

Distributed Games and Strategies

Glynn Winskel, Computer Laboratory, University of Cambridge

Abstract—A summary of work on distributed games and strategies done within the first three years of the ERC project ECSYM is presented.

I. MOTIVATION

My current interest is to develop a mathematical theory of *distributed* games in which a Player (or a team of players) can interact and compete against an Opponent (or a team of opponents) in a highly distributed fashion, without, for instance, enforcing that their moves occur in a sequential fashion, or need to alternate. Although phrased in terms of ‘Player’ and ‘Opponent’ the dichotomy Player *vs.* Opponent can stand for Process *vs.* Environment, Prover *vs.* Disprover, or Ally *vs.* Enemy. These alternatives indicate the wide range of potential applications in computer science, logic, and beyond.

The original motivation for developing distributed games was as a framework to represent and analyse the behaviour of computation, which can be distributed and highly interactive. It had become clear that many aspects of computation did not fit within traditional approaches. There was a need to tackle anomalies such as non-deterministic dataflow, which lay outside the usual techniques, and repair the divide between the two main approaches of denotational and operational semantics. Roughly, in representing computation, the role of Player is to represent that part of the system which is controllable and could be programmed, while that of Opponent stands for the unforeseeable and uncontrollable environment.

Strategies are as potentially fundamental as relations and functions. It is surely because of our limited mental capacity, and not because of its unimportance, that the mathematical concept of strategy has been uncovered relatively late. It is hard to think about the successive contingencies involved in playing a game in the same way that is hard to think about interacting processes. Developing strategies in the extra generality demanded by distributed interaction has enabled us to harness computer-science expertise in structure and distributed computation in their understanding and formalisation. The extra generality has revealed more clearly their essential mathematical nature, new insights, and a mathematical robustness to the concept of strategy.

As the work has developed it has become increasingly clear that strategies are at the crossroads of several areas both semantic and algorithmic, and beyond computer science and logic, in other sciences, where perhaps the structural emphasis, so part and parcel to computer science, has been missing. In this note I have tried to summarise the work on distributed games and strategies done over the last three years, and indicate how it is to be developed across a range of areas.

Outline The presentation begins with an introduction to event structures (Section II), the fundamental model on which we base the behaviour of distributed games; plays in games will take the form of partial orders of moves. We then develop the definition of distributed strategies (Section III), adopting as the guiding principle that certain ‘copy-cat’ strategies should behave as identities w.r.t. a composition of strategies; the definition of composition is driven by two fundamental operations on games, those of parallel composition and dual. Distributed games and strategies have a very rich structure, leading to a language for programming strategies (Section IV). The language is robust w.r.t. extensions by winning conditions (Section V), to probabilistic strategies (Section VI), quantum strategies (Section VII), games with payoff functions (Section VIII) and of imperfect information (Section IX) and games with symmetry (Section X). We conclude with remarks on extensions, present limitations, and ambitions for the future (Sections XI and XII).

II. EVENT STRUCTURES [1]

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 [2]. 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.

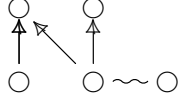
An *event structure* comprises (E, \leq, Con) , consisting of a set E of *events* (really event occurrences) which are partially ordered by \leq , the *causal dependency relation*, and a nonempty *consistency relation* Con consisting of finite subsets of E . The relation $e' \leq e$ expresses that event 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:

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

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 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:

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

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

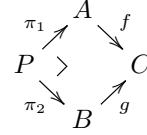
$$E \xrightarrow{p} E_0 \xrightarrow{t} E'$$

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

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

commutes. The event structure E_0 is obtained as the “projection” 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:



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 θ 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.

III. DISTRIBUTED GAMES AND STRATEGIES — THE DEFINITIONS [3]

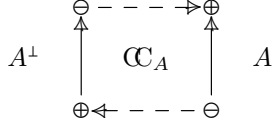
Often the behaviour of a game is represented by a tree in which the arcs correspond to occurrences of moves by Player or Opponent. Instead we can represent the behaviour of a distributed game more accurately by an event structure together with a polarity function from its events to $+$ or $-$ to signify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

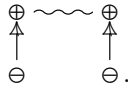


In characterising the configurations of the copy-cat strategy an important partial order on configurations is revealed. Clearly configurations of a game A are ordered by inclusion \subseteq . For configurations x and y , write $x \subseteq^- y$ and $x \subseteq^+ y$ when all the additional events of the inclusion are purely Opponent, respectively, Player moves. A configuration x of $\mathbb{C}\mathbb{C}_A$ 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 $\mathbb{C}\mathbb{C}_A$ are precisely those configurations of $A^\perp \parallel A$ for which it holds that

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

for some configuration y . 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 $\mathbb{C}\mathbb{C}_A$ and σ 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 σx of A . Directly from the fact that σ is a map, we know that any move in σx 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:

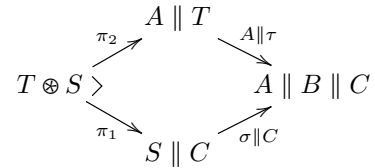


The first map (i) seems reasonable as a strategy; Player awaits the move of Opponent and then makes a move. However, the second map (ii) seems dubious; Player forces Opponent to wait until they have made their move, inappropriate in a distributed strategy.

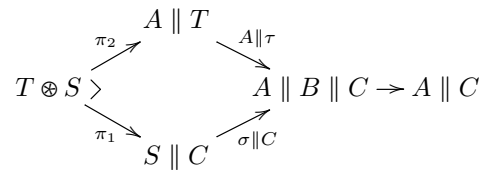
Instead of guessing, we seek a principled way to determine what further properties a strategy should satisfy. In fact, the further conditions we shall impose on strategies will be precisely those needed to ensure that the copy-cat strategy behaves as an identity w.r.t. the composition of strategies.¹ To do so we adapt 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 [4], [5]. 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 σ in $A^\perp \parallel B$ and another τ 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



“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



has defined part, yielding the composition

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

¹We 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 $\sigma = \sigma' f$.

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 [3]. 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 σx 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 Scott order makes a reappearance in a technically-important characterisation of strategies $\sigma : S \rightarrow A$ in a game A as certain discrete fibrations from the finite configurations of S and to those of A , under the Scott order. (Discrete fibrations correspond to presheaves, a generalisation of characteristic function to categories, in which ‘true’ and ‘false’ are replaced, respectively, by the set of ways to realise truth and the empty set.) The association of a discrete fibration with a strategy respects composition and explains how to regard strategies as special forms of profunctor [6]. (Profunctors are the generalisation of relations, usually between sets, to categories and have been proposed as a generalised domain theory [7], [8].) Considering what is lost in the transition from strategies to profunctors has led to an extension of distributed games to ‘rooted’ factorisation systems, in which strategies may be continuous processes [6].

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 [3], [9]. 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. Deterministic distributed strategies coincide with the *receptive* ingenuous strategies of Melliès and Mimram [10].

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 [11], and in the deterministic subcase one recovers exactly the *stable domain theory* of Gérard Berry [12].

We have not said much about what it means for a strategy to be winning or optimal, or how a strategy might be made probabilistic. This will come. But first we point out the richness of constructions in the world of distributed strategies and games. The language of games and strategies that ensues will be largely stable under the addition of extra features described in the subsequent sections.

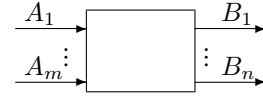
IV. A LANGUAGE FOR STRATEGIES [13]

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 [14] 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.

Terms denoting strategies 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 strategy from the game $A_1 \parallel \dots \parallel A_m$ to the game $B_1 \parallel \dots \parallel B_n$. We can think of the term t as a box with input and output wires for the variables:



Duality The duality of strategies, that a strategy from A to B can equally well be seen as a 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 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.

Nondeterministic sum We can form the nondeterministic sum of strategies of the same type:

$$\frac{\Gamma \vdash t_i \dashv \Delta \quad i \in I}{\Gamma \vdash \bigsqcup_{i \in I} t_i \dashv \Delta}$$

The construction is like that of the sum of games but where the initial Opponent events are identified to maintain receptivity. The empty nondeterministic sum denotes the minimum strategy in the game $\Gamma^\perp \parallel \Delta$.

Pullback The pullback of a strategy across a map of event structures is itself a strategy [15]. In particular, we can form the pullback of two 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.

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.” A general copy-cat term takes the form

$$\frac{\Gamma \vdash p' : C \quad \Delta \vdash p : C}{\Gamma \vdash p \in_C p' \dashv \Delta} \quad p[\emptyset] \in_C p'[\emptyset]$$

based on expressions p, p' for configurations of C and their typings; it is stipulated in the formation of a copy-cat term that the configurations denoted by p and p' are in the Scott order initially, when their variables are assigned the empty configuration. The configurations of the strategy denoted by the copy-cat term are built from configurations satisfying the relation $p \in_C p'$ by adding causal links in the manner of copy-cat. A term for copy-cat arises as a special case,

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

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 ,

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

and

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

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

A map $f : A \rightarrow B$ of games both lifts to a strategy $f! : A \dashv\rightarrow B$ described by

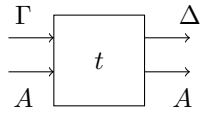
$$x : A \vdash y \in_B fx \dashv y : B$$

and to a strategy $f^* : B \dashv\rightarrow A$,

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

Forming the composition $f^* \circ t$ pulls back a strategy t in B across the map $f : A \rightarrow B$ to a strategy in A . It can introduce extra events and dependencies into the strategy. It subsumes operations of prefixing a strategy by an initial Player or Opponent move.

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



we obtain

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

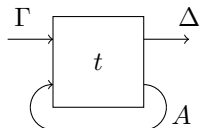
which post-composed with the term

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

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

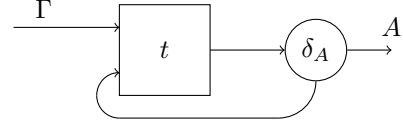
$$\Gamma \vdash \exists x : A^\perp, y : A. [t \parallel x \in_A y] \dashv \Delta,$$

representing its trace:

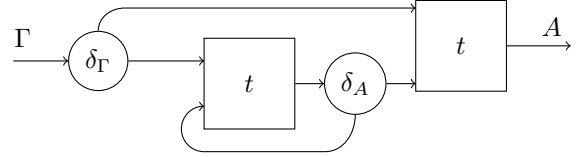


The composition introduces causal links from the Player moves of $y : A$ to the Opponent moves of $x : A$, and from the Player moves of $x : A$ to the Opponent moves of $y : A$ — these are the usual links of copy-cat γ_{A^\perp} as seen from the left of the turnstyle. This trace coincides with the feedback operation which has been used in the semantics of nondeterministic dataflow (where only games comprising solely Player moves are needed) [11].

Recursive definitions can be achieved from trace with the help of duplication terms, based on a strategy δ_A from a game A to $A \parallel A$, roughly, got by joining two copy-cat strategies together:



Provided the body t of the recursion respects δ_A the diagram above unfolds in the way expected of recursion, to:



In fact, recursive definitions can be made more generally, without the use of trace, by exploiting old techniques for defining event structures recursively. The substructure order \leq on event structures forms a “large complete partial order,” continuous operations on which possess least fixed points — see [14], [1]. Given $x : A, \Gamma \vdash t \dashv y : A$, the term $\Gamma \vdash \mu x : A. t \dashv y : A$ denotes the \leq -least fixed point amongst strategies $X : \Gamma \dashv\rightarrow A$ of the \leq -continuous operation $F(X) = t \circ (\text{id}_\Gamma \parallel X) \circ \delta_\Gamma$; here $\sigma \leq \sigma'$ between two strategies $\sigma : S \rightarrow \Gamma^\perp \parallel A$ and $\sigma' : S' \rightarrow \Gamma^\perp \parallel A$ signifies $S \leq S'$ and that the associated inclusion map $i : S \rightarrow S'$ makes $\sigma = \sigma' i$.

V. WINNING CONDITIONS [16], [17]

Winning conditions of a game A are given by specifying a subset of *winning configurations* W . An outcome in W is a win for Player; for simplicity assume that any other outcome is a win for Opponent. A strategy (for Player) is regarded as *winning* if it always prescribes moves for Player to end up in a winning configuration, no matter what the activity or inactivity of Opponent. Formally, say a configuration is $+$ -maximal if no additional Player moves can occur from it. Say a strategy $\sigma : S \rightarrow A$ is *winning* if σx is in W for all $+$ -maximal configurations x of S . This can be shown equivalent to all plays of σ against (deterministic) counter-strategies of Opponent result in a win for Player.

As the dual of a game A with winning conditions W we again reverse the roles of Player and Opponent to get A^\perp and take its winning conditions to be the set-theoretic complement of W . In a simple parallel composition of games with winning conditions, $A \parallel B$, Player wins if they win in either component. With these extensions we can take a winning strategy from a game A to a game B , where both games have winning conditions, to be a winning strategy in the game $A^\perp \parallel B$. The

choices ensure that the composition of winning strategies is winning. In order to have identities w.r.t. composition, we need a condition on games (in fact quite a general condition, implied by race-freeness) to guarantee that copy-cat strategies are winning.

Often questions can be reformulated in terms of the existence of a winning strategy for Player or Opponent. For this it is important to know whether there is a winning strategy for one of the players — the issue of *determinacy* of the game. A famous result is Martin’s theorem, at the boundary of the power of set theory, which shows that sequential games are determined if their winning conditions form a Borel set [18]. Distributed games with races need not be determined even though their winning configurations form a Borel set. However distributed games are determined provided that they are race-free and satisfy a structural condition, “bounded concurrency” [17]. Bounded concurrency expresses that no move of one of the players can be concurrent with infinitely many moves of the other — a condition trivially satisfied when *e.g.* all plays are finite, the games are sequential, or the games have rounds where simple choices are made (usual in traditional accounts). Like race-freeness, bounded concurrency is necessary for the Borel determinacy of distributed games.

VI. PROBABILISTIC EVENT STRUCTURES AND STRATEGIES [15]

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.² The continuous valuation assigns a probability to each open set and can then be extended to a probability measure on the Borel sets [19]. 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 finite configurations; for each finite configuration this specifies the probability of obtaining a result which extends the finite configuration. 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.

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

²A *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 .

(*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 .³

This prepares 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. 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 Player on Opponent moves will presuppose their causal dependence (the effect of the condition of “ \pm -independence” below); the “drop condition” only applies to \pm 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 .

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.

Probabilistic strategies compose — in the proof ‘drop functions’ come into their own — with probabilistic copy-cat strategies as identities because we restrict to race-free games.

³Samy Abbes has pointed out that the same “drop condition” appears in early work of the Russian mathematician V.A.Rohlin [20](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 $y \sqsubseteq^- x_1, \dots, x_n$, in which the configurations x_i extend y with a single event.

The result of a play between Player and Opponent in a game will be a probabilistic event structure.⁴

VII. QUANTUM EVENT STRUCTURES AND STRATEGIES [15], [21]

In a *quantum event structure* the events of an event structure are interpreted as unitary or projection operators in a Hilbert space in such a way that concurrent events are interpreted by commuting operators. Unitary operators are associated with events of preparation, such as a change of coordinates with which to make a measurement or a time period over which the system is allowed to evolve undisturbed. Projection operators are associated with events of elementary tests. A configuration of the event structure is thought of as a distributed quantum experiment; it describes which events of preparation and tests to perform and their partial order of dependency. The whole event structure can be thought of as a nondeterministic quantum experiment, and gives us the extra latitude to define a probabilistic quantum experiment as quantum event structure which also carries the structure of a probabilistic event structure. A quantum event structure comes with an initial state in the form of a density operator.

In more detail, a *quantum event structure* (over a Hilbert space with countable basis) comprises an initial state given by a density operator ρ and an event structure (E, \leq, Con) with an assignment Q_e of projection or unitary operators to events e of E such that if two events e_1 and e_2 are concurrent then the operators Q_{e_1} and Q_{e_2} commute. Each finite configuration, x of E , determines an operator A_x got as the composition

$$Q_{e_n} Q_{e_{n-1}} \cdots Q_{e_2} Q_{e_1}$$

for some serialization e_1, e_2, \dots, e_n of x in which $\{e_1, \dots, e_i\}$ is a configuration for all $i \leq n$. The operator is well-defined as any two serializations of x are obtainable, one from the other, by successively interchanging concurrent events. A quantum event structure assigns an intrinsic weight $v(x) =_{\text{def}} \text{Tr}(A_x^\dagger A_x \rho)$, the trace of the operator $A_x^\dagger A_x \rho$, to each finite configuration x ; in the case where ρ corresponds to a pure state $|\psi\rangle$ the weight will be the square of the norm of $A_x|\psi\rangle$. This does not make the *whole* event structure into a probabilistic event structure, but it does do so locally: for any configuration w of E , the function v is a configuration-valuation on the event structure obtained by restricting E to the events w .

Quantum theory is often described as a contextual theory, in that it is only sensible to consider outcomes w.r.t. a specified measurement context [22]. In a quantum event structure configurations assume the role of measurement contexts; w.r.t. a measurement context expressed as a configuration, the sub-configurations constitute the possible outcomes. This gives a non-traditional take on the consistent-histories approach to quantum theory, which provides decoherence conditions on histories to pick out those subfamilies of histories over which

⁴The 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.

it is meaningful to place a probability distribution; the approach via quantum event structures bypasses the decoherence conditions usually invoked [23].

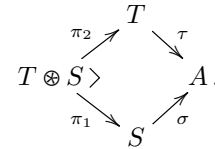
A *quantum strategy* is taken to be a distributed probabilistic strategy on a game which also carries the structure of a quantum event structure, so that moves perform operators on a Hilbert space. In a quantum game Player and Opponent interact to jointly create a probabilistic distributed experiment on a quantum system [24]. Extensions with pay-off and to games of imperfect information are dealt with as below.

VIII. PAYOFF [25]

We can add *payoff* to a game A as a function X from configurations to the real numbers. We can extend the operations of dual and simple parallel composition of games to games with payoff. As the payoff X_{A^\perp} of the dual game A^\perp we take the negative of the payoff X_A of A , $X_{A^\perp}(x) = -X_A(x)$ on configurations x — this implicitly makes the games zero-sum. As the payoff of $A \parallel B$ we take sums of the payoffs, $X_{A \parallel B}(x) = X_A(x_1) + X_B(x_2)$ on a configuration x of $A \parallel B$, with components x_1 in A and x_2 in B .

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. How we proceed to associate a value with a distributed game slightly differs according to whether strategies are assumed probabilistic.

In the probabilistic case, assume that payoff function X on the game A is Borel measurable. Given a probabilistic strategy in A , so a strategy $\sigma : S \rightarrow A$ and a configuration-valuation v_S for S , and a probabilistic counter-strategy in A , so a strategy $\tau : T \rightarrow A^\perp$ with a configuration-valuation v_T for T , we obtain their composition before hiding as the pullback



The pullback is associated with the map $f =_{\text{def}} \sigma \pi_1 = \tau \pi_2 : T \otimes S \rightarrow A$. The event structure $T \otimes S$ comes equipped with a configuration-valuation $v(x) = v_S(\pi_1 x) \times v_T(\pi_2 x)$ on its finite configurations x . This determines a Borel probability measure μ_v on all the configurations of $T \otimes S$. The *expected payoff* is obtained as the Lebesgue integral

$$\mathbf{E}_{\sigma, \tau}(X) = \int X(fx) d\mu_v(x)$$

over the configurations x of $T \otimes S$. Define

$$v(A) =_{\text{def}} \text{Sup}_{\sigma : A} \text{Inf}_{\tau : A^\perp} \mathbf{E}_{\sigma, \tau}(X),$$

where $\sigma : A$ signifies a probabilistic strategy in A and $\tau : A^\perp$ a probabilistic strategy counter-strategy. The game is *determined* w.r.t. probabilistic strategies if $v(A) = -v(A^\perp)$, with $v(A)$ then being called the *value* of the game; since the order of Sup and Inf are reversed in $v(A)$ and $-v(A^\perp)$, determinacy amounts to a minimax property. For a determined game, an *optimal* strategy is one $\sigma : A$ for which

$$v(A) = \text{Inf}_{\tau : A^\perp} \mathbf{E}_{\sigma, \tau}(X).$$

After [17], it is reasonable to conjecture that any bounded concurrent, race-free distributed game with measurable payoff function is determined w.r.t. probabilistic strategies — though this has not yet been proved.

Without probability, for distributed strategies which are in general nondeterministic, we instead consider both the *optimistic* $v^\uparrow(A)$ and *pessimistic* value $v^\downarrow(A)$ of a game A . The optimistic value stems from taking the result of the interaction of a strategy and a counter-strategy to be the supremum over the values resulting from all maximal plays (as the strategies are nondeterministic there may be many maximal plays). The pessimistic value is defined analogously using infimum. We say a game is determined with value $v(A)$ if

$$v(A) = v^\uparrow(A) = v^\downarrow(A) = -v^\downarrow(A^\perp) = -v^\uparrow(A^\perp).$$

In this case an optimal strategy is one with pessimistic value that of the game. It is to be expected that any bounded concurrent, race-free distributed game with Borel measurable payoff function is determined, though this has so far only been proved for well-founded games, in which all configurations are finite [25]. In *op. cit.* it is shown that optimal strategies compose, so fit within the language of Section IV. This makes a start on a general *structural game theory* in which games and optimal strategies are built up compositionally.

IX. IMPERFECT INFORMATION [26], [15]

As they stand the games so far described are games of *perfect information*. 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 distributed games to games with imperfect information in way that respects the operations of distributed games and strategies [26] and does not disturb the addition of extra features such as probability. A fixed preorder of *levels* (Λ, \leq) is pre-supposed. The levels are to be thought of as levels of access, or permission. Moves in games and strategies are to respect levels: a move is only permitted to causally depend on moves at equal or lower levels; it is as if from a level only moves of equal or lower level can be seen. A distributed game of imperfect information, a Λ -*game*, comprises a game A with a *level function* $l : A \rightarrow \Lambda$ such that if $a \leq_A a'$ then $l(a) \leq l(a')$ for all moves a, a' in A . A Λ -*strategy* in the Λ -game is a strategy $\sigma : S \rightarrow A$ for which if $s \leq_S s'$ then $l\sigma(s) \leq l\sigma(s')$ for all s, s' in S . One interpretation of Λ , pertinent to the treatment of quantum strategies, is as space-time with $\lambda \leq \lambda'$ meaning there is a causal curve from λ to λ' .

In particular, *Blackwell games* [27], of central importance in logic and computer science, become a special case of probabilistic Λ -games with payoff. Blackwell games are games of imperfect information for which an appropriate choice of Λ is the infinite event structure:



A Blackwell game is given by A , a race-free distributed game with payoff X , for which there is a (necessarily unique)

polarity-preserving rigid map from A to Λ —this map becomes the level function. Moves in A occur in rounds comprising a choice of move for Opponent and a choice of move for Player made independently. Traditionally, in Blackwell games a strategy (for Player) is a ‘total’ Λ -strategy in such a Λ -game — strategies are restricted to those assigning probability distributions at each round. In fact, the existing literature is most often concerned with *total* strategies which always progress, which we can express very generally for distributed strategies by insisting there is zero probability of ending at a configuration from which a Player move is possible.

Based on our experience and the insights of Martin [27], the strongest determinacy result we can hope for in distributed games with imperfect information is the conjecture that any finite-width, race-free distributed game with imperfect information, and Borel measurable payoff function is determined w.r.t. probabilistic strategies. This result would go a considerable way to subsuming the theory of Blackwell games within distributed games.

X. GAMES WITH SYMMETRY [28]

There are several reasons to consider symmetry in games, situations where distinct plays are essentially similar to one another. Symmetry can help in the analysis of games, by for instance reducing the number of cases to consider. Symmetry can also help compensate for the overly-concrete nature of event structures in representing games: many useful operations on games which are not algebraic operations become so *up to symmetry*. In particular, through symmetry we can support operations on types in the form of *monads* — well-known from functional programming — and this leads to richer type systems and, in particular, the ability to support backtracking in games. Although backtracking is not allowed in many games it is important in logic, in representing assertions by games and proofs as strategies between them, when it is very common for the proof to use the same assertion several times. Similarly, in computer science, where types are often represented by games and programs by strategies, a program may generally reuse its input. Backtracking is supported through a monad $!A$ on a game A where $!A$ consists of many copies of moves in the original game A ; while copies of the same move are distinct from each other they are essentially similar, a similarity we can express through symmetry in games.

The treatment of symmetry in games stems from earlier work on symmetry in event structures [29] and makes use of a general method of open maps for defining the equivalence of bisimulation in a variety of models [30]. Briefly, a symmetry in an event structure E is expressed as a bisimulation equivalence, given as a span of open maps $l, r : \tilde{E} \rightarrow E$. The finite configurations of \tilde{E} correspond to bijections between finite configurations of E expressing that one configuration is similar to the other. Together the finite configurations of \tilde{E} correspond to an *isomorphism family* comprising a non-empty family of bijections $\theta : x \cong_E y$ between pairs of finite configurations of E such that:

- (i) for all identities on finite configurations $\text{id}_x : x \cong_E x$; if $\theta : x \cong_E y$, then the inverse $\theta^{-1} : y \cong_E x$; and if $\theta : x \cong_E y$ and $\varphi : y \cong_E z$, then their composition $\varphi\theta : x \cong_E z$.

(ii) for $\theta : x \cong_E y$ and finite configurations $x' \subseteq x$ there is a (necessarily unique) finite configuration y' of E with $y' \subseteq y$ such that the restriction $\theta' : x' \cong_E y'$.

(iii) for $\theta : x \cong_E y$ whenever $x \subseteq x'$ with x' a finite configuration, there is an extension of θ to θ' so $\theta' : x' \cong_E y'$ for some (not necessarily unique) configuration y' with $y \subseteq y'$. (Because of (ii) the bijections in the isomorphism family respect the partial order of causal dependency on configurations inherited from E .)

A total map $f : A \rightarrow B$ between event structures with symmetry *preserves symmetry* when $x \cong_A^{\theta} y$ implies $fx \cong_B^{\tilde{f}\theta} fy$, where $\tilde{f}\theta$ is the composite bijection $fx \cong x \cong_A^{\theta} y \cong fy$. While two total maps $f, g : A \rightarrow B$ preserving symmetry, are *equivalent up to symmetry*, written $f \sim g$, if $fx \cong_B^{\phi_x} gx$ for all finite configurations x of A , where ϕ_x is the composite bijection $fx \cong x \cong g$.

With the addition of symmetry event structures no longer have pullbacks. But they do have pseudo pullbacks, a form of pullback up to symmetry. The *pseudo pullback* of total maps $f : A \rightarrow C, g : B \rightarrow C$ comprises

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

where $f\pi_1 \sim g\pi_2$ with the further property that for any object D and maps $p_1 : D \rightarrow A$ and $p_2 : D \rightarrow B$ such that $fp_1 \sim gp_2$, there is a unique map $h : D \rightarrow P$ such that $p_1 = \pi_1 h$ and $p_2 = \pi_2 h$. The pseudo pullback is defined up to isomorphism. Concretely, finite configurations of P correspond to the composite bijections

$$\theta : x \cong fx \cong_C^{\varphi} gy \cong y$$

between finite configurations x of A and y of B such that $fx \cong_C^{\varphi} gy$ for which the transitive relation generated on θ by $(a, b) \leq (a', b')$ if $a \leq_A a'$ or $b \leq_B b'$ forms a partial order.

A particular pseudo pullback is obtained as

$$\begin{array}{ccc} & A & \\ l_A \nearrow & & \searrow \text{id}_A \\ \tilde{A} & \sim & A \\ r_A \searrow & & \nearrow \text{id}_A \\ & A & \end{array}$$

It recovers \tilde{A} , the symmetry on A , but as an event structure with symmetry so *itself equipped with symmetry* $\tilde{\tilde{A}}$. This pseudo pullback is that associated with a *path object* of homotopy generally written A^I , where I stands for (a generalization of) the unit interval: asserting $\theta : x \cong_A y$, that a bijection between two configurations is in the isomorphism family of A , is analogous to specifying a path from x to y . There are also *cylinder objects*. In fact event structures with symmetry have the structure of a homotopy category. Just as in homotopy theory one considers operations up to homotopy it is sensible to consider operations within event structures *up to symmetry*.

A *game with symmetry* is represented by an event structure with polarity and symmetry (henceforth an e.p.s.). The operations of dual and simple parallel composition of games extend

to games with symmetry. The symmetry of A^\perp coincides with that of A , while the symmetry of $A \parallel B$ is built as $\tilde{A} \parallel \tilde{B}$.

A strategy in a game with symmetry A will be a total map $\sigma : S \rightarrow A$ of event structures with symmetry. We regard two strategies $\sigma : S \rightarrow A$ and $\sigma' : S' \rightarrow A$ as *equivalent* when there are maps $f : S \rightarrow S'$ and $g : S' \rightarrow S$ such that $\sigma = \sigma' f$ and $\sigma' = \sigma g$ with $gf \sim \text{id}_S$ and $fg \sim \text{id}_{S'}$. Following the now familiar pattern a strategy between games with symmetry, from A to B , will be a strategy in $A^\perp \parallel B$.

The addition of polarity and symmetry brings a new richness to the configurations of an event structure. The Scott order becomes a *Scott category*, where now maps between configurations are obtained as compositions

$$y \supseteq^- y' \cong_A^{\theta} x' \subseteq^+ x.$$

This fact drives the definition of copy-cat strategy for a game with symmetry A . The configurations of $\mathbb{C}\mathbb{C}_A$, in copy-cat $\gamma_A : \mathbb{C}\mathbb{C}_A \rightarrow A^\perp \parallel A$, correspond to instances of maps in the Scott category, to those configurations x of $A^\perp \parallel A$ for which

$$x_2 \supseteq^- x'_2 \cong_A^{\theta} x'_1 \subseteq^+ x_1,$$

for some instance $x'_2 \cong_A^{\theta} x'_1$ of the isomorphism family of A . The symmetry in copy-cat is constructed in a similar way, as

$$\widetilde{\mathbb{C}\mathbb{C}_A} = \mathbb{C}\mathbb{C}_{\tilde{A}}.$$

The definition of the composition $\tau \circ \sigma : T \otimes S \rightarrow A^\perp \parallel C$ of strategies $\sigma : S \rightarrow A^\perp \parallel B$ and $\tau : T \rightarrow B^\perp \parallel C$ between games with symmetry is exactly analogous to that without symmetry but using pseudo pullbacks in place of pullbacks. The symmetry on $T \otimes S$, composition before hiding, satisfies

$$\widetilde{T \otimes S} = \tilde{T} \otimes \tilde{S}.$$

The symmetry on $T \otimes S$, composition after hiding, is a little more complicated than $\tilde{T} \otimes \tilde{S}$ but only because the latter fails to satisfy joint monicity required of an equivalence. (For this and other reasons it would be an interesting exercise to redevelop symmetry on games by relaxing the present definition, in which a symmetry is expressed as a bisimulation equivalence, to one in which the requirement of joint monicity is dropped.)

Again we can ask for conditions on strategies that exactly ensure that copy-cat behaves as identity, more precisely that composition with copy-cat yields an equivalent strategy. The necessary and sufficient conditions are strengthenings of receptivity and innocence, seen earlier, to take account of symmetry: in addition, the strategy should be *strong-receptive*, *i.e.* receptivity should also respect symmetry, and *saturated*, *i.e.* behave identically over symmetric parts of the game — see [28] for the precise definitions.

The language of strategies extends to games with symmetry (pseudo pullbacks replace pullbacks) and allows extra types and terms because now in the presence of symmetry there are many \sim -monads (technically pseudo monads that satisfy the monad laws up to symmetry). One simple example is the \sim -monad constructing $!A$, infinitely many copies of a game A , one for each natural number, all made similar to each other through the appropriate symmetry. The unit of the \sim -monad essentially injects A into one chosen copy — which

one does not matter, up to symmetry — and the multiplication essentially flattens the “array” of copies, one for each pair of natural numbers, to the “row” $!A$. This \sim -monad is central to AJM games [31].

Under fairly general conditions \sim -monads on event structures with symmetry lift to monads on \sim -strategies, expressible within a slight extension of the language of strategies. Given such a monad T its unit and multiplication lift to strategies via copy-cat terms and the action of T itself on a strategy is essentially to enclose it within a “ T -box” — an additional construction in the language. Because of the duality of strategies, a \sim -monad on strategies can also be viewed as a \sim -comonad. In particular there are comonads to allow back-tracking of the form required in traditional AJM and HO games [31], [32], [28].

XI. EXTENSIONS AND LIMITATIONS

The story so far has been quite well-rounded, I believe. In this section I’ll try to gather the loose ends of which we’re presently aware. In particular, I’ll try to give an idea of an obstacle on which we are currently working, the extension of strategies with “parallel causes.”

First though, note that the language for strategies simultaneously supports both a mathematical and an operational semantics [13]. The operational semantics, presented as rules for transitions, gives a view of strategies as processes. It comes at the cost of taking internal, previously hidden, events seriously. Once one does so, questions as to what strategies may and must do arise. Fortunately it can be shown that as regards the may-and-must behaviour of strategies one can circumvent internal events and in their place work with a notion of maximal configuration (called “stopping configuration”) divorced from order-theoretically maximal configurations. In Section V $+$ -maximal configurations played a key role; their role can be taken over straightforwardly by stopping configurations.

One limitation that is not seen when working with purely nondeterministic strategies has revealed itself when strategies are made probabilistic. The simple event structures of Section II 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.

Probabilistic strategies, as presented, do not cope with stochastic behaviour, *e.g.* races as in the game

$$\ominus \sim \oplus .$$

To do such we would expect to have to equip events in the strategy with stochastic rates. So this is to be expected — and isn’t hard to do if internal events are not hidden. 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 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 prime event structures. The best we can do is a probabilistic strategy

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

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 II.

If we are to be able to express the intuitive strategy which wins with certainty we need to develop distributed probabilistic strategies to allow such parallel causes as in ‘general event structures’ (E, \vdash, Con) which allow *e.g.* two distinct compatible causes $X \vdash e$ and $Y \vdash e$ (see [1]). In this specific strategy 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 in which the two watchers succeed in reporting the Player move with different probabilities.

We are currently working on two ways to extend probabilistic strategies with parallel causes in such a way that they still support an appropriate operation of hiding; we hope and expect the two ways will converge. One way is roughly to extend general event structures so that consistency is between different enablings rather than events; an example shows an extension of this nature is needed if it is to support hiding. The other way involves a monad $?$ up to symmetry constructed so strategies $\sigma : S \rightarrow ?A$ can express several parallel ways in which a Player move can be enabled.

XII. THE FUTURE

The over-abstraction of traditional denotational semantics has largely removed it from algorithmic concerns. This has led to an artificial division between the semantics of programming languages (concerning their formalisation, algebra and proof of correctness) and algorithmics (concerning the efficiency and classification of computational problems), whereas in reality both have need of techniques of the other. The division reflects a fundamental tension between the approaches of semantics and algorithmics. Whereas semantics seeks denotations and equivalences sensible within a very broad range of contexts (generally that allowed by the programming languages of interest) algorithmics is often concerned with optimal methods within very specialised contexts (often algorithms to perform some specific task). The seemingly-conflicting approaches

of semantics and algorithmics can be reconciled provided a semantics can support contextual reasoning and is refined enough to express algorithmic concerns.

Distributed games give a faithful operational account of algorithms while having a rich algebraic structure. As one example, a basic tool for the efficient verification of systems is that of binary decision diagrams (bdd's); they are precisely sequential strategies between games representing the booleans. Another example is that important equivalences between programs can sometimes be settled automatically by exploiting the game semantics of programs, together with automata representations of strategies [33]. At a fundamental level, deep questions in complexity theory and logic can often be reformulated in terms of games. The idea of a strategy *from* one game *to* another was used by Conway in developing his “surreal numbers” as strengths of games [4]. Once one has a winning/optimal strategy from a game *A* to a game *B* by precomposition it provides a way to convert a winning/optimal strategy of *A* to a winning/optimal strategy of *B*; in this way it reduces finding a winning strategy of *B* to the problem of finding one for *A*. Similar reductions are plentiful in algorithmics and logic. Perhaps they could be systematized through distributed games, which can potentially support a range of reductions in the form of strategies from a game $!A$ to a game $?A$ w.r.t monads $!$ and $?$ up to symmetry. At the very least the determinacy result of Section IX would provide a new facility in expressing computational problems in terms of the existence of a strategy — presently, often problems are artificially forced into a formulation using Blackwell games. Nor at a general level should we underestimate the power of games in providing a vocabulary common to both semantics and algorithmics.

The fundamental nature and range of distributed games and strategies means that they have the potential to make revolutionary changes to the way we think about computing systems, and through these, other subjects too. But this will not come easily. It requires significant demonstration of their use in programming, verification, logic and, most importantly, in the beneficial cross-over between the algebraic and algorithmic worlds that they simultaneously inhabit.

Acknowledgements

This paper is dedicated to Pierre-Louis Curien on his 60th birthday. It's a great pleasure for me to acknowledge Pierre-Louis's inspiration and friendship over the years, inspiration and friendship that go back to our thesis work and first meetings in Sophia Antipolis and Paris in 1979.

Thanks to Samy Abbes, Nathan Bowler, Simon Castellan, Pierre Clairambault, Pierre-Louis Curien, Marcelo Fiore, Mai Gehrke, Julian Gutierrez, Jonathan Hayman, Martin Hyland, Alex Katovsky, Marc Lasson, Paul-André Melliès, Samuel Mimram, Gordon Plotkin, Silvain Rideau and Sam Staton for helpful discussions. The support of Advanced Grant ECSYM of the European Research Council is acknowledged with gratitude.

REFERENCES

- [1] G. Winskel, “Event structures,” in *Advances in Petri Nets*, ser. Lecture Notes in Computer Science, vol. 255. Springer, 1986, pp. 325–392.
- [2] M. Nielsen, G. Plotkin, and G. Winskel, “Petri nets, event structures and domains,” *Theoretical Computer Science*, vol. 13, pp. 85–108, 1981.
- [3] S. Rideau and G. Winskel, “Concurrent strategies,” in *LICS 2011*. IEEE Computer Society, 2011.
- [4] J. Conway, *On Numbers and Games*. Wellesley, MA: A K Peters, 2000.
- [5] A. Joyal, “Remarques sur la théorie des jeux à deux personnes,” *Gazette des sciences mathématiques du Québec*, 1(4), 1997.
- [6] G. Winskel, “Strategies as profunctors,” in *FOSSACS 2013*, ser. Lecture Notes in Computer Science. Springer, 2013.
- [7] M. Hyland, “Some reasons for generalising domain theory,” *Mathematical Structures in Computer Science*, vol. 20, no. 2, pp. 239–265, 2010.
- [8] G. L. Cattani and G. Winskel, “Profunctors, open maps and bisimulation,” *Mathematical Structures in Computer Science*, vol. 15, no. 3, pp. 553–614, 2005.
- [9] G. Winskel, “Deterministic concurrent strategies,” *Formal Asp. Comput.*, vol. 24, no. 4-6, pp. 647–660, 2012.
- [10] 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.
- [11] L. Saunders-Evans and G. Winskel, “Event structure spans for nondeterministic dataflow,” *Electr. Notes Theor. Comput. Sci.* 175(3): 109-129, 2007.
- [12] G. Berry, “Stable models of typed lambda-calculi,” in *ICALP*, ser. Lecture Notes in Computer Science, vol. 62. Springer, 1978, pp. 72–89.
- [13] S. Castellan, J. Hayman, M. Lasson, and G. Winskel, “Strategies as concurrent processes,” *Electr. Notes Theor. Comput. Sci.*, vol. 308, pp. 87–107, 2014.
- [14] G. Winskel, “Event structure semantics for CCS and related languages,” in *ICALP'82*, ser. LNCS, vol. 140. Springer, 1982.
- [15] —, “Distributed probabilistic and quantum strategies,” *Electr. Notes Theor. Comput. Sci.* 298: 403-425, 2013.
- [16] P. Clairambault, J. Gutierrez, and G. Winskel, “The winning ways of concurrent games,” in *LICS 2012: 235-244*, 2012.
- [17] J. Gutierrez and G. Winskel, “Borel determinacy of concurrent games,” in *CONCUR 2013: 516-530*, 2013.
- [18] D. A. Martin, “Borel determinacy,” *Annals of Mathematics*, vol. 102, no. 2, pp. 363–371, 1975.
- [19] C. Jones and G. Plotkin, “A probabilistic powerdomain of valuations,” in *LICS '89*. IEEE Computer Society, 1989.
- [20] V. A. Rohlin, “On the fundamental ideas of measure theory,” *Amer. Math. Soc. Translation 1952*, no. 71, p. 55.
- [21] 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.
- [22] S. Abramsky and A. Brandenburger, “A unified sheaf-theoretic account of non-locality and contextuality,” *CoRR abs/1102.0264*, 2011.
- [23] R. B. Griffiths, *Consistent quantum theory*. CUP, 2002.
- [24] J. Grabbe, “An introduction to quantum game theory,” *arXiv preprint quant-ph/0506219*, 2005.
- [25] P. Clairambault and G. Winskel, “On concurrent games with payoff,” *Electr. Notes Theor. Comput. Sci.* 298: 71-92, 2013.
- [26] 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.
- [27] D. Martin, “The determinacy of Blackwell games,” *Journal of Symbolic Logic*, vol. 63, no. 4, pp. 1565–1581, 1998.
- [28] S. Castellan, P. Clairambault, and G. Winskel, “Symmetry in concurrent games,” in *LICS'14*. ACM, 2014.
- [29] G. Winskel, “Event structures with symmetry,” *Electr. Notes Theor. Comput. Sci.* 172: 611-652, 2007.
- [30] A. Joyal, M. Nielsen, and G. Winskel, “Bisimulation from open maps,” *Inf. Comput.*, vol. 127, no. 2, pp. 164–185, 1996.
- [31] S. Abramsky, R. Jagadeesan, and P. Malacaria, “Full abstraction for PCF,” *Inf. Comput.* 163(2): 409-470, 2000.
- [32] J. M. E. Hyland and C.-H. L. Ong, “On full abstraction for PCF: I, II, and III,” *Inf. Comput.* 163(2): 285-408, 2000.
- [33] C.-H. L. Ong, “Model checking algol-like languages using game semantics,” in *FSTTCS*, ser. LNCS, vol. 2556. Springer, 2002, pp. 33–36.