

# On Probabilistic Distributed Strategies

Glynn Winskel

Computer Laboratory, University of Cambridge

**Abstract.** In a distributed game we imagine a team Player engaging a team Opponent in a distributed fashion. No longer can we assume that moves of Player and Opponent alternate. Rather the history of a play more naturally takes the form of a partial order of dependency between occurrences of moves. How are we to define strategies within such a game, and how are we to adjoin probability to such a broad class of strategies? The answer yields a surprisingly rich language of probabilistic distributed strategies and the possibility of programming (optimal) probabilistic strategies. Along the way we shall encounter solutions to: the need to mix probability and nondeterminism; the problem of parallel causes in which members of the same team can race to make the same move, and why this leads us to invent a new model for the semantics of distributed systems.

## 1 Introduction

I am working on a theory of distributed games and strategies. The games are distributed in the sense that they involve a team Player in competition with a team Opponent in widely-spread, possibly varying locations. It is no longer sensible to regard the history of the play as a sequence of alternating moves, the case in traditional games. Rather at a reasonable level of abstraction it is sensible to view a history as a partial order showing the dependency of moves on earlier moves. Of course the terms Player and Opponent are open to a variety of interpretations so the intended application areas are very broad.

My own original motivation comes from the wish to generalise domain theory as a basis for denotational semantics. While domain theory provides a beautiful paradigm for formalising and analysing computation it has been increasingly falling short in the burgeoning world of distributed, nondeterministic and probabilistic computation we live in today. In brief, with the wisdom of hindsight, domain theory abstracted from operational concerns too early. So one aim is to repair the “little divide” between operational and denotational semantics. There is also some hope that the common vocabulary and techniques games provide will help bridge the “big divide” in theoretical computer science between the fields of semantics and algorithmics.

One could summarise the enterprise as redoing traditional of games and strategies as a theory based on histories as partial orders of moves. However, the move from sequences to partial orders brings in its wake a lot of technical difficulty and potential for undue complexity unless it’s done artfully. Here we

have been in a good position to take advantage techniques from the early 1980's on a mathematical foundation for work of Hoare and Milner on synchronising processes in categories of models for concurrency [1] and in particular the model and techniques of event structures—an analogue of trees where branches have the form of partial orders [2, 3]. The work on distributed strategies described here could have been done then.

One surprise has been how adeptly distributed strategies fit with probability, at least once a general enough definition of probabilistic event structures was discovered and characterised. It was certainly an advantage to have started with *nondeterministic* strategies [4]. But as we shall see in the move from nondeterministic to probabilistic strategies new phenomena and an unexpected limitation appear.

It has become clear recently that there is a built-in limitation in basing strategies on traditional event structures. Sometimes a distributed strategy can rely on certain “benign races” where, intuitively, several members of team Player may race each other to make a common move. If we are to support benign races in strategies there is a need to work with mathematical structures which support parallel causes—in which an event can be enabled in several compatible ways. This extension seems not to be needed for nondeterministic strategies. It was only revealed in the extension to probabilistic strategies when it was realised that certain intuitively natural probabilistic strategies could not be expressed, with the event structures we were working with. Why, will be explained later.

Though event structures allowing parallel causes have been studied existing structures do not support an operation of hiding central to the composition of strategies. So to some extent we have had go back to the drawing board and invent appropriate structures to support parallel causes and simultaneously a hiding operation. We now know ways to do this. Fortunately the new structures are not so removed from traditional event structures. They involve the objectification of cause, so that one can express *e.g.* that one cause is in parallel with another or in conflict with another, and assign probabilities to causes—see the final section which sketches recent work with Marc de Visme.

## 2 Event structures [3]

The behaviour of distributed games is based on event structures, rather than trees. Instead of regarding a play in a game as a sequence of Player and Opponent moves it is given the structure of a partial order of occurrences of moves.

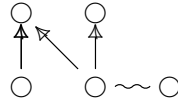
Event structures describe a process, or system, in terms of its possible event occurrences, their causal dependency and consistency. Just as it can be helpful to understand the behaviour of a state-transition diagram in terms of its unfolding to a tree, more detailed models, such as Petri nets, which make explicit the local nature of events and their changes on state, unfold to an event structure [5]. In this sense event structures are a concurrent, or distributed, analogue of trees; though in an event structure the individual ‘branches’ are no longer necessarily sequences but have the shape of a partial order of events.

An *event structure* comprises  $(E, \leq, \text{Con})$ , 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*  $\text{Con}$  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:

$$\begin{aligned} \{e' \mid e' \leq e\} &\text{ is finite for all } e \in E, \\ \{e\} &\in \text{Con for all } e \in E, \\ Y \subseteq X \in \text{Con} &\text{ implies } Y \in \text{Con}, \text{ and} \\ X \in \text{Con} \ \& \ e \leq e' \in X &\text{ implies } X \cup \{e\} \in \text{Con}. \end{aligned}$$

The first axiom says that an event causally depends on only a finite number of events, the second that there are no redundant events, which are in themselves inconsistent. The third axiom expresses the reasonable property that a subset of consistent events is consistent, while the final axiom entails that the  $\leq$ -down-closure of any consistent set of events is also consistent. Two events  $e, e'$  are considered to be *concurrent* if the set  $\{e, e'\}$  is in  $\text{Con}$  and neither event is causally dependent on the other.

It is sometimes convenient to draw event structures. For example,



illustrates an event structure consisting of five events where, in particular, the top event on the left causally depends on the previous occurrences of two concurrent events—the arrows express the causal dependency—one of which is inconsistent with the event on the far right—we have indicated the inconsistency between the two events by a wiggly line.

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 both consistent and down-closed w.r.t. causal dependency:

$$\begin{aligned} \text{Consistent: } X \subseteq x \text{ and } X \text{ is finite} &\text{ implies } X \in \text{Con}, \text{ and} \\ \text{Down-closed: } e' \leq e \in x &\text{ implies } e' \in x. \end{aligned}$$

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

It will be very useful to relate event structures by maps. A *map* of event structures  $f : E \rightarrow E'$  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  $x$ . In particular, when  $f$  is a total function it restricts to a bijection  $x \cong fx$  between any configuration  $x$  and its image  $fx$ .

A map  $f : E \rightarrow E'$  preserves concurrency: if two events in  $E$  are concurrent, then their images if defined are also concurrent. The map also 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  $f$ . In general a map of event structures need not preserve causal dependency; when it does we say it is *rigid*.

In describing distributed games and strategies we shall rely on two properties of maps. Firstly, any map of event structures  $f : E \rightarrow E'$  factors into the composition of a partial map of event structures followed by a total map of event structures

$$E \xrightarrow{p} E_0 \xrightarrow{t} E'$$

in such a way that for any other factorisation  $E \xrightarrow{p_1} E_1 \xrightarrow{t_1} E'$  with  $p_1$  partial and  $t_1$  total, there is a unique (necessarily total) map  $h : E_0 \rightarrow E_1$  such that

$$\begin{array}{ccc} & E_1 & \\ p_1 \nearrow & \downarrow h & \searrow t_1 \\ E & \xrightarrow{p} E_0 \xrightarrow{t} & E' \end{array}$$

commutes. The event structure  $E_0$  is obtained as the “projection,” or restriction, of the relations of causal dependency and consistency of the event structure  $E$  to the events on which  $f$  is defined. We call the total map  $t$  *the defined part* of  $f$ .

Secondly we shall use pullbacks of total maps. Pullbacks are an important construction in representing a process built from two processes sharing a common interface. Maps  $f : A \rightarrow C$  and  $g : B \rightarrow C$  always have pullbacks in the category of event structures, but they are more simple to describe in the case where  $f$  and  $g$  are total, and this is all we shall need:

$$\begin{array}{ccc} & A & \\ \pi_1 \nearrow & & \searrow f \\ P & \triangleright & C \\ \pi_2 \searrow & & \nearrow g \\ & B & \end{array}$$

Roughly, configurations of the pullback  $P$  are matches between configurations of  $A$  and  $B$  which satisfy the causal constraints of both. Precisely, finite configurations of  $P$  correspond to composite bijections

$$\theta : x \cong fx = gy \cong y$$

between finite configurations  $x$  of  $A$  and  $y$  of  $B$  such that  $fx = gy$ , for which the transitive relation generated on  $\theta$  by  $(a, b) \leq (a', b')$  if  $a \leq_A a'$  or  $b \leq_B b'$  has no non-trivial causal loops, and so forms a partial order.

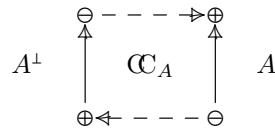
### 3 Distributed games and strategies—the definitions [4]

Often the behaviour of a game is represented by a tree in which the arcs correspond to occurrences of moves by Player or Opponent. Instead we can represent the behaviour of a distributed game more accurately by an event structure together with a polarity function from its events to  $+$  or  $-$  to signify whether they are move occurrences of Player or Opponent, respectively.

A game might generally have winning conditions, a subset of configurations at which Player is deemed to have won, or more generally a payoff function from configurations to the reals.

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 event structure with polarity  $A$  this amounts to reversing the polarities of events to produce the dual  $A^\perp$ . 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  $A^\perp$ . The other operation is a simple parallel composition of games, achieved on event structures with polarity  $A$  and  $B$  very directly by simply juxtaposing them, ensuring a finite subset of events is consistent if its overlaps with the two games are individually consistent, to form  $A \parallel B$ .

As an example of a strategy in a game consider the *copy-cat* strategy for a game  $A$ . This is a strategy in the game  $A^\perp \parallel A$  which, following the spirit of a copy-cat, has Player moves copy the corresponding Opponent moves in the other component. In more detail, the copy-cat strategy  $\mathbb{C}_A$  is obtained by adding extra causal dependencies to  $A^\perp \parallel A$  so that any Player move in either component causally depends on its copy, an Opponent move, in the other component. It can be checked that this generates a partial order of causal dependency. A finite set is taken to be consistent if its down-closure w.r.t. the order generated is consistent in  $A^\perp \parallel A$ . We illustrate the construction on the simple game comprising a Player move causally dependent on a single Opponent move:

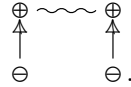


In characterising the configurations of the copy-cat strategy an important partial order on configurations is revealed. Clearly configurations of a game  $A$  are ordered by inclusion  $\sqsubseteq$ . For configurations  $x$  and  $y$ , write  $x \sqsubseteq^- y$  and  $x \sqsubseteq^+ y$  when all the additional events of the inclusion are purely Opponent, respectively, Player moves. A configuration  $x$  of  $\mathbb{C}_A$  is also a configuration of  $A^\perp \parallel A$  and as such splits into two configurations  $x_1$  on the left and  $x_2$  on the right. The extra causal constraints of copy-cat ensure that the configurations of  $\mathbb{C}_A$  are precisely those configurations of  $A^\perp \parallel A$  for which it holds that

$$x_2 \sqsubseteq_A x_1, \text{ defined as } x_2 \sqsupseteq^- y \sqsubseteq^+ x_1,$$

for some configuration  $y$  (necessarily  $x_1 \cap x_2$ ). The relation  $\sqsubseteq_A$  is in fact a partial order on configurations. Increasing in the order  $\sqsubseteq_A$  involves losing Opponent moves and gaining Player moves. Because it generalises the pointwise order of domain theory, initiated by Dana Scott, we call the order  $\sqsubseteq_A$  the *Scott order*.

Strategies in a game are not always obtained by simply adding extra causal dependencies to the game. For example, consider the game comprising two Opponent moves in parallel with a Player move and the strategy (for Player) in which Player make their move if Opponent makes one of theirs. Here the strategy is represented by



We are forced to split the Player move of the game into two moves, each dependent on different Opponent moves, and mutually inconsistent indicated by the wiggly line. For reasons such as this we are led to separate the actual moves of the strategy into an event structure with polarity  $S$  and, in order to track how actual moves correspond to moves in the game, formalise a strategy in a game  $A$  as a total map of event structures

$$\sigma : S \rightarrow A$$

which preserves polarity. (We have met a very special case of this in the copy-cat strategy where the role of  $S$  is taken by  $\mathbb{C}C_A$  and  $\sigma$  acts as the identity function on events.) The event structure  $S$  describes the possibly nondeterministic plays of the strategy. Automatically a state of play of the strategy, represented by a configuration  $x$  of  $S$ , determines a position of the game, a configuration  $\sigma x$  of  $A$ . Directly from the fact that  $\sigma$  is a map, we know that any move in  $\sigma x$  is due to the play of a unique move in  $x$ . The total map  $\sigma : S \rightarrow A$  really just expresses that  $S$  represents a nondeterministic play in the game  $A$ . More is expected of a strategy. For example, consider the game consisting of a Player move concurrent with a move of Opponent and the two total maps indicated:

$$\begin{array}{ccc} \text{(i)} & \begin{array}{ccc} S & \ominus & \rightarrow \oplus \\ \sigma \downarrow & \vdots \downarrow & \vdots \downarrow \\ A & \ominus & \oplus \end{array} & \text{(ii)} & \begin{array}{ccc} S & \oplus & \rightarrow \ominus \\ \sigma \downarrow & \vdots \downarrow & \vdots \downarrow \\ A & \oplus & \ominus \end{array} \end{array}$$

The first map (i) seems reasonable as a strategy; Player awaits the move of Opponent and then makes a move. However, the second map (ii) seems dubious; Player forces Opponent to wait until they have made their move, inappropriate in a distributed strategy.

Instead of guessing, we seek a principled way to determine what further properties a strategy should satisfy. In fact, the further conditions we shall impose on strategies will be precisely those needed to ensure that the copy-cat strategy behaves as an identity w.r.t. the composition of strategies.<sup>1</sup> To do so we adapt

<sup>1</sup> We consider two strategies  $\sigma : S \rightarrow A$  and  $\sigma' : S' \rightarrow A$  to be essentially the same if there is an isomorphism  $f : S \cong S'$  of event structures respecting polarity such that  $\sigma = \sigma' f$ .

an important idea of Conway followed up by Joyal, explaining how to extend the notion of strategy *in* a game to that of a strategy *between* games [6, 7]. The operations of dual and simple parallel composition of games are the key.

A strategy *from* a game  $A$  to a game  $B$  is a strategy in the compound game  $A^\perp \parallel B$ . In particular, copy-cat of a game  $A$  is now seen as a strategy from  $A$  to  $A$ .

In composing two strategies one  $\sigma$  in  $A^\perp \parallel B$  and another  $\tau$  in  $B^\perp \parallel C$  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

$$\begin{array}{ccccc}
 & & A \parallel T & & \\
 & \nearrow^{\pi_2} & & \searrow_{A \parallel \tau} & \\
 T \otimes S & & & & A \parallel B \parallel C \\
 & \searrow_{\pi_1} & & \nearrow_{\sigma \parallel C} & \\
 & & S \parallel C & & 
 \end{array}$$

“synchronises” matching moves of  $S$  and  $T$  over the game  $B$ . But we require a strategy over the game  $A^\perp \parallel C$  and the pullback  $T \otimes S$  has internal moves over the game  $B$ . We achieve this via the projection of  $T \otimes S$  to its moves over  $A$  and  $C$ . We make use of the partial map from  $A \parallel B \parallel C$  to  $A \parallel C$  which acts as the identity function on  $A$  and  $C$  and is undefined on  $B$ . The composite partial map

$$\begin{array}{ccccc}
 & & A \parallel T & & \\
 & \nearrow^{\pi_2} & & \searrow_{A \parallel \tau} & \\
 T \otimes S & & & & A \parallel B \parallel C \rightarrow A \parallel C \\
 & \searrow_{\pi_1} & & \nearrow_{\sigma \parallel C} & \\
 & & S \parallel C & & 
 \end{array}$$

has defined part, yielding the composition

$$\tau \circ \sigma : T \otimes S \rightarrow A^\perp \parallel C$$

once we reinstate polarities. The composition of strategies  $\tau \circ \sigma$  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.

Two further conditions, *receptivity* and *innocence*, are demanded of strategies. The conditions are necessary and sufficient to ensure that copy-cat strategies behave as identities w.r.t. composition [4]. Receptivity expresses that any Opponent move allowed from a reachable position of the game is present as a move in the strategy. In more detail,  $\sigma : S \rightarrow A$  is receptive when for any configurations  $x$  of  $S$  if  $\sigma x$  extends purely by Opponent events to a configuration  $y$  then there is a unique extension of  $x$  to a configuration  $x'$  of  $S$  such that  $\sigma x' = y$ . Innocence says a strategy can only adjoin new causal dependencies of the form  $\ominus \rightarrow \oplus$ , where Player awaits moves of Opponent, beyond those already inherited from the game.

The literature is often concerned with deterministic strategies, in which Player has at most one consistent response to Opponent. We can broaden the concept of deterministic strategy to distributed strategies by taking such a strategy to be *deterministic* if consistent moves of Opponent entail consistent moves of Player—see [4, 8]. Formally, we say an event structure with polarity is *deterministic* if any finite down-closed subset is consistent when its Opponent events form a consistent subset. In general the copy-cat strategy for a game need not be deterministic. Copy-cat is however deterministic precisely for games which are *race-free*, *i.e.* such that at any configuration, if both a move of Player and a move of Opponent are possible then they may occur together: if whenever  $x$ ,  $x \cup \{\oplus\}$  and  $x \cup \{\ominus\}$  are configurations of  $A$ , where the events  $\oplus$  and  $\ominus$  have the opposing polarities indicated, then  $x \cup \{\oplus, \ominus\}$  is a configuration. Deterministic distributed strategies coincide with the *receptive* ingenuous strategies of Mellies and Mimram [9].

Just as strategies generalise relations, deterministic strategies generalise functions. In fact, multirelations and functions are recovered as strategies, respectively deterministic strategies, in the special case where the games are composed solely of Player moves with trivial causal dependency and where only the empty set and singletons are consistent.

As would be hoped the concepts of strategy and deterministic strategy espoused here reduce to the expected traditional notions on traditional games. There have also been pleasant surprises. In the extreme case where games comprise purely Player moves, strategies correspond precisely to the ‘stable spans’ used in giving semantics to nondeterministic dataflow [10], and in the deterministic subcase one recovers exactly the *stable domain theory* of Gérard Berry [11].

We now turn to how a strategy might be made probabilistic. We first address an appropriately general way to adjoin probability to event structures.

## 4 Probabilistic event structures [12]

The extension of distributed strategies to probabilistic strategies required a new general definition of probabilistic event structure. A probabilistic event structure essentially comprises an event structure together with a continuous valuation on the Scott-open sets of its domain of configurations.<sup>2</sup> The continuous valuation assigns a probability to each open set and can then be extended to a probability measure on the Borel sets [13]. 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 basic open sets, generated by single

<sup>2</sup> 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*)  $w(\emptyset) = 0$ ; (*monotone*)  $U \subseteq V \implies w(U) \leq w(V)$ ; (*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. The idea:  $w(U)$  is the probability of a result in open set  $U$ .



finite configurations; for each finite configuration this specifies the probability of obtaining a result which extends the finite configuration. Such valuations on configurations 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.

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)  $v(\emptyset) = 1$  and has  
 (non-ve drop)  $d_v[y; x_1, \dots, x_n] \geq 0$  when  $y \subseteq x_1, \dots, x_n$  for finite configurations  $y, x_1, \dots, x_n$  of  $E$ ,

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

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

—the index  $I$  ranges over nonempty  $I \subseteq \{1, \dots, n\}$  such that the union  $\bigcup_{i \in I} x_i$  is a configuration. The “drop”  $d_v[y; x_1, \dots, x_n]$  gives the probability of the result being a configuration which includes the configuration  $y$  and does not include any of the configurations  $x_1, \dots, x_n$ .<sup>3</sup>

## 5 Probabilistic strategies [15]

The above has prepared the ground for a general definition of distributed probabilistic strategies, based on event structures. One hurdle is that in a strategy it is impossible to know the probabilities assigned by Opponent. We need to address the problem—notorious in domain theory—of how to mix probability (which Player attributes to their moves) and nondeterminism (ensuing from Player’s ignorance of the probabilities assigned to Opponent moves). A probabilistic strategy in a game  $A$ , presented as a race-free event structure with polarity, is a strategy  $\sigma : S \rightarrow A$  in which we endow  $S$  with probability, while taking account of the fact that in a strategy Player can’t be aware of the probabilities assigned by Opponent. We do this by extending the notion of configuration valuation so that: causal independence between Player and Opponent moves entails their probabilistic independence, or equivalently, so probabilistic dependence of

<sup>3</sup> Samy Abbes has pointed out that the same “drop condition” appears in early work of the Russian mathematician V.A.Rohlin [14](as relation (6) of Section 3, p.7). Its rediscovery in the context of event structures was motivated by the need to tie probability to the occurrences of events; it is sufficient to check the ‘drop condition’ for  $y \subset x_1, \dots, x_n$ , in which the configurations  $x_i$  extend  $y$  with a single event.

Player on Opponent moves will presuppose their causal dependence (the effect of the condition of “ $\pm$ -independence” below); the “drop condition” only applies to moves of Player. Precisely, a *configuration valuation* is now a function  $v$ , from finite configurations of  $S$  to the unit interval, which is

(*normalized*)  $v(\emptyset) = 1$ , has  
 ( $\pm$ -*independence*)  $v(x) = v(y)$  when  $x \sqsubseteq^- y$  for finite configurations  $x, y$  of  $S$ ,  
 and satisfies the  
 (*+ve drop condition*)  $d_v[y; x_1, \dots, x_n] \geq 0$  when  $y \sqsubseteq^+ x_1, \dots, x_n$  for finite configurations of  $S$ .

One can think of the value  $v(x)$ , where  $x$  is a finite configurations of  $S$ , as the probability of obtaining a result which extends  $x$  conditional on the Opponent moves in  $x$ .

We return to the point that “ $\pm$ -independence” expresses that causal independence between Player and Opponent moves entails their probabilistic independence. Consider two moves,  $\oplus$  of Player and  $\ominus$  of Opponent able to occur independently, *i.e.* concurrently, at some finite configuration  $x$ , taking it to the configuration  $x \cup \{\oplus, \ominus\}$ . There are intermediate configurations  $x \cup \{\oplus\}$  and  $x \cup \{\ominus\}$  associated with just one additional move. The condition of “ $\pm$ -independence” ensures  $v(x \cup \{\oplus, \ominus\}) = v(x \cup \{\oplus\})$ , *i.e.* the probability of  $\oplus$  with  $\ominus$  is the same as the probability of  $\oplus$  at configuration  $x$ . At  $x$  the probability of the Player move conditional on the Opponent move equals the probability of the Player move—the moves are probabilistically independent.

## 5.1 A bicategory of probabilistic strategies

Probabilistic strategies compose. Assume probabilistic strategies  $\sigma : S \rightarrow A^\perp \parallel B$  with configuration valuation  $v_S$  and  $\tau : T \rightarrow B^\perp \parallel C$  with configuration valuation  $v_T$ . Recall how the composition  $\tau \circ \sigma$  is obtained via pullback, to synchronise the strategies over common moves, followed by projection, to hide the synchronisations.

Given  $z$  a finite configuration of the pullback  $T \otimes S$  its image  $\pi_1 z$  under the projection  $\pi_1$  is a finite configuration of  $S \parallel C$ ; taking its left component we obtain  $(\pi_1 z)_1$ , a finite configuration of  $S$ . Similarly, taking the right component of the image  $\pi_2 z$  we obtain a finite configuration  $(\pi_2 z)_2$  of  $T$ . It can be shown that defining  $v(z) = v_S((\pi_1 z)_1) \times v_T((\pi_2 z)_2)$  for  $z$  a finite configuration of  $T \otimes S$  satisfies the conditions of a configuration valuation (with the proviso that we treat synchronisation and Player events alike in the drop condition). In the proof ‘drop functions’ come into their own. A finite configuration  $x$  of  $T \otimes S$ , after hiding, is a subset of  $T \otimes S$  so we can form its down-closure there to obtain  $[x]$ , a finite configuration of  $T \otimes S$ . The assignment of value  $v([x])$  to  $x$  a finite configuration of  $T \otimes S$  yields a configuration valuation to associate with the composition  $\tau \circ \sigma$ .

Above, notice in the special case where  $\sigma : S \rightarrow B$  and  $\tau : T \rightarrow B^\perp$ , *i.e.* of a strategy and a counter-strategy in the game  $B$ , that the resulting probabilistic play is captured by  $T \otimes S$ , which is now a probabilistic event structure.<sup>4</sup>

Because we restrict to race-free games, copy-cat strategies are deterministic ensuring that the assignment of one to each finite configuration of copy-cat is a configuration valuation; this provides us with identities w.r.t. composition.

We don't have a category however, as the laws for categories are only true up to isomorphism. Technically we have a bicategory of games and probabilistic strategies in which the objects are race-free games and the arrows are probabilistic strategies. The 2-cells, the maps between strategies, require some explanation.<sup>5</sup> Without the presence of probability it is sensible to take a 2-cell between two strategies  $\sigma : S \rightarrow A^\perp \parallel B$  and  $\sigma' : S' \rightarrow A^\perp \parallel B$  to be a map  $f : S \rightarrow S'$  making

$$\begin{array}{ccc}
 S & \xrightarrow{f} & S' \\
 & \searrow \sigma & \downarrow \sigma' \\
 & & A^\perp \parallel B
 \end{array}$$

commute. However, in the situation where the strategies are probabilistic, when  $\sigma$  is accompanied by a configuration valuation  $v$  and  $\sigma'$  by configuration valuation  $v'$ , we need a further constraint to relate probabilities. Normally probability distributions can be “pushed forward” across measurable functions. But configuration valuations don't correspond to probability distributions in the presence of Opponent moves and in general we can't push forward the configuration valuation  $v$  of  $S$  to a configuration valuation  $fv$  of  $S'$ . We can however do so when  $f$  is rigid: then defining

$$(fv)(y) =_{\text{def}} \sum \{v(x) \mid fx = y\},$$

for  $y \in \mathcal{C}(S')$ , yields a configuration valuation  $fv$  of  $S'$  —the *push-forward* of  $v$ . So finally we constrain 2-cells between probabilistic strategies, from  $\sigma$  with  $v$  to  $\sigma'$  with  $v'$ , to those rigid maps  $f$  for which  $\sigma = \sigma'f$  and the push-forward  $fv$  is pointwise less than or equal to  $v'$ .

The vertical composition of 2-cells is the usual composition of maps. Horizontal composition is given by the composition of strategies  $\odot$  (which extends to a functor on 2-cells via the universal properties of pullback and factorisation used in its definition).

2-cells include rigid embeddings preserving the value assigned by configuration valuations.<sup>6</sup> Amongst these are those 2-cells in which the rigid embedding is an inclusion—providing a very useful order for defining probabilistic strategies

<sup>4</sup> The use of “schedulers to resolve the probability or nondeterminism” in earlier work is subsumed by that of probabilistic and deterministic counter-strategies. Deterministic strategies coincide with those with assignment one to each finite configuration.

<sup>5</sup> Their treatment in [15] is slapdash.

<sup>6</sup> One way to define a rigid embedding is as a rigid map whose function is injective and reflects consistency.

recursively. Let  $\sigma : S \rightarrow A^\perp \parallel B$  with configuration valuation  $v$  and  $\sigma' : S' \rightarrow A^\perp \parallel B$  with  $v'$  be two probabilistic strategies. Define  $\sigma \preceq \sigma'$  when  $S \subseteq S'$  and the associated inclusion map is a rigid embedding and a 2-cell for which  $v(x) = v'(x)$  for all finite configurations of  $S$ . This enables us to exploit old techniques to define strategies recursively: the substructure order on event structures, of which we have an example, forms a “large complete partial order” on which continuous operations possess least fixed points—see [2, 3].

## 5.2 Extensions: payoff and imperfect information

We can add *payoff* to a game as a function from its configurations to the real numbers [16, 15]. For such quantitative games, determinacy is expressed in terms of the game possessing a *value*, a form of minimax property. The interest is now focussed on *optimal* strategies which achieve the value of the game. In games of *imperfect information* some moves are masked, or inaccessible, and strategies with dependencies on unseen moves are ruled out. It is straightforward to extend probabilistic distributed games with payoff and imperfect information in way that respects the operations of distributed games and strategies [17]. *Blackwell games* [18], of central importance in logic and computer science, become a special case of probabilistic distributed games of imperfect information with payoff [15].

## 6 Constructions on probabilistic strategies [19]

There is a richness of constructions in the world of distributed strategies and games. The language of games and strategies that ensues is largely stable under the addition of probability and extra features such as imperfect information and payoff. Though for instance we shall need to restrict to race-free games in order to have identities w.r.t. the composition of probabilistic strategies.

In the language for probabilistic strategies, race-free games  $A, B, C, \dots$  will play the role of types. There are operations on games of forming the dual  $A^\perp$ , simple parallel composition  $A \parallel B$ , sum  $\sum_{i \in I} A_i$  as well as recursively-defined games—the latter rest on well-established techniques [2] and will be ignored here. The operation of sum of games is similar to that of simple parallel composition but where now moves in different components are made inconsistent; we restrict its use to those cases in which it results in a game which is race-free.

Terms have typing judgements:

$$x_1 : A_1, \dots, x_m : A_m \vdash t \dashv y_1 : B_1, \dots, y_n : B_n ,$$

where all the variables are distinct, interpreted as a probabilistic strategy from the game  $\vec{A} = A_1 \parallel \dots \parallel A_m$  to the game  $\vec{B} = B_1 \parallel \dots \parallel B_n$ . We can think of the term  $t$  as a box with input and output wires for the variables:



The idea is that  $t$  denotes a probabilistic strategy  $S \rightarrow \vec{A}^\perp \parallel \vec{B}$  with configuration valuation  $v$ . The term  $t$  describes witnesses, finite configurations of  $S$ , to a relation between finite configurations  $\vec{x}$  of  $\vec{A}$  and  $\vec{y}$  of  $\vec{B}$ , together with their probability conditional on the Opponent moves involved.

**Duality** The duality, that a probabilistic strategy from  $A$  to  $B$  can equally well be seen as a probabilistic strategy from  $B^\perp$  to  $A^\perp$ , is caught by the rules:

$$\frac{\Gamma, x : A \vdash t \dashv \Delta}{\Gamma \vdash t \dashv x : A^\perp, \Delta} \quad \frac{\Gamma \vdash t \dashv x : A, \Delta}{\Gamma, x : A^\perp \vdash t \dashv \Delta}$$

**Composition** The composition of probabilistic strategies is described in the rule

$$\frac{\Gamma \vdash t \dashv \Delta \quad \Delta \vdash u \dashv \text{H}}{\Gamma \vdash \exists \Delta. [t \parallel u] \dashv \text{H}}$$

which, in the picture of strategies as boxes, joins the output wires of one strategy to input wires of the other.

**Probabilistic sum** For  $I$  countable and a sub-probability distribution  $p_i, i \in I$ , we can form the probabilistic sum of strategies of the same type:

$$\frac{\Gamma \vdash t_i \dashv \Delta \quad i \in I}{\Gamma \vdash \sum_{i \in I} p_i t_i \dashv \Delta.}$$

In the probabilistic sum of strategies, of the same type, the strategies are glued together on their initial Opponent moves (to maintain receptivity) and only commit to a component with the occurrence of a Player move, from which component being determined by the distribution  $p_i, i \in I$ . We use  $\perp$  for the empty probabilistic sum, when the rule above specialises to

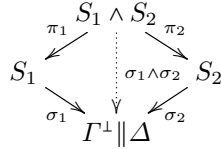
$$\Gamma \vdash \perp \dashv \Delta,$$

which denotes the minimum strategy in the game  $\Gamma^\perp \parallel \Delta$ —it comprises the initial segment of the game  $\Gamma^\perp \parallel \Delta$  consisting of its initial Opponent events.

**Conjoining two strategies** The pullback of a strategy across a map of event structures is itself a strategy [15]. We can use the pullback of one strategy against another to conjoin two probabilistic strategies of the same type:

$$\frac{\Gamma \vdash t_1 \dashv \Delta \quad \Gamma \vdash t_2 \dashv \Delta}{\Gamma \vdash t_1 \wedge t_2 \dashv \Delta}$$

Such a strategy acts as the two component strategies agree to act jointly. In the case where  $t_1$  and  $t_2$  denote the probabilistic strategies  $\sigma_1 : S_1 \rightarrow \Gamma^\perp \parallel \Delta$  with configuration valuation  $v_1$  and  $\sigma_2 : S_2 \rightarrow \Gamma^\perp \parallel \Delta$  with  $v_2$  the strategy  $t_1 \wedge t_2$  denotes the pullback



with configuration valuation  $x \mapsto v_1(\pi_1 x) \times v_2(\pi_2 x)$  for  $x \in \mathcal{C}(S_1 \wedge S_2)$ .

**Copy-cat terms** Copy-cat terms are a powerful way to lift maps or relations expressed in terms of maps to strategies. Along with duplication they introduce new “causal wiring.” Copy-cat terms have the form

$$x : A \vdash gy \sqsubseteq_C fx \dashv y : B,$$

where  $f : A \rightarrow C$  and  $g : B \rightarrow C$  are maps of event structures preserving polarity. (In fact,  $f$  and  $g$  may even be “affine” maps, which don’t necessarily preserve empty configurations, provided  $g\emptyset \sqsubseteq_C f\emptyset$ —see [19].) This denotes a deterministic strategy—so a probabilistic strategy with configuration valuation constantly one—provided  $f$  reflects  $--$ compatibility and  $g$  reflects  $+$ -compatibility. The map  $g$  reflects  $+$ -compatibility if whenever  $x \sqsubseteq^+ x_1$  and  $x \sqsubseteq^+ x_2$  in the configurations of  $B$  and  $fx_1 \cup fx_2$  is consistent, so a configuration, then so is  $x_1 \cup x_2$ . The meaning of  $f$  reflecting  $--$ compatibility is defined analogously.

A term for copy-cat arises as a special case,

$$x : A \vdash y \sqsubseteq_A x \dashv y : A,$$

as do terms for the  $j$ th injection into and  $j$ th projection out of a sum  $\Sigma_{i \in I} A_i$  w.r.t. its component  $A_j$ ,

$$x : A_j \vdash y \sqsubseteq_{\Sigma_{i \in I} A_i} jx \dashv y : \Sigma_{i \in I} A_i$$

and

$$x : \Sigma_{i \in I} A_i \vdash jy \sqsubseteq_{\Sigma_{i \in I} A_i} x \dashv y : A_j,$$

as well as terms which split or join ‘wires’ to or from a game  $A \parallel B$ .

In particular, a map  $f : A \rightarrow B$  of games which reflects  $--$ compatibility lifts to a deterministic strategy  $f_! : A \dashv \Rightarrow B$ :

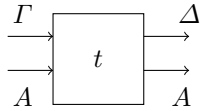
$$x : A \vdash y \sqsubseteq_B fx \dashv y : B.$$

A map  $f : A \rightarrow B$  which reflects  $+$ -compatibility lifts to a deterministic strategy  $f^* : B \dashv \Rightarrow A$ :

$$y : B \vdash fx \sqsubseteq_B y \dashv x : A.$$

The construction  $f^* \circledast t$  denotes the pullback of a strategy  $t$  in  $B$  across the map  $f : A \rightarrow B$ . It can introduce extra events and dependencies in the strategy. It subsumes the operations of prefixing by an initial Player or Opponent move on games and strategies.

**Trace** A probabilistic *trace*, or feedback, operation is another consequence of such “wiring.” Given a probabilistic strategy  $\Gamma, x : A \vdash t \dashv y : A, \Delta$  represented by the diagram



we obtain

$$\Gamma, \Delta^\perp \vdash t \dashv x : A^\perp, y : A$$

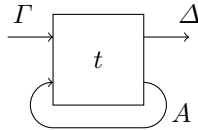
which post-composed with the term

$$x : A^\perp, y : A \vdash x \sqsubseteq_A y \dashv,$$

denoting the copy-cat strategy  $\gamma_{A^\perp}$ , yields

$$\Gamma \vdash \exists x : A^\perp, y : A. [t \parallel x \sqsubseteq_A y] \dashv \Delta,$$

representing its trace:



The composition introduces causal links from the Player moves of  $y : A$  to the Opponent moves of  $x : A$ , and from the Player moves of  $x : A$  to the Opponent moves of  $y : A$ —these are the usual links of copy-cat  $\gamma_{A^\perp}$  as seen from the left of the turnstyle. If we ignore probabilities, this trace coincides with the feedback operation which has been used in the semantics of nondeterministic dataflow (where only games comprising solely Player moves are needed) [10].

**Duplication** Duplications of arguments is essential if we are to support the recursive definition of strategies. We duplicate arguments through a probabilistic strategy  $\delta_A : A \dashv\rightarrow A \parallel A$ . Intuitively it behaves like the copy-cat strategy but where a Player move in the left component may choose to copy from either of the two components on the right. In general the technical definition is involved, even without probability—see [19]. The introduction of probability begins to reveal a limitation within probabilistic strategies as we have defined them, a point we will follow up on in the next section. We can see the issue in the second of two simple examples. The first is that of  $\delta_A$  in the case where the game  $A$  consists of a single Player move  $\oplus$ . Then,  $\delta_A$  is the deterministic strategy



in which the configuration valuation assigns one to all finite configurations —we have omitted the obvious map to the game  $A^\perp \parallel A \parallel A$ . In the second example, assume  $A$  consists of a single Opponent move  $\ominus$ . Now  $\delta_A$  is no longer deterministic and takes the form



and the strategy is forced to choose probabilistically between reacting to the upper or lower move of Opponent in order to satisfy the drop condition of its

configuration valuation. Given the symmetry of the situation, in this case any configuration containing a Player move is assigned value a half by the configuration valuation associated with  $\delta_A$ . (In the definition of the probabilistic duplication for general  $A$  the configuration valuation is distributed uniformly over the different ways Player can copy Opponent moves.) But this is odd: in the second example, if the Opponent makes only one move there is a 50% chance that Player will not react to it! There are mathematical consequences too. In the absence of probability  $\delta_A$  forms a comonoid with counit  $\perp : A \multimap \emptyset$ . However, as a probabilistic strategy  $\delta_A$  is no longer a comonoid—it fails associativity. It is hard to see an alternative definition of a probabilistic duplication strategy within the limitations of the event structures we have been using. We shall return to duplication, and a simpler treatment through a broadening of event structures in the next section.

**Recursion** Once we have duplication strategy we can treat recursion. Recall that 2-cells, the maps between probabilistic strategies, include the approximation order  $\preceq$  between strategies. The order forms a ‘large complete partial order’ with a bottom element the minimum strategy  $\perp$ . Given  $x : A, \Gamma \vdash t \multimap y : A$ , the term  $\Gamma \vdash \mu x : A. t \multimap y : A$  denotes the  $\preceq$ -least fixed point amongst probabilistic strategies  $X : \Gamma \multimap A$  of the  $\preceq$ -continuous operation  $F(X) = t \odot (\text{id}_\Gamma \parallel X) \odot \delta_\Gamma$ . (With one exception,  $F$  is built out of operations which it’s been shown can be defined concretely in such a way that they are  $\preceq$ -continuous; the one exception which requires separate treatment is the ‘new’ operation of projection, used to hide synchronisations.) With probability, as  $\delta_\Gamma$  is no longer a comonoid not all the “usual” laws of recursion will hold, though the unfolding law will hold by definition.

## 7 A limitation

One limitation that is not seen when working with purely nondeterministic strategies has revealed itself when strategies are made probabilistic. The simple event structures on which we have based games and strategies do not support “parallel causes” and this has the consequence that certain informal but intuitively convincing strategies are not expressible. We met this in the previous section in our treatment of a probabilistic duplication strategy  $\delta_A : A \multimap A \parallel A$ .

Probabilistic strategies, as presented, do not cope with stochastic behaviour such as races as in the game

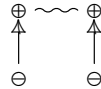
$$\ominus \rightsquigarrow \oplus .$$

To do such we would expect to have to equip events in the strategy with stochastic rates (which isn’t hard to do if synchronisation events are not hidden). So this is to be expected. But at present probabilistic strategies do not cope with benign Player-Player races either! Consider the game

$$\begin{array}{c} \oplus \\ \ominus \quad \ominus \end{array}$$



where Player would like a strategy in which they play a move iff Opponent plays one of theirs. We might stipulate that Player wins if a play of any  $\ominus$  is accompanied by the play of  $\oplus$  and *vice versa*. Intuitively a winning strategy would be got by assigning watchers (in the team Player) for each  $\ominus$  who on seeing their  $\ominus$  race to play  $\oplus$ . This strategy should win with certainty against any counter-strategy: no matter how Opponent plays one or both of their moves at least one of the watchers will report this with the Player move. But we cannot express this with event structures. The best we can do is a probabilistic strategy



with configuration valuation assigning  $1/2$  to configurations containing either Player move and  $1$  otherwise. Against a counter-strategy with Opponent playing one of their two moves with probability  $1/2$  this strategy only wins half the time. In fact, the strategy together with the counter-strategy form a Nash equilibrium when a winning configuration for Player is assigned payoff  $+1$  and a loss  $-1$ . This strategy really is the best we can do presently in that it is optimal amongst those expressible using the simple event structures of Section 2.

## 8 A solution

If we are to be able to express the intuitive strategy which wins with certainty we need to develop distributed probabilistic strategies which allow such parallel causes as in ‘general event structures’  $(E, \vdash, \text{Con})$  which permit *e.g.* two distinct compatible causes  $X \vdash e$  and  $Y \vdash e$  (see [3]). In the informal strategy described in the previous section both Opponent moves would individually enable the Player move, with all events being consistent. But it can be shown that general event structures do not support an appropriate operation of hiding. Nor is it clear how within general event structures one could express a variant of the strategy above, in which the two watchers succeed in reporting the Player move with different probabilities.

It is informative to see why general event structures are not closed under hiding. The following describes a general event structure.

Events:  $a, b, c, d$  and  $e$ .

Enablings: (1)  $b, c \vdash e$  and (2)  $d \vdash e$ , with all events other than  $e$  being enabled by the empty set.

Consistency: all subsets are consistent unless they contain the events  $a$  and  $b$ ; in other words, 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 (or 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  $\{c\} \subseteq \{c, e\}$  and  $\{c\} \subseteq \{a, c\}$  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 a general event structure it would have to also be a configuration.)

The first general event structure can be built out of the composition *without hiding* of strategies described by general event structures, one from a game  $A$  to a game  $B$  and the other from  $B$  to  $C$ ; the second structure, not a general event structure, would arise when hiding the events over the intermediate game  $B$ .

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 is one of inconsistency between complete enablings rather than events.

Marc de Visme and I have explored the space of possibilities and discovered a refinement of event structures into which general event structures embed, which supports hiding, and provides a basis on which to develop probabilistic distributed strategies with disjunctive and parallel causes. One is led to introduce structures in which we *objectify* cause: a minimal causal enabling is no longer an instance of a relation but an object that realises that instance (*cf.* a proof in contrast to a judgement of theorem-hood). This is in order to express inconsistency between complete enablings, inexpressible as inconsistencies on events, that can arise when hiding.

An event structure with disjunctive causes (an edc) is a structure

$$(P, \leq, \text{Con}_P, \equiv)$$

where  $(P, \leq, \text{Con}_P)$  satisfies the axioms of a event structure and  $\equiv$  is an equivalence relation on  $P$  such that

$$\forall p_1, p_2 \leq p. p_1 \equiv p_2 \implies p_1 = p_2.$$

The events of  $P$  represent *prime causes* while the  $\equiv$ -equivalence classes of  $P$  represent *disjunctive events*:  $p$  in  $P$  is a prime cause of the 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 causally independent. A *configuration* of the edc is a configuration of  $(P, \leq, \text{Con}_P)$ . An edc dissociates the two roles of enabling and atomic action conflated in the events of an event structures. The elements of  $P$  are to be thought of as complete

minimal enablings and the equivalence classes as atomic actions representing the occurrence of at least one prime cause.

When the equivalence relation  $\equiv$  of an edc is the identity it is essentially an event structure. This view is reinforced in our choice of maps. A map from  $(P, \leq_P, \text{Con}_P, \equiv_P)$  to  $(Q, \leq_Q, \text{Con}_Q, \equiv_Q)$  is a partial function  $f : P \rightarrow Q$  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)  $\forall p_1, p_2 \in x. f(p_1) \equiv_Q f(p_2) \implies p_1 \equiv_P p_2$ .

Edc's support a hiding operation along the same lines as event structures. There is an adjunction expressing the sense in which general event structures embed within edc's. There is also an adjunction (in fact a coreflection) from event structures to edc's which helps give a tight connection between strategies based on event structures and their generalisation to edc's. Probability extends straightforwardly to strategies based on edc's. The work is recent and in the process of being written up [20]. We conclude by presenting the *deterministic* strategy in the game

$$\oplus$$

$$\ominus \quad \ominus$$

in which Player makes a move iff Opponent does:

$$\begin{array}{ccc} \oplus & = & \oplus \\ \uparrow & & \uparrow \\ \ominus & & \ominus \end{array}$$

(In the deterministic case each watcher succeeds with certainty. We can also represent the situation where one watcher succeeds with probability  $p \in [0, 1]$  and the other with probability  $q \in [0, 1]$  through the obvious configuration valuation.) The same strategy serves as the duplication strategy for the game comprising a single Opponent event. This indicates how within the broader framework of edc's there are deterministic duplication strategies  $\delta_A : A \multimap A \parallel A$  in which a Player move in the left component is alerted in parallel by a corresponding Opponent move in either of the two components on the right. The fact that the duplication strategies are now deterministic obviates the difficulties we encountered earlier: duplication now forms a comonoid and we recover the usual laws for recursive definitions of strategies.

**Acknowledgements**

Thanks to Samy Abbes, Nathan Bowler, Simon Castellan, Pierre Clairambault, Marcelo Fiore, Mai Gehrke, Julian Gutierrez, Jonathan Hayman, Martin Hyland, Marc Lasson, Silvain Rideau, Daniele Varacca and Marc de Visme for helpful discussions. The concluding section is based on recent joint work with Marc de Visme while on his internship from ENS Paris. The support of Advanced Grant ECSYM of the European Research Council is acknowledged with gratitude.

## References

1. G. Winskel and M. Nielsen, *Handbook of Logic in Computer Science (Vol. 4)*. Oxford University Press, 1995, ch. Models for Concurrency, pp. 1–148.
2. G. Winskel, “Event structure semantics for CCS and related languages,” in *ICALP’82*, ser. LNCS, vol. 140. Springer, 1982.
3. —, “Event structures,” in *Advances in Petri Nets*, ser. Lecture Notes in Computer Science, vol. 255. Springer, 1986, pp. 325–392.
4. S. Rideau and G. Winskel, “Concurrent strategies,” in *LICS 2011*. IEEE Computer Society, 2011.
5. M. Nielsen, G. Plotkin, and G. Winskel, “Petri nets, event structures and domains,” *Theoretical Computer Science*, vol. 13, pp. 85–108, 1981.
6. J. Conway, *On Numbers and Games*. Wellesley, MA: A K Peters, 2000.
7. A. Joyal, “Remarques sur la théorie des jeux à deux personnes,” *Gazette des sciences mathématiques du Québec*, 1(4), 1997.
8. G. Winskel, “Deterministic concurrent strategies,” *Formal Asp. Comput.*, vol. 24, no. 4-6, pp. 647–660, 2012.
9. P.-A. Melliès and S. Mimram, “Asynchronous games: Innocence without alternation,” in *CONCUR*, ser. Lecture Notes in Computer Science, L. Caires and V. T. Vasconcelos, Eds., vol. 4703. Springer, 2007, pp. 395–411.
10. L. Saunders-Evans and G. Winskel, “Event structure spans for nondeterministic dataflow,” *Electr. Notes Theor. Comput. Sci.* 175(3): 109-129, 2007.
11. G. Berry, “Stable models of typed lambda-calculi,” in *ICALP*, ser. Lecture Notes in Computer Science, vol. 62. Springer, 1978, pp. 72–89.
12. G. Winskel, “Probabilistic and quantum event structures,” in *Festschrift for Prakash Panangaden*, ser. Lecture Notes in Computer Science, vol. 8464. Springer, 2014, pp. 476–497.
13. C. Jones and G. Plotkin, “A probabilistic powerdomain of valuations,” in *LICS ’89*. IEEE Computer Society, 1989.
14. V. A. Rohlin, “On the fundamental ideas of measure theory,” *Amer. Math. Soc. Translation 1952*, no. 71, p. 55.
15. G. Winskel, “Distributed probabilistic and quantum strategies,” *Electr. Notes Theor. Comput. Sci.* 298: 403-425, 2013.
16. P. Clairambault and G. Winskel, “On concurrent games with payoff,” *Electr. Notes Theor. Comput. Sci.* 298: 71-92, 2013.
17. G. Winskel, “Winning, losing and drawing in concurrent games with perfect or imperfect information,” in *Festschrift for Dexter Kozen*, ser. LNCS, vol. 7230. Springer, 2012.
18. D. Martin, “The determinacy of Blackwell games,” *Journal of Symbolic Logic*, vol. 63, no. 4, pp. 1565–1581, 1998.
19. S. Castellan, J. Hayman, M. Lasson, and G. Winskel, “Strategies as concurrent processes,” *Electr. Notes Theor. Comput. Sci.*, vol. 308, pp. 87–107, 2014.
20. M. de Visme and G. Winskel, “Strategies with parallel causes,” *Draft*, 2015.