads of [10] are undisturbed by the presence of polarity. Monads to support copying w.r.t. maps of e.p.s.’s can often translate to monads w.r.t. strategies. Following Girard’s work on linear logic [18], this opens up the possibility of modelling programming languages that are not resource-sensitive, in that (copies of) the same resource can be used multiple times.

We describe, in particular, how AJM games [9] and HO games [13] generalize and can be recovered from concurrent games. The (co)monads involved rely pivotally on the presence of symmetry. Their structure lifts from simple structural maps.

A. Maps as strategies

A structural pair \( f = (f^L, f^R) : A \to B \) comprises

- \( f^L : A \to B \), a total map of e.p.s.’s, as left component, and
- \( f^R : B \to A \), an injective, partial map of event structures, not necessarily preserving symmetry, as right component, such that \( f^R \circ f^L = id_A \). (Such pairs correspond to Kahn and Plotkin’s rigid embeddings if we ignore symmetry.)

A structural pair \( f : A \to B \), lifts to a strategy \( \overline{f} : S(f) \to A^\perp \parallel B \), obtained via a rigid family \( Q \). Whenever

\[
x \supseteq^+ x' \overset{f^L}{\mapsto} f^L x' \overset{\theta}{\mapsto} y', y' \subseteq y,
\]

with \( x, x' \in C(A) \) and \( y, y' \in C(B) \), define a typical \( q(x, \theta, y) \in Q \) to have underlying set \( x \parallel y \), and causal dependency that
Logic (MELL) is a construction in [12] of a non-polarized adaptation of AJM involutive negation and are restricted to modelling “polarized” of Classical Linear Logic. In contrast, polarized games lose reversal, and hence model directly the involutive negation because they permit an account of negation as just polarity-initial moves share the same polarity, a condition imposed in lifts as strategies.

Additionally, one can check that saturated, so a strategy. Note that this lifting operation also inherited from \( S\frac{f}{g} \). From this lemma it follows easily that lifting is functorial.

We also need to examine the composition of a strategy with the structural pair. Then the pre-strategy \( \sigma \) is equivalent to the strategy \( \frac{f}{g} \sigma \) can be restricted to make \( \frac{f}{g} \) a total map preserving symmetry. Write \( \frac{f}{g} \sigma \) isomorphism family. Then \( \sigma \) isomorphism family that part preserved by \( \frac{f}{g} \). Addition-

From a game \( A \), we form the game \( !A \) comprising \( \omega \) similar copies of \( A \). Its events are pairs \( (i, a) \) where \( a \in A \) and \( i \in \omega \), with causal dependency

\[
(i_1, a_1) \leq (i_2, a_2) \iff i_1 = i_2 \& a_1 \leq_A a_2
\]

and consistency relation

\[
\text{Con}_A = \bigcup_{i \in I} \{ i \} \times X_i,
\]

where \( I \) is a finite subset of \( \omega \), and for each \( i \in I \), \( X_i \in \text{Con}_A \). Polarity is inherited from \( A \). We describe its symmetry as an isomorphism family. If \( x, y \in C(!A) \), we have \( x \cong_A y \) if, writing \( x = \bigcup_{i \in I} \{ i \} \times X_i \) and \( y = \bigcup_{j \in J} \{ j \} \times Y_j \) with each \( x_i \) and \( y_j \) nonempty, there is a bijection \( \pi : I \cong J \) so \( x_i \cong_A y_{\pi(i)} \) for each \( i \in I \), where \( \theta((i, a)) = (\pi(i), \theta(a)) \) for all \( (i, a) \in x \).

The construction extends to a functor on \( \text{EPS} \), if \( f : A \to B \) then \( f : !A \to !B \) sends \((i, a)\) to \((i, f(a))\). It is convenient to investigate the monad/monoid structure of \(!\text{EPS}\) first, their duals will be eventually deduced by duality in \( \text{Strat} \). In \( \text{EPS} \), we have:

\[
\eta_A : A \to !A \quad e_A : 1 \to !A \\
\mu_A : !!A \to !A \quad m_A : !A \| !A \to !A \\
q_{A,B} : !!(A \| B) \to !A \| !B
\]

where \( \eta_A \) sends any event \( a \) to \((0, a)\); \( \mu_A \) tracks an arbitrary bijection between \( \omega \times \omega \) and \( \omega \); \( q_{A,B} \) is the obvious distribution map; \( e_A \) is the empty map; and \( m_A \) tracks an arbitrary bijection between \( \omega + \omega \) and \( \omega \). All these maps are natural in their parameters in the category \( \text{EPS} \). In particular, \((!, \eta, \mu)\) forms a monad on \( \text{EPS} \).

The functors \( \| \) and \( ! \) on \( \text{EPS} \) extend to functors on strategies written \( \| S \) and \( ! S \) for disambiguation: given strategies \( \sigma_1 : S_1 \to A_1^2 \| B_1 \) and \( \sigma_2 : S_2 \to A_2^2 \| B_2 \), we have \( \| S \sigma_1 : S_1 \| S_2 \to (A_1 \| A_2) \| (B_1 \| B_2) \) given by the obvious map of \( \text{EPS} \). To define \( ! S \sigma \), the obvious choice is the composition \( q_{A_1^2, B_1} \circ \sigma : !S \to !A_1^2 \| !B_1 \), which we must then compose with copycat to obtain a strategy. These yield a bifunctor \( ! S \) and a functor \( \| S \) on strategies, using the fact that \( ! \) and \( \| \) preserve pseudo-pullbacks.

The natural transformations above all have adjoints together with which they form structural pairs. By the techniques of Section IV-A, lifting maps to strategies, they become (apart from \( e_A \)) natural as families of strategies:
Lemma 17. The strategies \( \overline{\epsilon}_A, \delta_A, \overline{d}_A \) and \( \overline{q}_{A,B} \) are natural in \( A, B \) in the category \( \text{Strat}/\simeq \),

Proof. Each of these maps \( f \) has an adjoint \( f^* \) making \( (f, f^*) \) a structural pair. By Lemmas 15 and 16 the compositions involved in the naturality squares can be computed by simple composition of maps, and the equivalences then amount to elementary verifications.

The monoid and monad laws follow from those in \( \mathcal{EPS} \) by functoriality of lifting. The fact that algebra morphisms between free algebras \( (\mathcal{A}, \mu_A) \) satisfy the laws needed to be morphisms of monoids follows from a simple diagram chase, using that \( \mu_A \) is an invertible monoid morphism and that this property remains true of \( \overline{\mu}_A \).

We have established that the natural transformations above define the dual of a linear exponential comonad, short of the naturality of \( e_A \). By self-duality of \( \text{Strat}/\simeq \), we have:

**Theorem 18.** The category \( \text{Strat}/\simeq \) with \( ! \) is a model of CLL in the sense of [12].

C. HO games and HO-innocence

Finally, we sketch an application of our setting of objects to construct a notion of concurrent games with pointers, obtaining a concurrent generalization of HO games. An e.p.s., \( A \) is negative when all its minimal events have negative polarity.

1) Concurrent games with pointers: An arena is a countable forest \( (A, \leq_A, \text{pol}_A) \) with polarities (but without conflict or symmetry), which is also negative, and alternating in the sense that if \( a_1 \to a_2 \) then \( \text{pol}_A(a_1) \neq \text{pol}_A(a_2) \).

From an arena \( A \), we now define a game \( ?A \) understood as “\( A \) with pointers”. Its definition requires the notion of an exponential slice: a slice for an event \( a \) is a function \( \alpha : [a] \to \omega \), giving a copy index to each event on which \( a \) depends. For two slices \( \alpha : [a] \to \omega \) and \( \alpha' : [a'] \to \omega \), we write \( \alpha \leq \alpha' \) when \( a \leq a' \) and \( \alpha(b) = \alpha'(b) \) for all \( b \leq a \). The game \( ?A \) has:

- events, pairs \( (a, o) \) where \( a : [a] \to \omega \),
- causal dependency, \( (\alpha, a) \leq (\alpha', a') \) iff \( \alpha \leq \alpha' \),
- consistency, all finite subsets of \( ?A \),
- polarity, inherited from \( A \), and
- isomorphism family, order-isos \( \theta : x \equiv y \) such that for all \( (a, o) \in x \) there is \( \alpha' : [a] \to \omega \) with \( \theta(\alpha, a) = (\alpha', a) \).

To convey the similarity with the plays with pointers of HO games, we could say that \( (\alpha, a) \) “points to” \( (\alpha', a') \) when \( a \leq a' \) and \( a' \to a \). However \( ?A \) leaves duplicated events causally unrelated whereas they would appear in some chronological order in a play with pointers. Configurations of \( ?A \) are more closely related to the thick subtrees of Boudes [19].

Games of the form \( ?A \) are comonoids in \( \text{Strat}/\simeq \). Indeed, we have maps of event structures \( m_A : ?A^1 \parallel \cdasharrow ?A^1 \parallel \cdasharrow ?A \) and \( e_A : 1 \to ?A^1 \) satisfying monoid laws up to symmetry, and the lifting operation yields strategies \( \overline{m}_A : ?A \rightarrow ?A \parallel \cdasharrow ?A \) and \( \overline{e}_A : 1 \cdasharrow ?A \) in \( \text{Strat}/\simeq \), satisfying comonoid laws. We will now describe a subcategory of \( \text{Strat}/\simeq \) whose morphisms preserve this comonoid structure.

1) A cartesian closed category: A strategy \( \sigma : S \to ?A^1 \parallel ?B \) is single-threaded iff (1) for all \( s \in S \), \([s]\) has exactly one minimal event, which is negative, and (2) there is a map \( \langle -,- \rangle : S \parallel ?S \to S \) such that \( \sigma \circ \langle -,- \rangle = (m_A^1 \parallel m_B) \circ (\sigma \parallel S) \), and such that for all \( x_1, x_2 \in C(S) \) disjoint and compatible, the composite bijection \( x_1 \cup x_2 \cong x_1 \parallel x_2 \cong \langle x_1, x_2 \rangle \) is in the isomorphism family of \( S \). Single-threaded strategies are stable under composition, and lifting yields single-threaded strategies (so copycat is single-threaded).

Condition (1) expresses that causality is local to threads (i.e., configurations with a unique minimal event), it will be used to decompose configurations of \( S \) into threads. On the other hand, condition (2) allows us to merge a set of threads back into a configuration; (1) and (2) together allow us to prove the following key lemma.

**Lemma 19.** Single-threaded strategies are comonoid morphisms for the comonoids \( (?A, m_A^1 \parallel e_A^1) \) in \( \text{Strat}/\simeq \).

Since we want a category of comonoids and comonoid morphisms, we will now restrict our attention to single-threaded strategies between games of the form \( ?A \). An HO-strategy from arena \( A \) to arena \( B \) is a single-threaded strategy \( \sigma : S \to \cdasharrow ?A^1 \parallel ?B \). There is a category \( \text{HOStrat} \) having arenas as objects, and HO-strategies as morphisms.

 Arenas are closed under \( \parallel \), and there are obvious isomorphisms \( \parallel (A \parallel B) \cong \parallel A \parallel B \). Using these, \( \text{HOStrat} \) inherits from \( \text{Strat}/\simeq \) its symmetric monoidal structure and terminal object. Moreover, arenas also inherit a comonoid structure that HO-strategies preserve by Lemma 19, hence (see e.g. Corollary 17 in [17]) \( \text{HOStrat} \) is cartesian.

We also need \( \text{HOStrat} \) to be monoidal closed. For that we introduce the usual arrow arena construction [13], that (following [20]) we denote by \( A \dashv B \):

- events, \( \langle \text{b} \parallel \text{min}(B) \rangle \parallel B \) where \( \text{min}(B) \) denotes the set of minimal events of \( B \), and
- causality, that of \( \langle \text{b} \parallel \text{min}(B) \rangle \parallel A^1 \parallel B \) enriched with \( (\text{b}, \text{min}(B)) \) \( a \in A \) \& \( b \in \text{min}(B) \).
- polarity, inherited in \( B \) and reversed in \( A \).

HO-strategies from \( A \) to \( B \) can be represented as HO-strategies on \( A \dashv \parallel B \) through the following proposition.

**Proposition 20.** For any arenas \( A, B, C \) there is a bijection (up to equivalence) \( \Phi_{A,B,C} \) between single-threaded \( \sigma : C \dashv \cdasharrow \cdasharrow ?A^1 \parallel ?B \) and single-threaded \( \sigma' : C \dashv \cdasharrow \cdasharrow (A \dashv \parallel B) \).

Moreover \( \Phi \) preserves composition, in the sense that for all \( \tau : D \dashv \cdasharrow \cdasharrow C \) we have \( \Phi(\sigma \circ \tau) = \Phi(\sigma) \circ \tau \).

Using this, the category \( \text{HOStrat} \) inherits the closed structure of \( \text{Strat}/\simeq \). Since its monoidal product is actually cartesian, it is cartesian closed, hence a model of the simply-typed \( \lambda \)-calculus. It has in fact much more structure and we hope to be able to recast and generalize within it various games models of programming languages. In this paper though, we will only show that it contains (a nondeterministic generalization of) the usual HO category of innocent strategies.

3) Sequential HO-innocence: An HO-strategy \( \sigma : S \to \cdasharrow ?A \) is sequential HO-innocent (or an HOIS-strategy) if for any
by distinct positive events \(s_1, s_2\), then \([s] \cup \{s_1, s_2\} \notin \text{Con}_S\).

Intuitively, a prime \([s]\) corresponds to a P-view: just like a P-view, a prime configuration of an HOIS-strategy cannot contain two negative events \(s_1, s_2\) whose mapping to \(?A\) “point” to the same event \((\alpha, a)\): by innocence \(s_1\) and \(s_2\) would have to be concurrent in \([s]\), which is impossible by condition (1). So similarly to standard HO innocence the condition (1) expresses that the strategy \(\sigma\) is blind to Opponent reopenings: every Player event contains in its causal history at most one Opponent event pointing to a specific Player event.

**Proposition 21.** Arenas and sequential HO-innocent strategies form a subCCC, \(\text{HOISS}\text{trat} \) of \(\text{HOStrat}\).

Note in passing that condition (1) is stable under composition on its own, but is too restrictive to be a satisfactory notion of concurrent HO-innocence as it forbids natural concurrent strategies such as that for the parallel or. We leave for future work the design and study of a more satisfactory notion of concurrent HO-innocence. We now go on and isolate a deterministic subcategory of \(\text{HOISS}\text{trat}\), isomorphic to the usual category of arenas and innocent strategies [13].

An HOIS-strategy \(\sigma : S \to \?A\) is deterministic iff whenever \(x \not\in S\) and \(y \not\in S\), then \(x' \not\in S\), and for some \(x''\) and \(y''\) in \(\mathcal{C}(S)\) with \(\varphi' : x'' \not\in S\) \(y''\) such that \(\varphi \in \varphi'\) —this is a generalization of the notion of deterministic concurrent strategies [21] in the presence of symmetry.

**Theorem 22.** The subcategory of \(\text{HOISS}\text{trat}\) having arenas as objects and deterministic sequential HO-innocent strategies as morphisms is isomorphic to the usual category of arenas and innocent strategies.

**Sketch.** Let \(\sigma : S \to \?A\) be a deterministic HOIS-strategy. For \(s \in S\) positive, the prime configuration \([s]\) needs to be a chain

\[
s_0 \rightarrow s_1 \rightarrow s_2 \rightarrow \ldots \rightarrow s^*,
\]

by condition (1) of HOIS-strategies along with innocence and the alternation and negativity conditions of arenas. This induces a sequence of moves of \(A\):

\[
\pi_2 \sigma(s_0) \pi_2 \sigma(s_1) \ldots \pi_2 \sigma(s)
\]

In turn, immediate dependency in \(A\) equips this sequence with pointers such that (by innocence) Opponent always points to the previous move. In other words, this is a P-view.

Applying this to all primes in \(S\) we get a set of P-views that is O-branching by receptivity, condition (2) of HOIS-strategies, the extension operation on \(S\) and determinism; so a deterministic innocent strategy in the sense of HO games.

This operation has an inverse up to equivalence of strategies and preserves composition, yielding an isomorphism of categories with standard innocent strategies.

**V. CONCLUSION**

Concurrent games with symmetry have proved versatile enough to accommodate and extend in a single framework two radically different fundamental games models: the saturated AJM games model of CLL and the HO innocent games model of the simply-typed \(\lambda\)-calculus, with a generalization to nondeterminism—in the past, providing a notion of non-deterministic HO innocence has proved challenging [22]. The framework is grounded in the mathematically-achievable setting of event structures with symmetry, with potentially fruitful connections to homotopy.

In future, we intend to exploit the framework introduced here to develop concurrent-games models for various programming languages. Its versatility and its proximity to traditional game semantics suggests that it is adequate to give precise partial-order semantics to complex concurrent programming language, including such features as higher-order procedures and shared memory—features which, to our knowledge, have only been modelled through interleaving.

**Acknowledgements** Thanks to Martin Hyland for advice and encouragement and to the ERC for Advanced Grant ECSYM.

**REFERENCES**


