Strategies with Parallel Causes

Marc de Visme
Ecole Normale Supérieure de Paris, France

Glynn Winskel
Computer Laboratory, University of Cambridge, UK

***Abstract*—In a distributed game we imagine a team Player engaging a team Opponent in a distributed fashion. Such games and their strategies have been formalised within event structures. However there are limitations in founding strategies on traditional event structures. Sometimes a probabilistic distributed strategy relies on certain benign races where, intuitively, several members of team Player may race each other to make a common move. Although there are event structures which support such parallel causes, in which an event is enabled in several compatible ways, they do not support an operation of hiding central to the composition of strategies; nor do they support probability adequately. An extension of traditional event structures is devised which supports parallel causes and hiding, as well as the mix of probability and nondeterminism needed to account for probabilistic distributed strategies. The extension is located within existing models for concurrency and tested in the construction of a bicategory of probabilistic distributed strategies with parallel causes. The bicategory is rich in operations relevant to probabilistic as well as deterministic parallel programming.**

I. Prolegomena

This article addresses a fundamental, potentially widespread issue of which few are aware. It concerns the accurate modelling of parallel causes in probabilistic distributed strategies; we are thinking for instance of a strategy in which it is advantageous to allow two or more members of the same team to race each other cooperatively, without conflict, to perform some common move. It fixes the absence of a computational model which simultaneously handles parallel causes, probability and an operation of hiding internal events; it provides such a model, locates it via adjunctions within existing models and tests it in the construction of a bicategory of probabilistic distributed strategies supporting parallel causes.

Roughly, there are three reasons why such a model has not been invented previously:

"no causality:" many models, based on "interleaving" approaches to concurrency, do not consider causes explicitly so are blind to the issue of parallel causes;

"no probability:" when not considering probability, in the presence of nondeterminism, parallel causes can be fudged and treated as conflicting disjunctive causes;

"no hiding:" for example, in the graphical models of statistics, which do involve parallel causes, one generally avoids hiding features in order to ensure conditional independencies.

But there are good reasons to want to combine probability and hiding with parallel causes. Causality is essential in the local analysis of systems and as we shall see some optimal strategies are impossible to define without parallel causes. Hiding is a fundamental mechanism in abstraction, while dealing with probabilities is a fact of life in computing today. A great many aspects of computation can be described within games. It is not surprising that without all three features we cannot give a fully adequate compositional account of all the distributed strategies we may wish to consider.

Our development starts with event structures. The choice is not arbitrary. Event structures occupy a central position in models for concurrent computation, both "interleaving" and "causal" [17], and can claim to be the concurrent or causal analogue of trees; just as a transition system unfolds to a tree so a Petri net unfolds to an event structure. It is only by having developed games within event structures that we came to realise the limitations of existing models. The original point in developing "concurrent games" in event structures was to recast game semantics within a broader framework of concurrent/interactive/distributive computation where it more properly belongs, and can more rightly claim to be the wide semantic foundation it aspires to.

For these reasons, while phrased in the language of games based on event structures, the issues addressed, those of modelling a distributed system while simultaneously coping with causality, probability and hiding, are of broad relevance.

II. Introduction

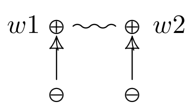
We consider probabilistic distributed games between two teams, Player and Opponent. To set the scene, imagine a simple distributed game in which team Opponent can perform two moves, called 1 and 2, far apart from each other, and that team Player can just make one move, 3. Suppose that for Player to win they must make their move iff Opponent makes one or more of their moves. Informally Player can win by assigning two members of their team, one to watch out for the Opponent move 1 and the other Opponent move 2. When either watcher sees their respective Opponent move they run back and make the Player move 3. Opponent could possibly play both 1 and 2 in which case both watchers would run back and could make their move together. Provided the watchers are perfectly reliable this provides a winning strategy for Player. No matter how Opponent chooses to play or not play their moves, Player will win; if Opponent is completely inactive the watchers wait forever but then Player does win, eventually.

We can imagine variations in which the watchers are only reliable with certain probabilities with a consequent reduction in the probability of Player winning against Opponent strategies. In such a probabilistic strategy Player can only determine probabilities of their moves conditionally on those of Opponent. Because Player has no say in the probabilities of Opponent moves beyond those determined by causal dependencies of the strategy we are led to a *Limited Markov Condition*, of the kind discussed in [8]:

> **(LMC)** In a situation x in which both a Player move $\oplus$ and an Opponent move $\ominus$ could occur individually, if the Player move and the Opponent move are causally independent, then they are probabilistically independent; in a strategy for Player, $\mathrm{Prob}(\oplus \mid x, \ominus) = \mathrm{Prob}(\oplus \mid x)$.

The LMC is borne out in the game of "matching pennies" where Player and Opponent in isolation, so independent from each other, each make their choice of head or tails. Note we do not expect that in all strategies for Player that two causally independent Player moves are necessarily probabilistically independent; in fact, looking ahead, because composition of strategies involves hiding internal moves such a property would not generally be preserved by composition.

Let us try to describe the informal strategy above in terms of event structures. In 'prime' event structures in which causally dependency is expressed a partial order, an event is causally dependent on a unique set of events, *viz.* those events below it in the partial order. For this reason within prime event structures we are forced to split the Player move into two events one for each watcher making the move, one $w1$ dependent on Opponent move 1 and the other $w2$ on Opponent move 2. The two moves of the two watchers stand for the same move in the game. Because of this they are in conflict (or inconsistent) with each other. We end up with the event structure drawn below:



The polarities + and − signify moves of Player and Opponent, respectively. The arrows represent the (immediate) causal dependencies and the wiggly line conflict. As far as purely nondeterministic behaviour goes, we have expressed the informal strategy reasonably well: no matter how Opponent makes or doesn't make their moves any maximal play of Player is assured to win. However consider assigning conditional probabilities to the watcher moves. Suppose the probability of $w1$ conditional on 1 is p_1, *i.e.* $\mathrm{Prob}(w1 \mid 1) = \mathrm{Prob}(w1, 1 \mid 1) = p_1$ and that similarly for $w1$ its conditional probability $\mathrm{Prob}(w2 \mid 2) = p_2$. Given that move $w1$ of Player and move 2 of Opponent are causally independent, from (LMC) we expect that $w1$ is probabilistically independent of move 2, *i.e.*

$$\mathrm{Prob}(w1 \mid 1, 2) = \mathrm{Prob}(w1 \mid 1) = p_1 ;$$

whether Opponent chooses to make move 2 or not should have no influence on the watcher of move 1. Similarly,

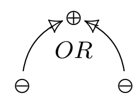
$$\mathrm{Prob}(w2 \mid 1, 2) = \mathrm{Prob}(w2 \mid 2) = p_2 .$$

But $w1$ and $w2$ are in conflict, so mutually exclusive, and can each occur individually when 1 and 2 have occurred ensuring that

$$p_1 + p_2 \leq 1$$

—we haven't insisted on one or the other occurring, the reason why we have not written equality. The best Player can do is assign $p_1 = p_2 = 1/2$. Against a counter-strategy with Opponent playing one of their two moves with probability 1/2 this strategy only wins half the time. We have clearly failed to express the informal winning strategy accurately!

Present notions of "concurrent strategies," the most general of which are presented in [14], are or can be expressed using prime event structures. If we are to be able to express the intuitive strategy which wins with certainty we need to develop distributed probabilistic strategies which allow *parallel causes* in which an event can be enabled in distinct but compatible ways. 'General' event structures are one such model [13]. In the informal strategy described in the previous section both Opponent moves would individually enable the Player move, with all events being consistent, illustrated below:



But as we shall see general event structures do not support an appropriate operation of hiding central to the composition of strategies. Nor is it clear how within general event structures one could express the variant of the strategy above in which the two watchers succeed in reporting with different probabilities.

It has been necessary to develop a new model—*event structures with disjunctive causes* (edc's)—which support hiding and probability adequately, and into which both prime and general event structures embed. Conceptually, one is forced to objectify cause in a way that is reminiscent of formal proof being an objectification of theoremhood. Formally, this is achieved by extending prime event structures with an equivalence relation; the equivalence classes are thought of as 'disjunctive events' of which the representatives are 'prime causes.' In this way causes may conflict or not, possess probabilities, and be correlated or independent. The new model provides a foundation on which to build a theory and rich language of probabilistic distributed strategies with parallel causes. Even without probability, it provides a new bicategory of *deterministic* parallel strategies which includes, for example, a deterministic strategy for computing "parallel or"—Section VIII-C.

Full proofs can be found in [16]. Appendix A summarises the simple instances of concepts we borrow from enriched categories [5] and 2-categories [9].

III. Event structures

Event structures describe a process, or system, in terms of its possible event occurrences, their causal dependencies and consistency. The simplest form, 'prime' event structures, are a concurrent, or distributed, analogue of trees; though in such an event structure the individual 'branches' are no longer necessarily sequences but have the shape of a partial order of events.

A. Prime event structures

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

$[e] =_{\mathrm{def}} \{e' \mid e' \leq e\}$ is finite for all $e \in E$,
$\{e\} \in \mathrm{Con}$ for all $e \in E$,
$Y \subseteq X \in \mathrm{Con}$ implies $Y \in \mathrm{Con}$, and
$X \in \mathrm{Con}$ & $e \leq e' \in X$ implies $X \cup \{e\} \in \mathrm{Con}$.

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

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

A configuration inherits a partial order from the ambient event structure, and represents a possible partial-order history.

Two events e, e' are considered to be causally independent, and called *concurrent* if the set $\{e, e'\}$ is in Con and neither event is causally dependent on the other. The relation of *immediate* dependency $e \rightarrowtriangle e'$ means e and e' are distinct with $e \leq e'$ and no event in between. Write $\mathcal{C}^\infty(E)$ for the configurations of E and $\mathcal{C}(E)$ for its finite configurations. For configurations x, y, we use $x \mathrel{-\!\!\subset} y$ to mean y covers x, *i.e.* $x \subset y$ with nothing in between, and $x \xrightarrow{e}\!\!\subset y$ to mean $x \cup \{e\} = y$ for an event $e \notin x$. We sometimes use $x \xrightarrow{e}\!\!\subset$, expressing that event e is enabled at configuration x, when $x \xrightarrow{e}\!\!\subset y$ for some y.

It will be very useful to relate event structures by maps. A *map* of event structures $f : E \to E'$ is a partial function f from E to E' such that the image of a configuration x is a configuration fx and any event of fx arises as the image of a unique event of x. Maps compose as partial functions. Write $\mathcal{E}$ for the ensuing category.

A map $f : E \to E'$ reflects causal dependency locally, in the sense that if e, e' are events in a configuration x of E for which $f(e') \leq f(e)$ in E', then $e' \leq e$ also in E; the event structure E inherits causal dependencies from the event structure E' via the map f. Consequently, a map preserves concurrency: if two events are concurrent, then their images if defined are also concurrent. In general a map of event structures need not preserve causal dependency; when it does and is total we say it is *rigid*.

B. General event structures

A *general event structure* [11], [13] comprises $(E, \mathrm{Con}, \vdash)$ where E is a set of event occurrences, the consistency relation Con is a non-empty collection of finite subsets of E satisfying

$$X \subseteq Y \in \mathrm{Con} \implies X \in \mathrm{Con}$$

and the *enabling relation* $\vdash \subseteq \mathrm{Con} \times E$ satisfies

$$Y \in \mathrm{Con} \;\&\; Y \supseteq X \;\&\; X \vdash e \implies Y \vdash e\,.$$

A *configuration* is a subset of E which is

consistent: $X \subseteq_{\mathrm{fin}} x \implies X \in \mathrm{Con}$ and
secured: $\forall e \in x \exists e_1, \cdots, e_n \in x.\;\; e_n = e \;\;\&$
$\forall i \leq n. \{e_1, \cdots, e_{i-1}\} \vdash \; e_i\,.$

Again we write $\mathcal{C}^\infty(E)$ for the configurations of E and $\mathcal{C}(E)$ for its finite configurations.

The notion of secured has been expressed through the existence of a securing chain to express an enabling of an event within a set which is a complete enabling in the sense that everything in the securing chain is itself enabled by earlier members of the chain. One can imagine more refined ways in which to express complete enablings which are rather like proofs. Later the idea that complete enablings are consistent partial orders of events in which all events are enabled by earlier events in the order—"causal realisations"—will play an important role in generalising general event structures to structures supporting hiding and parallel causes.

A *map* $f : (E, \mathrm{Con}, \vdash) \to (E', \mathrm{Con}', \vdash')$ of general event structures is a partial function $f : E \rightharpoonup E'$ such that

$X \in \mathrm{Con} \implies fX \in \mathrm{Con}'$ &
$\forall e_1, e_2 \in X.\; f(e_1) = f(e_2) \implies e_1 = e_2$ and
$X \vdash e$ & $f(e)$ is defined $\implies fX \vdash' f(e)\,.$

Maps compose as partial functions with identity maps being identity functions. Write $\mathcal{G}$ for the category of general event structures.

We can characterise those families of configurations arising from a general event structure. A *family of configurations* which comprises a family $\mathcal{F}$ of sets such that

if $X \subseteq \mathcal{F}$ is finitely compatible in $\mathcal{F}$ then $\bigcup X \in F$; and
if $e \in x \in \mathcal{F}$ then there exists a securing chain $e_1, \cdots, e_n = e$ in x such that $\{e_1, \cdots, e_i\} \in \mathcal{F}$ for all $i \leq n$.

The latter condition is equivalent to saying (i) that whenever $e \in x \in \mathcal{F}$ there is a finite $x_0 \in \mathcal{F}$ such that $e \in x_0 \in \mathcal{F}$ and (ii) that if $e, e' \in x$ and $e \neq e'$ then there is $y \in \mathcal{F}$ with $y \subseteq x$ s.t. $e \in y \iff e' \neq y$. The elements of the underlying set $\bigcup \mathcal{F}$ are its *events*. Such a family is *stable* when for any compatible non-empty subset X of $\mathcal{F}$ its intersection $\bigcap X$ is a member of $\mathcal{F}$. We shall use $x \uparrow y$ or $X \uparrow$ to signify configurations x and y or a subset of configurations X are compatible.

A configuration $x \in \mathcal{F}$ is *irreducible* iff there is a necessarily unique $e \in x$ such that $\forall y \in \mathcal{F}.\; e \in y \subseteq x$ implies $y = x$. Irreducibles coincide with complete join irreducibles w.r.t. the order of inclusion. It is tempting to think of irreducibles as representing minimal complete enablings. But, as sets, irreducibles both (1) lack sufficient structure: in the formulation we are led to of minimal complete enabling as prime causal realisations, several prime realisations can have the same irreducible as their underlying set; and (2) are not general enough: there are prime realisations whose underlying set is not an irreducible.

A map between families of configurations from $\mathcal{F}$ to $\mathcal{G}$ is a partial function $f : \bigcup\mathcal{F} \rightharpoonup \bigcup\mathcal{G}$ between their events such that for any $x \in \mathcal{F}$ its image $fx \in \mathcal{G}$ and

$$\forall e_1, e_2 \in x.\ f(e_1) = f(e_2) \implies e_1 = e_2 \,.$$

Maps between general event structures satisfy this property. Maps of families compose as partial functions.

The forgetful functor taking a general event structure to its family of configurations has a left adjoint, which constructs a canonical general event structure from a family: given $\mathcal{A}$, a family of configurations with underlying events A, construct a general event structure $(A, \mathrm{Con}, \vdash)$ with

$X \in \mathrm{Con}$ iff $X \subseteq_{\mathrm{fin}} y$, for some $y \in \mathcal{A}$, and
$X \vdash a$ iff $a \in A$, $X \in \mathrm{Con}$ & $e \in y \subseteq X \cup \{a\}$, for some $y \in \mathcal{A}$.

The above yields a coreflection of families of configurations in general event structures. It cuts down to an equivalence between families of configurations and *replete* general event structures. A general event structure $(E, \mathrm{Con}, \vdash)$ is *replete* iff

$$\forall e \in E \exists X \in \mathrm{Con}.\ X \vdash e \,,$$
$$\forall X \in \mathrm{Con} \exists x \in \mathcal{C}(E).\ X \subseteq x \text{ and}$$
$$X \vdash e \implies \exists x \in \mathcal{C}(E).\ e \in x \ \&\ x \subseteq X \cup \{e\} \,.$$

The last condition is equivalent to stipulating that each minimal enabling $X \vdash e$, where X is a minimal consistent set enabling e, corresponds to an irreducible configuration $X \cup \{e\}$.

C. On relating prime and general event structures

Clearly a prime event structure $(P, \leq, \mathrm{Con})$ can be identified with a (replete) general event structure $(P, \vdash, \mathrm{Con})$ by taking

$$X \vdash p \text{ iff } X \in \mathrm{Con} \ \&\ [p] \subseteq X \cup \{p\} \,.$$

Indeed under this identification there is a full and faithful embedding of $\mathcal{E}$ in $\mathcal{G}$. However (contrary to the claim in [13]) there is no adjoint to this embedding. This leaves open the issue of providing a canonical way to describe a general event structure as a prime event structure. This issue has arisen as a central problem in reversible computation [3] and now more recently in the present limitation of concurrent strategies described in the introduction. A corollary of our work will be that the embedding of prime into general event structures does have a *pseudo* right adjoint, got at the slight cost of enriching prime event structures with equivalence relations.

IV. Problems with general event structures

Why not settle for general event structures as a foundation for distributed strategies? Because although they allow parallel causes, they don't support hiding so composition of strategies; nor do they support probability generally enough.

A. Probability and parallel causes

We return to the general-event-structure description of the strategy in the Introduction. To turn this into a probabilistic strategy for Player we should assign probabilities to configurations conditional on Opponent moves. The watcher of 1 is causally independent of Opponent move 2. Given this we might expect that the probability of the watcher of 1 making the Player move 3 should be probabilistically independent of move 2; after all, both moves 3 and 2 can occur concurrently from configuration $\{1\}$. Applying LMC naively would yield

$$\mathrm{Prob}(1, 3 \mid 1) = \mathrm{Prob}(1, 2, 3 \mid 1, 2) \,.$$

But similarly, $\mathrm{Prob}(2, 3 \mid 2) = \mathrm{Prob}(1, 2, 3 \mid 1, 2)$, which forces $\mathrm{Prob}(1, 3 \mid 1) = \mathrm{Prob}(2, 3 \mid 2)$, *i.e.* that the conditional probabilities of the two watchers succeeding are the same! In blurring the distinct ways in which move 3 can be caused we have obscured causal independence which has led us to identify possibly distinct probabilities.

B. Hiding

With one exception, all the operations used in building strategies and, in particular, the bicategory of concurrent strategies [10] extend to general event structures. The one exception, that of hiding, is crucial in ensuring composition of strategies yields a bicategory.

Consider a general event structure with *events* a, b, c, d and e; *enabling* (1) $b, c \vdash e$ and (2) $d \vdash e$, with all events other than e being enabled by the empty set; and *consistency* in which all subsets are consistent unless they contain the events a and b —the events a and b are in conflict.

Any configuration will satisfy the assertion

$$(a \wedge e) \implies d$$

because if e has occurred it has to have been enabled by (1) or (2) and if a has occurred its conflict with b has prevented the enabling (1), so e can only have occurred via enabling (2).

Now imagine the event b is hidden, so allowed to occur invisibly in the background. The configurations after hiding are those obtained by hiding (*i.e.* removing) the invisible event b from the configurations of the original event structure. The assertion above will still hold of the configurations after hiding. There isn't a general event structure with events a, c, d and e, and configurations those which result when we hide (remove) b from the configurations of the original event structure. One way to see this is to observe that amongst the configurations after hiding we have

$$\{c\} {-\!\!\subset} \{c, e\} \text{ and } \{c\} {-\!\!\subset} \{a, c\}$$

where both $\{c, e\}$ and $\{a, c\}$ have upper bound $\{a, c, d, e\}$, and yet $\{a, c, e\}$ is not a configuration after hiding as it fails to satisfy the assertion. (In configurations of a general event structure if $x {-\!\!\subset} y$ and $x {-\!\!\subset} z$ and y and z are bounded above, then $y \cup z$ is a configuration.) Precisely the same problem can arise in the composition (with hiding) of strategies based on general event structures (unless they are deterministic [16]).

To obtain a bicategory of strategies with disjunctive causes we need to support hiding. We need to look for structures more general than general event structures. The example above gives a clue: the inconsistency should be one of inconsistency between (minimal complete) enablings rather than events.

V. Adding disjunctive causes

To cope with disjunctive causes and hiding we must go beyond general event structures. We introduce structures in which we *objectify* cause; a minimal complete enabling is no longer an instance of a relation but a structure that realises that instance (*cf.* a judgement of theorem-hood in contrast to a proof). This is in order to express inconsistency between minimal complete enablings, inexpressible as inconsistencies on events, that can arise when hiding.

Fortunately we can do this while staying close to prime event structures. The twist is to regard "disjunctive events" as comprising subsets of events of a prime event structure, the events of which are now to be thought of as representing "prime causes" standing for minimal complete enablings. Technically, we do this by extending prime event structures with an equivalence relation on events.

In detail, an *event structure with equivalence* (an ese) is a structure

$$(P, \leq, \mathrm{Con}, \equiv)$$

where $(P, \leq, \mathrm{Con})$ satisfies the axioms of a (prime) event structure and $\equiv$ is an equivalence relation on P.

An ese dissociates the two roles of enabling and atomic action conflated in the events of a prime event structures. The intention is that the events p of P, or really their corresponding down-closures $[p]$, describe minimal complete enablings, *prime causes*, while the $\equiv$-equivalence classes of P represent *disjunctive events*: p is a prime cause of the disjunctive event $\{p\}_{\equiv}$. Notice there may be several prime causes of the same event and that these may be *parallel causes* in the sense that they are consistent with each other and not related in the order $\leq$.

A *configuration* of the ese is a configuration of $(P, \leq, \mathrm{Con})$ and we shall use the notation of earlier on event structures $\mathcal{C}^\infty(P)$ and $\mathcal{C}(P)$ for its configurations, respectively finite configurations. We say a configuration is *unambiguous* if it has no two distinct elements which are $\equiv$-equivalent, We modify the relation of concurrency a little and say $p_1, p_2 \in P$ are *concurrent* and write $p_1 co\, p_2$ iff $[p_1] \cup [p_2]$ is an *unambiguous* configuration of P and neither $p_1 \leq p_2$ nor $p_2 \leq p_1$.

When the equivalence relation $\equiv$ of an ese is the identity we essentially have a prime event structure. This view is reinforced in our choice of maps. A map from ese $(P, \equiv_P)$ to $(Q, \equiv_Q)$ is a partial function $f : P \rightharpoonup Q$ which *preserves* $\equiv$, *i.e.* if $p_1 \equiv_P p_2$ then either both $f(p_1)$ and $f(p_2)$ are undefined or both defined with $f(p_1) \equiv_Q f(p_2)$), such that for all $x \in \mathcal{C}(P)$

(i) the direct image $fx \in \mathcal{C}(Q)$, and

(ii) $\forall p_1, p_2 \in x.\ f(p_1) \equiv_Q f(p_2) \implies p_1 \equiv_P p_2$.

Maps compose as partial functions with the usual identities.

Such maps preserve the concurrency relation. They are only assured to reflect causal dependency locally w.r.t. unambiguous configurations.

We regard two maps $f_1, f_2 : P \to Q$ as equivalent, and write $f_1 \equiv f_2$, iff they are equi-defined and yield equivalent results, *i.e.*

if $f_1(p)$ is defined then so is $f_2(p)$ and $f_1(p) \equiv_Q f_2(p)$, and

if $f_2(p)$ is defined then so is $f_1(p)$ and $f_1(p) \equiv_Q f_2(p)$.

Composition respects $\equiv$: if $f_1, f_2 : P \to Q$ with $f_1 \equiv f_2$ and $g_1, g_2 : Q \to R$ with $g_1 \equiv g_2$, then $g_1 f_1 \equiv g_2 f_2$. Write $\mathcal{E}_\equiv$ for the category of ese's; it is *enriched* in the category of sets with equivalence relations—see Appendix A.

Ese's support a hiding operation. Let $(P, \leq, \mathrm{Con}_P, \equiv)$ be an ese. Let $V \subseteq P$ be a $\equiv$-closed subset of 'visible' events. Define the *projection* of P on V, to be $P{\downarrow}V =_{\mathrm{def}} (V, \leq_V, \mathrm{Con}_V, \equiv_V)$, where $v \leq_V v'$ iff $v \leq v'$ & $v, v' \in V$ and $X \in \mathrm{Con}_V$ iff $X \in \mathrm{Con}$ & $X \subseteq V$ and $v \equiv_V v'$ iff $v \equiv v'$ & $v, v' \in V$.

Hiding is associated with a factorisation of partial maps. Let

$$f : (P, \leq_P, \mathrm{Con}_P, \equiv_P) \to (Q, \leq_Q, \mathrm{Con}_Q, \equiv_Q)$$

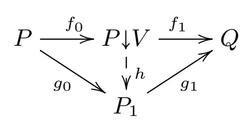
be a partial map between two ese's. Let

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

Then f factors into the composition

$$P \xrightarrow{f_0} P{\downarrow}V \xrightarrow{f_1} Q$$

of f_0, a partial map of ese's taking $p \in P$ to itself if $p \in V$ and undefined otherwise, and f_1, a total map of ese's acting like f on V. We call f_1 the *defined part* of the partial map f. Because $\equiv$-equivalent maps share the same domain of definition, $\equiv$-equivalent maps will determine the same projection and $\equiv$-equivalent defined parts. The factorisation is characterised to within isomorphism by the following universal characterisation: for any factorisation $P \xrightarrow{g_0} P_1 \xrightarrow{g_1} Q$ where g_0 is partial and g_1 is total there is a (necessarily total) unique map $h : P{\downarrow}V \to P_1$ such that



commutes.

The category $\mathcal{E}_\equiv$ of ese's supports hiding in the sense above. We next show how replete general event structures embed in ese's.

VI. A pseudo adjunction

The (pseudo) functor from $\mathcal{G}$ to $\mathcal{E}_\equiv$ is quite subtle but arises as a right adjoint to a more obvious functor from $\mathcal{E}_\equiv$ to $\mathcal{G}$.

Given an ese $(P, \leq, \mathrm{Con}, \equiv)$ we can construct a (replete) general event structure $ges(P) =_{\mathrm{def}} (E, \mathrm{Con}_E, \vdash)$ by taking

$E = P_\equiv$, the equivalence classes under $\equiv$;

$X \in \mathrm{Con}_E$ iff $\exists Y \in \mathrm{Con}.\ X = Y_\equiv$; and

$X \vdash e$ iff $X \in \mathrm{Con}$ & $e \in E$ &

$$\exists p \in P.\ e = \{p\}_\equiv \ \&\ [p]_\equiv \subseteq X \cup \{e\}.$$

The construction extends to a functor $ges : \mathcal{E}_\equiv \to \mathcal{G}$ as maps between ese's preserve $\equiv$; the functor takes a map $f : P \to Q$ of ese's to the map $ges(f) : ges(P) \to ges(Q)$ obtained as the partial function induced on equivalence classes. Less obvious is that there is a (pseudo) right adjoint to ges. Its construction relies on extremal causal realisations which provide us with an appropriate notion of minimal complete enabling of events in a general event structure.

A. *Causal realisations*

Let $\mathcal{A}$ be a family of configurations with underlying set A. A *(causal) realisation* of $\mathcal{A}$ comprises a partial order

$$(E, \leq),$$

its *carrier*, such that the set $\{e' \in E \mid e' \leq e\}$ is finite for all events $e \in E$, together with a function $\rho : E \to A$ for which the image $\rho x \in \mathcal{A}$ when x is a down-closed subset of E.

A map between realisations $(E, \leq), \rho$ and $(E', \leq'), \rho'$ is a partial surjective function $f : E \rightharpoonup E'$ which preserves down-closed subsets and satisfies $\rho(e) = \rho'(f(e))$ when $f(e)$ is defined. It is convenient to write such a map as $\rho \geq^f \rho'$. Occasionally we shall write $\rho \geq \rho'$, or the converse $\rho' \leq \rho$, to mean there is a map of realisations from ρ to ρ'.

Such a map factors into a "projection" followed by a total map

$$\rho \geq_1^{f_1} \rho_0 \geq_2^{f_2} \rho'$$

where ρ_0 stands for the realisation $(E_0, \leq_0), \rho_0$ where

$$E_0 = \{r \in R \mid f(r) \text{ is defined}\},$$

the domain of definition of f, with $\leq_0$ the restriction of $\leq$, and f_1 is the inverse relation to the inclusion $E_0 \subseteq E$, and f_2 is the total function $f_2 : E_0 \to E'$. We are using $\geq_1$ and $\geq_2$ to signify the two kinds of maps. Notice that $\geq_1$-maps are reverse inclusions. Notice too that $\geq_2$-maps are exactly the total maps of realisations. Total maps $\rho \geq_2^f \rho'$ are precisely those functions f from the carrier of ρ to the carrier of ρ' which preserve down-closed subsets and satisfy $\rho = \rho' f$.

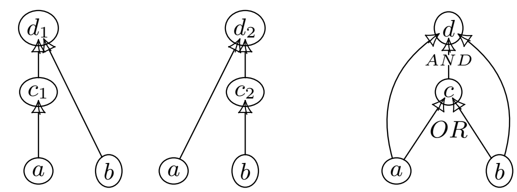
We shall say a realisation ρ is *extremal* when $\rho \geq_2^f \rho'$ implies f is an isomorphism, for any realisation ρ'.

In the special case where $\mathcal{A}$ is the family of configurations of a prime event structure, it is easy to show that an extremal realisation ρ forms a bijection with a configuration of the event structure and that the order on the carrier coincides with causal dependency there.

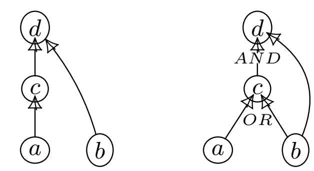
The construction is more interesting when $\mathcal{A}$ is the family of configurations of a general event structure. In general, there is at most one map between extremal realisations. Hence extremal realisations of A under $\leq$ form a preorder. The *order of extremal realisations* has as elements isomorphism classes of extremal realisations ordered according to the existence of a map between representatives of isomorphism classes. As we shall see, the order of extremal realisations forms a prime-algebraic domain [7] with complete primes represented by those extremal realisations which have a top element—a direct corollary of Proposition VI.4 in the next section. (We say a realisation has a top element when its carrier contains an element which dominates all other elements in the carrier.)

We provide examples illustrating the nature of extremal realisations. In the examples it is convenient to describe families of configurations by general event structures, taking advantage of the economic representation they provide.

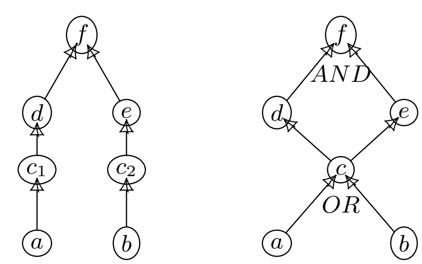
Example VI.1. This and the following example shows that extremal realisations with a top do not correspond to irreducible configurations. Below, on the right we show a general event structure with irreducible configuration $\{a, b, c, d\}$. On the left we show two extremals with tops d_1 and d_2 which both have the same irreducible configuration $\{a, b, c, d\}$ as their image. The lettering indicates the functions associated with the realisations, *e.g.* events d_1 and d_2 in the partial orders map to d in the general event structure.



Example VI.2. On the other hand there are extremal realisations with top of which the image is not an irreducible configuration. Below the extremal with top on the left describes a situation where d is enabled by b and c is enabled by a. It has image the configuration $\{a, b, c, d\}$ which is not irreducible, being the union of the two configurations $\{a\}$ and $\{b, c, d\}$.



Example VI.3. It is also possible to have extremal realisations in which an event depends on an event of the family having been enabled in two distinct ways, as in the following extremal realisation with top, on the left.



The extremal describes the event f being enabled by d and e where they are in turn enabled by different ways of enabling c. (Such phenomena will be disallowed in edc's.)

B. *A right adjoint to ges*

The right adjoint $er : \mathcal{G} \to \mathcal{E}_\equiv$ is defined on objects as follows. Let A be a general event structure. Define $er(A) = (P, \mathrm{Con}_P, \leq_P, \equiv_P)$ where

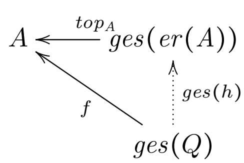
- P consists of a choice from within each isomorphism class of those extremals p of $\mathcal{C}^\infty(A)$ with a top element—we write $top_A(p)$ for the image of the top element in A;
- Causal dependency $\leq_P$ is $\leq$ on P;
- $X \in \mathrm{Con}_P$ iff $X \subseteq_{\mathrm{fin}} P$ and $top_A[X] \in \mathcal{C}^\infty(A)$ —the set $[X]$ is the $\leq_P$-downwards closure of X;
- $p_1 \equiv_P p_2$ iff $p_1, p_2 \in P$ and $top_A(p_1) = top_A(p_2)$.

Proposition VI.4. *The configurations of P, ordered by inclusion, are order-isomorphic to the order of extremal realisations of $\mathcal{C}^\infty(A)$: an extremal realisation ρ corresponds, up to isomorphism, to the configuration $\{p \in P \mid p \leq \rho\}$ of P; conversely, a configuration x of P corresponds to an extremal realisation $top_A : x \to A$ with carrier $(x, \leq)$, the restriction of the order of P to x.*

From the above proposition we see that the events of $er(A)$ correspond to completely-prime extremal realisations [7]. Henceforth we shall use the term 'prime extremal' instead of the clumsier 'extremal with top element.'

The component of the counit of the adjunction at A is given by the function top_A which determines a map $top_A : ges(er(A)) \to A$ of general event structures.

Theorem VI.5. *Let $A \in \mathcal{G}$. For all $f : ges(Q) \to A$ in $\mathcal{G}$, there is a map $h : Q \to er(A)$ in $\mathcal{E}_\equiv$ such that $f = top_A \circ ges(h)$ i.e. so the diagram*

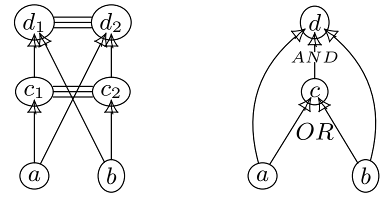


commutes. Moreover, if $h' : Q \to er(A)$ is a map in $\mathcal{E}_\equiv$ such that $f = top_A \circ ges(h')$, then $h' \equiv h$.

The theorem does not quite exhibit a standard adjunction, because the usual cofreeness condition specifying an adjunction is weakened to only having uniqueness up to $\equiv$. However the condition it describes does specify an exceedingly simple case of *pseudo adjunction* between 2-categories—a set together with an equivalence relation is a very simple example of a category (see Appendix A). As a consequence, whereas with the usual cofreeness condition allows us to extend the right adjoint to arrows, so obtaining a functor, in this case following that same line will only yield a pseudo functor er as right adjoint: thus extended, er will only necessarily preserve composition and identities up to $\equiv$.

The pseudo adjunction from $\mathcal{E}_\equiv$ to $\mathcal{G}$ cuts down to a reflection (*i.e.* the counit is a natural isomorphism) when we restrict to the subcategory of $\mathcal{G}$ where all general event structures are replete. Its right adjoint provides a pseudo functor embedding replete general event structures (and so families of configurations) in ese's.

Example VI.6. On the right we show a general event structure and on its left the ese which it gives rise to under er:



VII. EDC'S

Our major motivation in developing and exploring ese's was in order to extend strategies with parallel causes while maintaining the central operation of hiding. What about the other operation key to the composition of strategies, *viz.* pullback?

It is well-known to be hard to construct limits such as pullback within prime event structures, so that we often rely on first carrying out the constructions in stable families. It is sensible to seek an analogous way to construct pullbacks or pseudo pullbacks in $\mathcal{E}_\equiv$.

A. Equivalence families

In fact, the pseudo adjunction from $\mathcal{E}_\equiv$ to $\mathcal{G}$ factors through a more basic pseudo adjunction to families of configurations which also bear an equivalence relation on their underlying sets. An *equivalence-family* (ef) is a family of configurations $\mathcal{A}$ with an equivalence relation $\equiv_A$ on its underlying set $\bigcup \mathcal{A}$. We can identify a family of configurations $\mathcal{A}$ with the ef $(\mathcal{A}, =)$, taking the equivalence to be simply equality on the underlying set. A map $f : (\mathcal{A}, \equiv_A) \to (\mathcal{B}, \equiv_B)$ between ef's is a partial function $f : A \rightharpoonup B$ between their underlying sets which preserves $\equiv$ so that

$$x \in \mathcal{A} \Rightarrow fx \in \mathcal{B} \ \&\ \forall a_1, a_2 \in x.\ f(a_1) \equiv_B f(a_2) \Rightarrow a_1 \equiv_A a_2\,.$$

Composition is composition of partial functions. We regard two maps

$$f_1, f_2 : (\mathcal{A}, \equiv_A) \to (\mathcal{B}, \equiv_B)$$

as equivalent, and write $f_1 \equiv f_2$, iff they are equidefined and yield equivalent results. Composition respects $\equiv$. This yields a category of equivalence families $\mathcal{F}am_\equiv$ enriched in the category of sets with equivalence relations.

Clearly we can regard an ese $(P, \equiv_P)$ as an ef $(\mathcal{C}^\infty(P), \equiv_P)$ and a function which is a map of ese's as a map between the associated ef's, and this operation forms a functor. The functor has a pseudo right adjoint built from causal realisations in a very similar manner to er. The configurations of a general event structure form an ef with the identity relation as its equivalence. This operation is functorial and has a left adjoint which collapses an ef to a general event structure in a similar way to ges; the adjunction is enriched in equivalence relations. In summary, the pseudo adjunction

$$\mathcal{E}_\equiv \underset{ges}{\overset{er}{\rightleftarrows}} \mathcal{G} \quad (\top)$$

factors into a pseudo adjunction followed by an adjunction

$$\mathcal{E}_\equiv \underset{}{\overset{}{\rightleftarrows}}_{\top} \mathcal{F}am_\equiv \underset{}{\overset{}{\rightleftarrows}}_{\top} \mathcal{G}\,.$$

$\mathcal{F}am_\equiv$ has pullbacks and pseudo pullbacks which are easy to construct. For example, let $f:\mathcal{A}\to\mathcal{C}$ and $g:\mathcal{B}\to\mathcal{C}$ be total maps of ef's. Assume $\mathcal{A}$ and $\mathcal{B}$ have underlying sets A and B. Define $D =_{\rm def} \{(a,b)\in A\times B \mid f(a)\equiv_C g(b)\}$ with projections π_1 and π_2 to the left and right components. On D, take $d\equiv_D d'$ iff $\pi_1(d)\equiv_A \pi_1(d')$ and $\pi_2(d)\equiv_B \pi_2(d')$. Define a family of configurations of the *pseudo pullback* to consist of $x\in\mathcal{D}$ iff $x\subseteq D$ such that $\pi_1 x\in\mathcal{A}$ & $\pi_2 x\in\mathcal{B}$, and

$$\forall d\in x\exists d_1,\cdots,d_n\in x.\ d_n = d\ \&$$
$$\forall i\le n.\ \pi_1\{d_1,\cdots,d_i\}\in\mathcal{A}\ \&\ \pi_2\{d_1,\cdots,d_i\}\in\mathcal{B}\,.$$

The ef $\mathcal{D}$ with maps π_1 and π_2 is the pseudo pullback of f and g. It would coincide with pullback if $\equiv_C$ were the identity.

But unfortunately (pseudo) pullbacks in $\mathcal{F}am_\equiv$ don't provide us with (pseudo) pullbacks in $\mathcal{E}_\equiv$ because the right adjoint is only a pseudo functor; in general it will only carry pseudo pullbacks to bipullbacks. While $\mathcal{E}_\equiv$ does have bipullbacks (in which commutations and uniqueness are only up to the equivalence $\equiv$ on maps) it doesn't always have pseudo pullbacks or pullbacks—Appendix B. Whereas pseudo pullbacks and pullbacks are characterised up to isomorphism, bipullbacks are only characterised up to a weaker equivalence, that induced on objects by the equivalence on maps.While we could develop strategies with parallel causes in the broad context of ese's in general, doing so would mean that the composition of strategies that ensued was not defined up to isomorphism. This in turn would weaken our intended definition and characterisation of such strategies as those maps into games which are stable under composition with copycat.

B. Edc's defined

Fortunately there is a subcategory of $\mathcal{E}_\equiv$ which supports hiding, pullbacks and pseudo pullbacks. Define $\mathcal{EDC}$ to be the subcategory of $\mathcal{E}_\equiv$ with objects ese's satisfying

$$p_1,p_2\le p\ \&\ p_1\equiv p_2 \implies p_1 = p_2\,.$$

We call such objects *event structures with disjunctive causes* (edc's). In an edc an event can't causally depend on two distinct prime causes of a common disjunctive event, and so rules out realisations such as that illustrated in Example VI.3. In general, within $\mathcal{E}_\equiv$ we lose the local injectivity property that we're used to seeing for maps of event structures; the maps of event structures are injective from configurations, when defined. However for $\mathcal{EDC}$ we recover local injectivity w.r.t. prime configurations: if $f:P\to Q$ is a map in $\mathcal{EDC}$, then

$$p_1,p_2\le_P p\ \&\ f(p_1)=f(p_2) \implies p_1=p_2\,.$$

The factorisation property associated with hiding in $\mathcal{E}_\equiv$ is inherited by $\mathcal{EDC}$.

As regards (pseudo) pullbacks, we are fortunate in that the complicated pseudo adjunction between ese's and ef's restricts to a much simpler adjunction, in fact a coreflection, between edc's and *stable* ef's. In an equivalence family $(\mathcal{A},\equiv_A)$ say a configuration $x\in\mathcal{A}$ is *unambiguous* iff

$$\forall a_1,a_2\in x.\ a_1\equiv_A a_2 \implies a_1=a_2\,.$$

An equivalence family $(\mathcal{A},\equiv_A)$, with underlying set of events A, is *stable* iff it satisfies

$$\forall x,y,z\in\mathcal{A}.\ x,y\subseteq z\ \&\ z \text{ is unambiguous } \Rightarrow x\cap y\in\mathcal{A} \text{ and}$$
$$\forall a\in A, x\in\mathcal{A}.\, a\in x \Rightarrow \exists z\in\mathcal{A}.\, z \text{ is unambiguous } \&\, a\in z\subseteq x\,.$$

In effect a stable equivalence family contains a stable subfamily of unambiguous configurations out of which all other configurations are obtainable as unions. Local to any unambiguous configuration x there is a partial order on its events $\le_x$: each $a\in x$ determines a *prime configuration*

$$[a]_x =_{\rm def} \bigcap\{y\in\mathcal{A}\mid a\in y\subseteq x\}\,,$$

the minimum set of events on which a depends within x; taking $a\le_x b$ iff $[a]_x\subseteq[b]_x$ defines causal dependency between $a,b\in x$. Write $\mathcal{SF}am_\equiv$ for the subcategory of stable ef's.

(Pseudo) pullbacks in stable ef's are obtained from those in ef's simply by restricting to those configurations which are unions of unambiguous configurations.

The configurations of an edc with its equivalence are easily seen to form a stable ef providing a full and faithful embedding of $\mathcal{EDC}$ in $\mathcal{SF}am_\equiv$. The embedding has a right adjoint Pr. It is built out of prime extremals but we can take advantage of the fact that in a stable ef unambiguous prime extremals have the simple form of prime configurations. From a stable ef$(\mathcal{A},\equiv_A)$ we produce an edc ${\rm Pr}(\mathcal{A},\equiv_A) =_{\rm def} (P,{\rm Con},\le,\equiv)$ in which P comprises the prime configurations with

$$[a]_x\equiv[a']_{x'} \text{ iff } a\equiv_A a'\,,$$
$$Z\in{\rm Con} \text{ iff } Z\subseteq P\ \&\ \bigcup Z\in\mathcal{F} \text{ and,}$$
$$p\le p' \text{ iff } p,p'\in P\ \&\ p\subseteq p'\,.$$

The adjunction is enriched in the sense that its natural bijection preserves and reflects the equivalence on maps:

$$\mathcal{EDC} \underset{\longrightarrow}{\overset{\rm Pr}{\underset{\top}{\longleftarrow}}} \mathcal{SF}am_\equiv$$

We can now obtain a (pseudo) pullback in edc's by first forming the (pseudo) pullback of the stable ef's obtained as their configurations and then taking its image under the right adjoint Pr. We now have the constructions we need to support strategies based on edc's.

C. Coreflective subcategories of edc's

$\mathcal{EDC}$ is a coreflective subcategory of $\mathcal{E}_\equiv$; the right adjoint simply cuts down to those events satisfying the edc property. In turn $\mathcal{EDC}$ has a coreflective subcategory $\mathcal{E}^0_\equiv$ comprising those edc's which satisfy

$$\{p_1,p_2\}\in{\rm Con}\ \&\ p_1\equiv p_2 \implies p_1=p_2\,.$$

Consequently its maps are traditional maps of event structures which preserve the equivalence. We derive adjunctions

$$\mathcal{E}^0_\equiv \underset{\longrightarrow}{\overset{}{\underset{\top}{\longleftarrow}}} \mathcal{EDC} \underset{\longrightarrow}{\overset{}{\underset{\top}{\longleftarrow}}} \mathcal{E}_\equiv \underset{\underset{ges}{\longrightarrow}}{\overset{er}{\underset{\top}{\longleftarrow}}} \mathcal{G}\,.$$

Note the last is only a pseudo adjunction. Consequently we obtain a pseudo adjunction from $\mathcal{E}^0_\equiv$, the a category of prime event structures with equivalence relations and general event structures—this makes good the promise of Section III-C. Inspecting the composite of the last two adjunctions, we also obtain the sense in which replete general event structures embed via a reflection in edc's.

There is an obvious 'inclusion' functor from the category of prime event structures $\mathcal{E}$ to the category $\mathcal{EDC}$; it extends an event structure with the identity equivalence. Regarding $\mathcal{EDC}$ as a plain category, so dropping the enrichment by equivalence relations, the 'inclusion' functor

$$\mathcal{E} \hookrightarrow \mathcal{EDC}$$

has a right adjoint, *viz.* the forgetful functor which given an edc $P = (P, \leq, \mathrm{Con}, \equiv)$ produces an event structure $P_0 = (P, \leq, \mathrm{Con}')$ by dropping the equivalence $\equiv$ and modifying the consistency relation to

$$X \in \mathrm{Con}' \text{ iff } X \subseteq P \;\&\; X \in \mathrm{Con} \;\&\; p_1 \not\equiv p_2, \text{ for all } p_1, p_2 \in X\,.$$

The configurations of P_0 are the unambiguous configurations of P. The adjunction is a coreflection because the inclusion functor is full. Of course it is not the case that the adjunction is enriched: the natural bijection of the adjunction cannot respect the equivalence on maps; it cannot compose with the pseudo adjunction from $\mathcal{EDC}$ to $\mathcal{G}$ to yield a pseudo adjunction from $\mathcal{E}$ to $\mathcal{G}$.

Despite this the adjunction from $\mathcal{E}$ to $\mathcal{EDC}$ has many useful properties. Of importance for us is that the functor forgetting equivalence will preserve all limits and especially pullbacks. It is helpful in relating composition of edc-strategies to the composition of strategies based on prime event structures in [10]. In composing strategies in edc's we shall only be involved with pullbacks of maps $f : A \to C$ and $g : B \to C$ of edc's. (When C is essentially an event structure, *i.e.* an edc in which the equivalence is the identity relation, the construction of such pullbacks coincides with that of pseudo pullbacks.) While this does not entail that composition of strategies is preserved by the forgetful functor—because the forgetful functor does not commute with hiding—it will give us a strong relationship, expressed as a map, between composition of the two kinds of strategies (based on edc's and based on prime event structures) after and before applying the forgetful functor. This has been extremely useful in some proofs, in importing results about concurrent strategies from [10].

VIII. Strategies based on edc's

We develop strategies in edc's in a similar way to that of strategies in [10], *viz.* as certain maps stable under composition with copycat.

An *edc with polarity* comprises $(P, \equiv, pol)$, an edc $(P, \equiv)$ in which each element $p \in P$ carries a polarity $pol(p)$ which is $+$ or $-$, according as it represents a move of Player or Opponent, and where the equivalence relation $\equiv$ respects polarity.

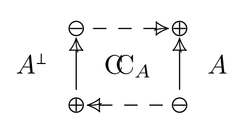
A *map* of edc's with polarity is a map of the underlying edc's which preserves polarity when defined. The adjunctions of the previous section are undisturbed by the addition of polarity.

There are two fundamentally important operations on two-party games. One is that of forming the dual game in which the moves of Player and Opponent are reversed. On an edc with polarity A this amounts to reversing the polarities of events to produce the dual $A^\perp$; all other relations including equivalence remain the same. The other operation is a simple parallel composition of games, achieved on edc's with polarity A and B by simply juxtaposing them, ensuring a finite subset of events is consistent if its overlaps with the two games are individually consistent, to form $A \| B$; its equivalence is inherited from those of A and B; its configurations x correspond to pairs of configurations x_1 of A and x_2 of B.

A game is represented by an edc with polarity. A *pre-strategy* in edc's, or an *edc pre-strategy*, in a game A is a total map $\sigma : S \to A$ of edc's. A pre-strategy from a game A to a game B is a pre-strategy in the game $A^\perp \| B$. We shall shortly refine the notion of pre-strategy to strategy. By a strategy in a game we will mean a strategy for Player. A strategy for Opponent, or a counter-strategy, in a game A will be identified with a strategy in $A^\perp$. A map $f : \sigma \Rightarrow \sigma'$ of edc pre-strategies $\sigma : S \to A$ and $\sigma' : S' \to A$ is a map $f : S \to S'$ of edc's with polarity such that $\sigma = \sigma' f$; in the standard way this determines isomorphisms of edc pre-strategies, important for us in a moment.

A. Copycat

An important example of a strategy is the *copycat* strategy for a game A. This is a strategy in the game $A^\perp \| A$ which, following the spirit of a copycat, has Player moves copy the corresponding Opponent moves in the other component. In more detail, the copycat strategy comprises $\text{æ}_A : \mathbb{CC}_A \to A^\perp \| A$ where $\mathbb{CC}_A$ is obtained by adding extra causal dependencies to $A^\perp \| A$ so that any Player move in either component causally depends on its copy, an Opponent move, in the other [10]. This generates a partial order of causal dependency. The equivalence of $\mathbb{CC}_A$ is that of $A^\perp \| A$. A finite set is taken to be consistent if its down-closure w.r.t. the order generated is consistent in $A^\perp \| A$; the map æ_A is the identity function on events. We illustrate the construction on the simple game comprising a Player move causally dependent on a single Opponent move:



B. Composing edc pre-strategies

In composing two edc pre-strategies one σ in $A^\perp \| B$ and another τ in $B^\perp \| C$ one firstly instantiates the Opponent moves in component B by Player moves in $B^\perp$ and *vice versa*, and then secondly hides the resulting internal moves over B. The first step is achieved efficiently via pullback. Temporarily ignoring polarities, the pullback in edc's "synchronises" matching moves of S and T over the game B. But we require a strategy over the game $A^\perp \| C$ and the pullback $T \circledast S$ has internal moves over the game B. We achieve this via the projection of $T \circledast S$ to its moves over A and C. We make use of the partial map from $A\|B\|C$ to $A\|C$ which acts as the identity function on A and C and is undefined on B. The composite partial map

$$\begin{array}{ccccc} & & A \parallel T & & \\ & \overset{\pi_2}{\nearrow} & & \overset{A\|\tau}{\searrow} & \\ T \circledast S \rightarrowtail & & & & A \parallel B \parallel C \rightharpoonup A \parallel C \\ & \underset{\pi_1}{\searrow} & & \underset{\sigma\|C}{\nearrow} & \\ & & S \parallel C & & \end{array}$$

has defined part, yielding the composition

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

once we reinstate polarities. The composition of edc strategies $\tau \odot \sigma$ is a form of synchronised composition of processes followed by the hiding of internal moves, a view promulgated by Abramsky within traditional game semantics of programs.

C. Edc strategies

The article [10] characterises through the properties of "innocence" and "receptivity" those pre-strategies based on event structures which are stable under composition with the copycat strategy; the characterisation becomes the definition of concurrent strategy.[1] We imitate [10] and provide necessary and sufficient conditions for a pre-strategy in edc's to be stable up to isomorphism under composition with copycat. Fortunately we can inherit a great deal from the proof of [10] via the coreflection of event structures in edc's of Section VII-C.

Say an edc pre-strategy $\sigma : S \to A$ is an *edc strategy* iff
(i) the image $\sigma_0 : S_0 \to A_0$ of σ (under the right adjoint to the inclusion of event structures in edc's) is a strategy of concurrent games, as in [10]; and in addition
(ii) $s_1 \equiv_S s_2 \iff \sigma(s_1) \equiv_A \sigma(s_2)$, for all $s_1, s_2 \in S$; with
(iii) $x \xrightarrow{s}\!\!\subset z \;\&\; x \xrightarrow{s'}\!\!\subset z' \;\&\; s$ −ve $\& \; \sigma z \uparrow \sigma z' \implies z \uparrow z'$.

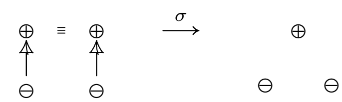
Theorem VIII.1. *Let $\sigma : S \to A$ be an edc pre-strategy. Then, $\sigma \cong \text{\ae}_A \odot \sigma$ iff σ is an edc strategy as above.*

We obtain a bicategory in which the objects are games, the arrows $\sigma : A \xrightarrow{+}\!\!\!\!\!\!\rightarrow B$ are edc strategies σ from A to B and 2-cells are total maps of pre-strategies with vertical composition their usual composition. Horizontal composition is given by composition $\odot$, which extends to a functor on 2-cells via the universality of pullback and the factorisation property of hiding. An edc strategy $\sigma : A \xrightarrow{+}\!\!\!\!\!\!\rightarrow B$ corresponds to its dual $\sigma^\perp : B^\perp \xrightarrow{+}\!\!\!\!\!\!\rightarrow A^\perp$, yielding (a bicategorical variant of) compact-closure though this can weaken to $*$-autonomy with the addition of extra structure such as winning conditions or pay-off. The story is undisturbed if we restrict to *rigid* 2-cells which preserve causal dependency, or restricting further to 2-cells which are rigid embeddings which gives the machinery to define strategies recursively.

An edc strategy $\sigma : S \to A$ is *deterministic* if S is deterministic as an edc with polarity, *i.e.* in $\mathcal{C}(S)$,

$$x \xrightarrow{s}\!\!\subset z \;\&\; x \xrightarrow{s'}\!\!\subset z' \;\&\; pol(s) = + \Rightarrow z \uparrow z' \,.$$

Copycat strategies $\text{\ae}_A$ are deterministic iff the game A is *race-free:* if $x \xrightarrow{a}\!\!\subset z$ and $x \xrightarrow{a'}\!\!\subset z'$ with a and a' of opposing polarities, then $z \uparrow z'$, in $\mathcal{C}(A)$. We obtain a sub-bicategory of deterministic edc strategies between race-free games analogous to that of [10]. But now there are deterministic strategies with parallel causes, including the strategy sketched informally in the Introduction in which Player makes a move iff Opponent makes one or more of their moves:



Along the same lines there is now a deterministic strategy for computing "parallel or."

IX. Probabilistic edc strategies

A. Probabilistic event structures

A probabilistic event structure essentially comprises an event structure together with a continuous valuation on the Scott-open sets of its domain of configurations.[2] The continuous valuation assigns a probability to each open set and can then be extended to a probability measure on the Borel sets [4]. However open sets are several levels removed from the events of an event structure, and an equivalent but more workable definition is obtained by considering the probabilities of sub-basic open sets, generated by single finite configurations; for each finite configuration x this specifies $\text{Prob}(x)$ the probability of obtaining events x, so as result a configuration which extends the finite configuration x. Such valuations on configuration determine the continuous valuations from which they arise, and can be characterised through the device of "drop functions" which measure the drop in probability across certain generalised intervals. The characterisation yields a workable general definition of probabilistic event structure as event structures with *configuration-valuations*, *viz.* functions from finite configurations to the unit interval for which the drop functions are always nonnegative [14].

In detail, a *probabilistic event structure* comprises an event structure E with a *configuration-valuation*, a function v from the finite configurations of E to the unit interval which is

(normalized) $v(\varnothing) = 1$ and has

[1] A total map of event structures with polarity $\sigma : S \to A$ is *receptive* if when $\sigma x \xrightarrow{a}\!\!\subset$ with a −ve there is a unique s such that $x \xrightarrow{s}\!\!\subset$ and $\sigma(s) = a$. It is *innocent* if $s \rightarrowtail s'$ with s +ve or s' −ve implies $\sigma(s) \rightarrowtail \sigma(s')$.

[2] A *Scott-open* subset of configurations is upwards-closed w.r.t. inclusion and such that if it contains the union of a directed subset S of configurations then it contains an element of S. A *continuous valuation* is a function w from the Scott-open subsets of $\mathcal{C}^\infty(E)$ to $[0,1]$ which is *((normalized)* $w(\mathcal{C}^\infty(E)) = 1$; *(strict)* $w(\varnothing) = 0$; *(monotone)* $U \subseteq V \implies w(U) \le w(V)$; *(modular)* $w(U \cup V) + w(U \cap V) = w(U) + w(V)$; and *(continuous)* $w(\bigcup_{i \in I} U_i) = \sup_{i \in I} w(U_i)$, for *directed* unions.

(non −ve drop) $d_v[y; x_1, \cdots, x_n] \geq 0$ when $y \subseteq x_1, \cdots, x_n$ for finite configurations $y, x_1, \cdots, x_n$ of E,

where the "drop" across the generalized interval starting at y and ending at one of the $x_1, \cdots, x_n$ is given by

$$d_v[y; x_1, \cdots, x_n] =_{\text{def}} v(y) - \sum_I (-1)^{|I|+1} v(\bigcup_{i \in I} x_i)$$

—the index I ranges over nonempty $I \subseteq \{1, \cdots, n\}$ such that the union $\bigcup_{i \in I} x_i$ is a configuration. The "drop" $d_v[y; x_1, \cdots, x_n]$ gives the probability of the result being a configuration which includes the configuration y and does not include any of the configurations $x_1, \cdots, x_n$.

If $x \subseteq y$ in $\mathcal{C}(E)$, then $\mathrm{Prob}(y \mid x) = v(y)/v(x)$; this is the probability that the resulting configuration includes the events y conditional on it including the events x.

B. Probability with an Opponent

This prepares the ground for a general definition of distributed probabilistic strategies, based on edc's. Firstly though, we should restrict to race-free games, in particular because without copycat being deterministic there would be no probabilistic identity strategies. A probabilistic edc strategy in a game A, is an edc strategy $\sigma : S \to A$ in which we endow S with probability, while taking account of the fact that in the strategy Player can't be aware of the probabilities assigned by Opponent. We do this through extending the definition of configuration-valuation via an axiom (lmc) which implies the Limited Markov Condition, LMC, of the Introduction.

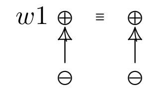
Precisely, a *configuration-valuation* is now a function v, from finite configurations of S to the unit interval, which is

(normalized) $v(\varnothing) = 1$, satisfies
(lmc) $v(x) = v(y)$ when $x \subseteq^- y$ for finite configurations x, y of S, and the
(+ve drop condition) $d_v[y; x_1, \cdots, x_n] \geq 0$ when $y \subseteq^+ x_1, \cdots, x_n$ for finite configurations of S.

When $x \subseteq^+ y$ in $\mathcal{C}(S)$, we can still express $\mathrm{Prob}(y \mid x)$, the conditional probability of Player making the moves $y \setminus x$ given x, as $v(y)/v(x)$. In fact all such conditional probabilities determine v via normalisation and lmc. As A is race-free it follows S is also race-free. Hence if x is a finite configuration at which $x \overset{\oplus}{-\!\!\!\subset}$ and $x \overset{\ominus}{-\!\!\!\subset}$ then $x \cup \{\oplus, \ominus\}$ is also a configuration, and both moves are $\oplus, \ominus$ are causally independent (or concurrent). From lmc we obtain LMC directly: $\mathrm{Prob}(\oplus \mid x) = \mathrm{Prob}(x, \oplus \mid x) = v(x \cup \{\oplus\})/v(x) = v(x \cup \{\oplus, \ominus\})/v(x \cup \{\ominus\}) = \mathrm{Prob}(x, \oplus, \ominus \mid x, \ominus) = \mathrm{Prob}(\oplus \mid x, \ominus)$. A dual form of LMC will hold of a counterstrategy, a strategy for Opponent; the LMCs for Player and Opponent will together ensure the probabilistic independence of Player and Opponent moves from a common configuration.

A *probabilistic edc strategy* in race-free game A comprises an edc strategy $\sigma : S \to A$ with a configuration-valuation v for S. A *probabilistic edc strategy* between race-free games A to B is a probabilistic edc strategy in $A^\perp \| B$. Note that the configuration-valuation of an edc doesn't necessarily respect the equivalence of the edc; different prime causes of a common disjunctive event may well be associated with different probabilities.

Example IX.1. Recall the game of the Introduction. In the edc strategy



of Section VIII-C individual success of the two watchers may be associated with probabilities $p_1 \in [0,1]$ and $p_2 \in [0,1]$, respectively, and their joint success with $q \in [0,1]$ provided they form a configuration valuation v. In other words, $v(x) = p_1$ if x contains $w1$ and not $w2$; $v(x) = p_2$ if x contains $w2$ and not $w1$; and $v(x) = q$ if x contains both $w1$ and $w2$; $v(x) = 1$ otherwise; and $p_1 + p_2 - q \leq 1$, in order to satisfy the +-drop condition. To enliven this a little we might imagine the two watchers have a drinking problem and the correlation depends on whether they are sharing from a common bottle; if they had their own bottles we might imagine the drunken unreliability of one independent of that of the other, so $q = p_1.p_2$. □

We extend the usual composition of edc strategies to probabilistic edc strategies. Assume probabilistic edc strategies $\sigma : S \to A^\perp \| B$, with configuration-valuation v_S, and $\tau : T \to B^\perp \| C$ with v_T. Their composition is defined to be $\tau \odot \sigma : T \odot S \to A^\perp \| C$ with a configuration-valuation v given by

$$v(x) = v_S(\pi_1^S x).v_T(\pi_2^T x)$$

for x a finite configuration of $T \odot S$. The configuration $\pi_1^S x$ is the component in $\mathcal{C}(S)$ of the projection $\pi_1 x \in \mathcal{C}(S \| C)$ from the pullback defined in Section VIII-B; similarly $\pi_2^T x$ is the T-component of $\pi_2 x$. The proof that v is indeed a configuration-valuation is quite subtle and relies heavily on properties of "drop" functions. Parallel composition $\sigma \| \tau$ is a special case.

C. A bicategory of probabilistic edc strategies

We obtain a bicategory of probabilistic edc strategies in which objects are race-free games. Maps are probabilistic edc strategies. Identities are given by copycat strategies, which for race-free games are deterministic, so permit configuration-valuations which are constantly 1. Generally, a probabilistic edc strategy is *deterministic* if its configuration-valuation assigns 1 to all finite configurations; its underlying edc strategy is then necessarily deterministic too.

The 2-cells of the bicategory require consideration. Whereas we can always "push forward" a probability measure from the domain to the codomain of a measurable function this is not true generally for configuration-valuations involving Opponent moves. However:

Theorem IX.2. *Let $f : \sigma \Rightarrow \sigma'$ be a rigid 2-cell between edc strategies $\sigma : S \to A$ and $\sigma' : S' \to A$. Let v be a configuration-valuation on S. Defining, for $y \in \mathcal{C}(S')$,*

$$(fv)(y) =_{\text{def}} \sup_X \sum_{\varnothing \neq Z \subseteq X \,\&\, Z\uparrow} (-1)^{|Z|+1} v(\bigcup Z)$$

as X ranges over finite subsets of $\{x \in \mathcal{C}(S) \mid y = fx\}$, yields a configuration-valuation fv of S' —the push-forward *of v.*

A 2-cell from σ, v to σ', v' is a 2-cell $f : \sigma \Rightarrow \sigma'$ of edc strategies in which $f : S \to S'$ is rigid and the push-forward fv satisfies $(fv)(x') \le v'(x')$, for all configurations $x' \in \mathcal{C}(S')$. The situation restricts to 2-cells which are rigid embeddings preserving the value of the configuration-valuations, giving us the technology to define probabilistic strategies recursively.

X. Constructions on probabilistic edc strategies

With edc's we have made good our promise of an extension to traditional event structures which supports parallel causes and hiding, as well as the mix of probability and nondeterminism needed to account for probabilistic distributed strategies. The bicategory of now-probabilistic edc strategies remains compact-closed and supports a metalanguage, a mix of dataflow and higher-order process algebra, detailed in [2].

As is to be expected, with the addition of probability there is an additional construction of *probabilistic sum*: if $\sigma_i : A$ are probabilistic edc strategies in a game A for $i \in I$, assumed countable, with $p_i, i \in I$, a sub-probability distribution, we can form $\Sigma_{i \in I} p_i \sigma_i : A$. There is a form of *synchronised composition* $\sigma_1 \wedge \sigma_2 : A$ of two probabilistic edc strategies σ_1 and σ_2 in a common game A obtained as their pullback.

There are also new constructions hinging on parallel causes:

1) Duplication and contraction: Duplication of arguments is essential if we are to support the recursive definition of strategies. We duplicate arguments through an edc strategy $\delta_A : A \twoheadrightarrow A \| A$. Intuitively it behaves like the copycat strategy but where a Player move in the left component may be caused in parallel by either of its corresponding Opponent moves from the two components on the right. We show δ_A when A consists of a single Player move $\oplus$ and, respectively, a single Opponent move $\ominus$:

The general definition is in Appendix C. In general, duplication δ_A is deterministic iff $A^\perp$ is deterministic as an edc with polarity. Then δ_A extends directly to a probabilistic edc strategy and is a comonoid. This depends crucially on parallel causes. By duality we obtain a *contraction* strategy $\gamma_A : A \| A \twoheadrightarrow A$ which is deterministic iff A is deterministic.

2) Disjunction: Provided game A is deterministic, a form of *disjunction* $\sigma_1 \vee \sigma_2 : A$ of two probabilistic edc strategies $\sigma_1 : A$ and $\sigma_2 : A$ is obtained as $\gamma_A \odot (\sigma_1 \| \sigma_2)$, the composition of the strategy $\sigma_1 \| \sigma_2$ with the deterministic contraction strategy $\gamma_A : A \| A \twoheadrightarrow A$. Such a disjunction of strategies can introduce parallel causes: for example, the two parallel watchers of Example IX.1, in the case where they are independent (so $q = p_1 p_2$), is obtained as a disjunction of two strategies, one for each of the individual watchers.

3) Recursion: Once we have duplication strategies we can treat recursion using standard machinery of rigid embeddings based on inclusions [12]; recall that 2-cells, the maps between probabilistic strategies, include rigid embeddings. Given an expression $t(x) : A \| B^\perp$ with a variable x in A and parameters in B, the recursive definition $\mu\, x{:}A.\, t : A \| B^\perp$ denotes the least fixed point amongst probabilistic strategies X in $B^\perp \| A$ of the continuous operation $F(X) = t \odot (\mathrm{id}_B \| X) \odot \delta_B$. The definition requires $B^\perp$ is deterministic so that duplication δ_B, associated with the copying of parameters, is a probabilistic strategy.

4) Dataflow: If we restrict to games in which all moves are those of Player we obtain a framework for probabilistic dataflow. Through allowing parallel causes, its deterministic strategies $\sigma : A \twoheadrightarrow B$ will yield (non-injectively) *all* Scott-continuous functions from $\mathcal{C}^\infty(A)$ to $\mathcal{C}^\infty(B)$, not just the stable functions, as is the case for concurrent strategies [10].

5) New deterministic strategies: In restricting to the sub-bicategory of deterministic edc strategies we essentially ignore all probabilistic information. Nevertheless deterministic strategies with parallel causes have their uses, most obviously in the game semantics of languages with "parallel or."

Edc's significantly broaden our ways to describe probabilistic games and strategies in a compositional fashion. Constructions on strategies induce constructions on distributions. Although fragments can support probabilistic sampling and conditioning, the main features of probabilistic programming, it is not clear how conditioning can extend to more general probabilistic edc strategies, to support the machine-learning of strategies. The games here can be easily extended to games with *imperfect information* and *pay-off* as in [14]; then they include the widely-used Blackwell games [6].

Acknowledgments

Thanks to Simon Castellan, Pierre Clairambault, Mai Gehrke, Jonathan Hayman and Martin Hyland for advice and encouragement, to ENS Paris for supporting Marc de Visme's internship and to the ERC for Advanced Grant ECSYM.

References

[1] Simon Castellan, Pierre Clairambault, and Glynn Winskel. Symmetry in concurrent games. In *LICS'14*. ACM, 2014.
[2] Simon Castellan, Jonathan Hayman, Marc Lasson, and Glynn Winskel. Strategies as concurrent processes. *ENTCS*, 308, 2014.
[3] Ioana Cristescu. *Operational and denotational semantics for the reversible pi-calculus*. PhD thesis, PPS, Université Paris Diderot, 2015.
[4] Claire Jones and Gordon Plotkin. A probabilistic powerdomain of valuations. In *LICS '89*. IEEE Computer Society, 1989.
[5] G. M. Kelly. *Basic concepts of enriched category theory*. Lecture Notes in Mathematics 64. CUP, 1982.
[6] Donald Martin. The determinacy of Blackwell games. *Journal of Symbolic Logic*, 63(4):1565–1581, 1998.
[7] Mogens Nielsen, Gordon Plotkin, and Glynn Winskel. Petri nets, event structures and domains. *TCS*, 13:85–108, 1981.
[8] Judea Pearl. *Causality*. CUP, 2013.
[9] John Power. 2-categories. *BRICS Notes Series NS-98-7*, 1998.
[10] Silvain Rideau and Glynn Winskel. Concurrent strategies. In *LICS 2011*.
[11] Glynn Winskel. *Events in computation*. 1980. PhD thesis, Edinburgh.
[12] Glynn Winskel. Event structure semantics for CCS and related languages. In *ICALP'82*, volume 140 of *LNCS*. Springer, 1982.
[13] Glynn Winskel. Event structures. In *Advances in Petri Nets*, volume 255 of *LNCS*, pages 325–392. Springer, 1986.
[14] Glynn Winskel. Distributed probabilistic and quantum strategies. *Electr. Notes Theor. Comput. Sci. 298: 403-425*, 2013.
[15] Glynn Winskel. On probabilistic distributed strategies. In *ICTAC 2015*, volume 9399 of *Lecture Notes in Computer Science*. Springer, 2015.
[16] Glynn Winskel. *Event Structures, Stable Families and Concurrent Games*. http://www.cl.cam.ac.uk/~gw104/ecsym-notes.pdf, 2016.
[17] Glynn Winskel and Mogens Nielsen. *Handbook of Logic in Computer Science 4*, chapter Models for Concurrency, pages 1–148. OUP, 1995.

Appendix

A. Equivalence-enriched categories

Here we explain in more detail what we mean when we say "enriched in the category of of sets with equivalence relations" and employ terms such as "enriched adjunction," "pseudo adjunction" and "pseudo pullback."

Equiv is the category of *equivalence relations*. Its objects are $(A, \equiv_A)$ comprising a set A on which there is an equivalence relation $\equiv_A$. Its maps $f : (A, \equiv_A) \to (B, \equiv_B)$ are total functions $f : A \to B$ which preserve equivalence.

We shall use some basic notions from enriched category theory [5]. We shall be concerned with categories enriched in Equiv, called Equiv-enriched categories, in which the homsets possess the structure of equivalence relations, respected by composition. This is the sense in which we say categories are enriched in (the category of) equivalence relations. We similarly borrow the concept of an Equiv-enriched functor between Equiv-enriched categories which preserve equivalence in acting on homsets. An Equiv-enriched adjunction is a usual adjunction in which the natural bijection preserves and reflects equivalence.

Because an object in Equiv can be regarded as a (very simple) category, we can regard Equiv-enriched categories as a (very simple) 2-categories to which notions from 2-categories apply [9].

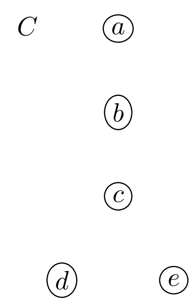
A *pseudo functor* between Equiv-enriched categories is like a functor but the usual laws only need hold up to equivalence. A *pseudo adjunction* (or biadjunction) between 2-categories permits a weakening of the usual natural isomorphism between homsets, now also categories, to a natural equivalence of categories. In the special case of a pseudo adjunction between Equiv-enriched categories the equivalence of homset categories amounts to a pair of $\equiv$-preserving functions whose compositions are $\equiv$-equivalent to the identity function. With traditional adjunctions by specifying the action of one adjoint solely on objects we determine it as a functor; with pseudo adjunctions we can only determine it as a pseudo functor—in general a pseudo adjunction relates two pseudo functors. Pseudo adjunctions compose in the expected way. An Equiv-enriched adjunction is a special case of a 2-adjunction between 2-categories and a very special case of pseudo adjunction. In this article there are many cases in which we compose an Equiv-enriched adjunction with a pseudo adjunction to obtain a new pseudo adjunction.

Similarly we can specialise the notions pseudo pullbacks and bipullbacks from 2-categories to Equiv-enriched categories. Let $f : A \to C$ and $g : B \to C$ be two maps in an Equiv-enriched category. A *pseudo pullback* of f and g is an object D and maps $p : D \to A$ and $q : D \to B$ such that $f \circ p \equiv g \circ q$ which satisfy the further property that for any D' and maps $p' : D' \to A$ and $q' : D' \to B$ such that $f \circ p' \equiv g \circ q'$, there is a unique map $h : D' \to D$ such that $p' = p \circ h$ and $q' = q \circ h$. There is an obvious weakening of pseudo pullbacks to the situation in which the uniqueness is replaced by uniqueness up to $\equiv$ and the equalities by $\equiv$—these are simple special cases of bilimits called *bipullbacks*.

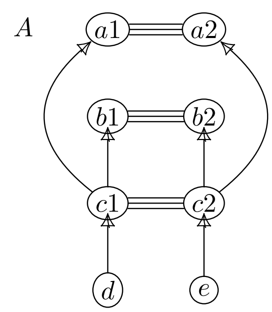
Right adjoints in a 2-adjunction preserve pseudo pullbacks whereas right adjoints in a pseudo adjunction are only assured to preserve bipullbacks.

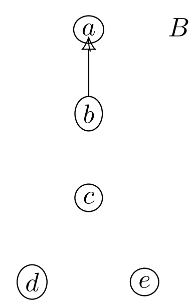
B. On (pseudo) pullbacks of ese's

We show that the enriched category of ese's $\mathcal{E}_\equiv$ does not always have pullbacks and pseudo pullbacks of maps $f : A \to C$ and $g : B \to C$, the reason why we use the subcategory $\mathcal{EDC}$, which does, as a foundation on which to develop strategies with parallel causes. It suffices to exhibit the lack of pullbacks when C is an (ese of an) event structure as then pullbacks and pseudo pullbacks coincide. Take C to be



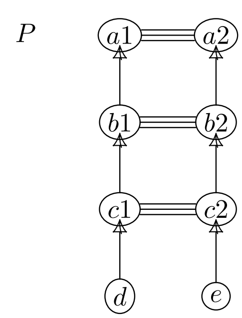
with A and B being respectively





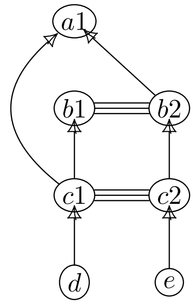
with the obvious maps $f : A \to C$ and $g : B \to C$ (given by the lettering). In fact, A and B are edc's.

The pullback in edc's $\mathcal{EDC}$ does exist and is given by



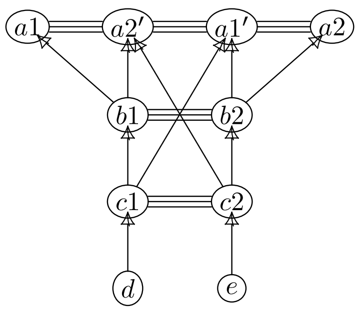
with the obvious projection maps. However this is not a pullback in $\mathcal{E}_\equiv$. Consider the ese

D



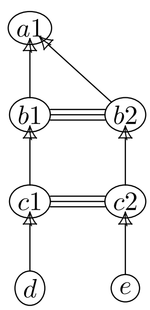
with the obvious total maps to A and B; they form a commuting square with f and g. This cannot factor through P: event $b2$ has to be mapped to $b2$ in P, but then $a1$ cannot be mapped to $a1$ (it wouldn't yield a map) nor to $a2$ (it would violate commutation required of a pullback).

There is a bipullback got by applying the pseudo functor er to the pullback in ef's:



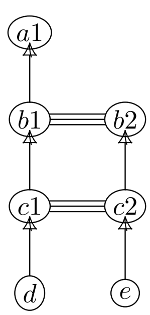
But this is not a pullback because in the ese E below the required mediating map is not unique in that $a1$ can go to either $a1$ or $a1'$:

E



In fact, there is no pullback of f and g. To show this we use an additional ese:

F



Suppose Q with projection maps to A and B were a pullback of f and g in $\mathcal{E}_\equiv$. Consider the three ese's D, E and F with their obvious maps to A and B; in each case they form a commuting square with f and g. There are three unique maps $h_D : D \to Q$, $h_E : E \to Q$, and $h_F : F \to Q$ such that the corresponding pullback diagrams commute. We remark that there are also obvious maps $k_D : E \to D$ and $k_F : E \to F$ (given by the lettering) which commute with the maps to the components A and B. By uniqueness, we have $h_D \circ k_D = h_E = h_F \circ k_F$, so we have $h_D(a1) = h_F(a1)$. From the definition of the maps, the event $h_D(a1) = h_F(a1)$ has at most one $\leq$-predecessor in Q which is sent to b in C (as D only has one). Because of the projection to B, it has at least one (as B has one). So the event $h_D(a1) = h_F(a1)$ has exactly one predecessor which is sent to b. From the definition of maps, this event is $h_D(b2)$ which equals $h_F(b1)$. But $h_D(b2)$ cannot equal $h_F(b1)$ as they go to two different events of A —a contradiction. Hence there can be no pullback of f and g in $\mathcal{E}_\equiv$. (By adding intermediary events, we would encounter essentially the same example in the composition, before hiding, of strategies if they were to be developed within the broader category of ese's.)

C. The edc duplication strategy

We present the general definition of the edc *duplication* strategy $\delta_A : A \mathrel{-\!\!\!+\!\!\!\!>} A \| A$ for a race-free game A.

For each triple (x, y_1, y_2), where $x \in \mathcal{C}(A^\perp)$ and $y_1, y_2 \in \mathcal{C}(A)$, which is *balanced*, *i.e.*

$$\forall a \in y_1 \cup y_2.\ pol_A(a) = + \implies a \in x \text{ and}$$
$$\forall a \in x.\ pol_{A^\perp}(a) = + \implies a \in y_1 \text{ or } a \in y_2\,,$$

and *choice* function $\chi : x^+ \to \{1, 2\}$, from the positive events of x denoted by x^+, such that $\chi(a) = 1 \implies a \in y_1$ and $\chi(a) = 2 \implies a \in y_2$, the order $q(x, y_1, y_2; \chi)$ is defined to have underlying set $\{0\} \times x \cup \{1\} \times y_1 \cup \{2\} \times y_2$ with order generated by that inherited from $A^\perp \| A \| A$ together with

$$\{((0,a),\,(1,a)) \mid a \in y_1 \;\&\; pol_A(a) = +\} \cup$$
$$\{((0,a),(2,a)) \mid a \in y_2 \;\&\; pol_A(a) = +\} \cup$$
$$\{((\chi(a),a),\,(0,a)) \mid a \in x \;\&\; pol_{A^\perp}(a) = +\}\,.$$

Now we can define $\delta_A : D_A \to A^\perp \| A \| A$. The edc D_A comprises $(D_A, \leq, \mathrm{Con}, \equiv, pol)$ with

events D_A consisting of all $d = q(x, y_1, y_2; \chi)$, for balanced (x, y_1, y_2) and choice function χ, which have a top element $\delta_A(d)$;
causal dependency $d \leq d'$ iff there is a rigid inclusion map from d into d' (regarded as event structures);
consistency $X \in \mathrm{Con}$ iff $X \subseteq_{\mathrm{fin}} D_A$ and the image of its $\leq$-down-closure, $\delta_A[X]$, is consistent in $A^\perp \| A \| A$;
equivalence $d \equiv d'$ iff $\delta_A(d) \equiv \delta_A(d')$, *i.e.* they have equivalent top elements in $A^\perp \| A \| A$; and
with the *polarity* of events D_A inherited from the polarity of their top elements, *i.e.* $pol(d) = pol_A(\delta_A(d))$ for $d \in D_A$.

We obtain an edc strategy $\delta_A : A \mathrel{-\!\!\!+\!\!\!\!>} A \| A$ in which $\delta_A : D_A \to A^\perp \| A \| A$ sends a prime to its top element. The edc strategy δ_A forms a comonoid with counit $\perp : A \mathrel{-\!\!\!+\!\!\!\!>} \varnothing$.

The duplication strategy δ_A is deterministic iff no Opponent moves in A are in immediate conflict, *i.e.* if $x \xrightarrow{a_1}\subset$ and $x \xrightarrow{a_2}\subset$ in $\mathcal{C}(A)$ and $pol_A(a_1) = pol_A(a_2) = -$ then $x \cup \{a_1, a_2\} \in \mathcal{C}(A)$. Given that A is race-free, δ_A is deterministic iff $A^\perp$ is deterministic as an edc with polarity—a condition we call *deterministic for Opponent*. Under the condition that $A^\perp$ is deterministic, as δ_A is a deterministic edc strategy it extends directly to a probabilistic edc strategy with configuration-valuation having constant value 1. Then the probabilstic edc strategy δ_A forms a comonoid with counit $\perp : A \rightarrow\!\!\!\!+ \varnothing$.