Distributed Strategies

Glynn Winskel

LIAFA (Algorithmics) $\leftrightarrow$ PPS (Semantics)

A tension, *Algorithmics:* (optimal) computation *in specific contexts*
Semantics: algebra of computation *in general contexts*

Challenge: to find a notion of computational process, rich in algebraic structure, while faithful to algorithmic concerns (so necessarily intensional).

A possible candidate: *distributed strategies*, at the junction of algorithmics, semantics and logic.

Paris 22 April 2013

Acknowledgements (and Tribute)

Gilles Kahn [and Gordon Plotkin]

Gérard Berry

Pierre-Louis Curien

Thomas Ehrhard and Antonio Bucciarelli

Daniele Varacca and Samy Abbes

Claudia Faggian and Mauro Piccolo

Paul-André Melliès and Samuel Mimram

Vincent Danos and Russ Harmer

Acknowledgements: Silvain Rideau, Pierre Clairambault, Julian Gutierrez, Simon Castellan, Martin Hyland

Logic and Algebra

The tension between computation and its algebraic formulation has precursors in Logic:

Arend Heyting: the algebra of intuitionistic logic *vs.* Brouwer's view

Alfred Tarski: the algebra of logic;
definition of truth of first-order predicate calculus

⋮

Jean-Yves Girard: more informative semantics, often by bringing algebra of logic closer to the computational

Computational process?

Pre 1930's: **An algorithm** *(informal)*

Post 1930's: **An effective partial function** $\mathbf{f} : \mathbb{N} \to \mathbb{N}$ *(mathematical)*

Mid 1960's : Christopher Strachey founded denotational semantics to understand *stored programs*, *loops*, *recursive programs* on *advanced datatypes*, often with *infinite objects* (at least conceptually): infinite lists, infinite sets, functions, functions on functions on functions, ...
A program denotes a term within the λ-calculus, a calculus of functions (but is it?): $\mathbf{t} ::= \mathbf{x} \mid \lambda\mathbf{x}.\mathbf{t} \mid (\mathbf{t}\,\mathbf{t}')$

Late 1960's: Dana Scott: Computable functions acting on infinite objects can only do so via approximations (topology!). **A computational process is an (effective) continuous function $\mathbf{f} : \mathbf{D} \to \mathbf{E}$ between special topological spaces, 'domains.'** Recursive definitions as least fixed points.

But, operational and compositionality concerns:

1975 Gilles Kahn and Gordon Plotkin: Motivated by full-abstraction, **concrete data structures** and **sequential functions**. But no function spaces!

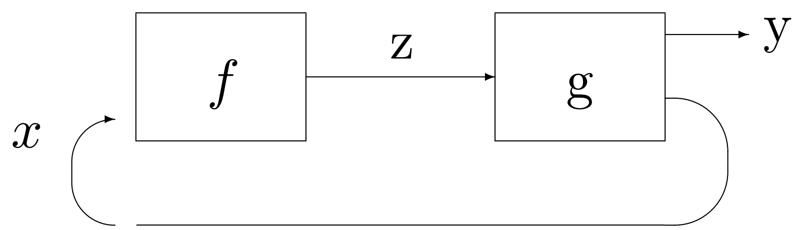
1978 Gérard Berry: **dI-domains and stable functions** and the stable order; **bidomains** marrying the Scott and stable order. Have function spaces.

1980 Gérard Berry and Pierre-Louis Curien: Concrete data structures and **sequential algorithms**. Have 'function' spaces.

1981 Brock and Ackerman anomaly for nondeterministic dataflow
$\leadsto$ semantics outside Scott's domain theory.

All are (sometimes implicitly) based on **event structures**, *events, their causal dependencies and conflicts (cf. distributed computation), and will be seen as instances of distributed strategies between distributed games.*

Deterministic dataflow—Kahn networks



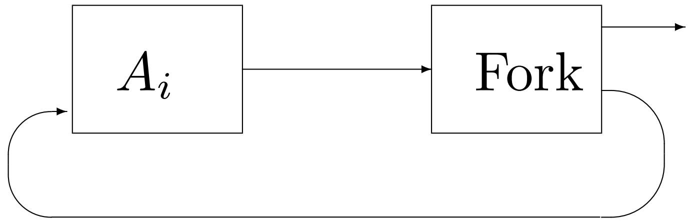
A process built from basic processes connected by channels at which they input and output.

Simple semantics: Associate channels with streams x, y, z.
Provided f and g are continuous functions on streams there is a least fixed point

$$(x, y, z) = (g(z)_2, g(z)_1, f(x)) \ .$$

Nondeterministic dataflow—the Brock-Ackerman anomaly

$C[A_i] =$



Both nondeterministic processes

$$A_1 = O + OIO \qquad \text{and} \qquad A_2 = O + IOO$$

have the same I/O relation, comprising

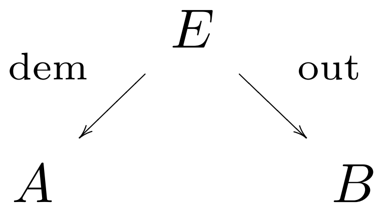
$$(\varepsilon, O),\ (I, O),\ (I, OO)\ .$$

But

$$C[A_1] = O + OO \qquad \text{and} \qquad C[A_2] = O\ .$$

The solution: 'non-deterministic stable functions'

A process with input event structure A and output event structure B:



where E is an event structure—generalizing the *trace* of a stable function, *describing the ways to compute between input and output*

$\mathrm{out} : E \to B$ is a map expressing the output produced by an event in E,

$\mathrm{dem} : E \to A$ is a map expressing the requirement on input of an event in E.

Event structures

An **event structure** comprises $(E, \leq, \mathrm{Con})$, consisting of a set of *events* E

- partially ordered by $\leq$, the **causal dependency relation**, and

- a nonempty family Con of finite subsets of E, the **consistency relation**,

which satisfy

$$\{e' \mid e' \leq e\} \text{ is finite for all } e \in E,$$

$$\{e\} \in \mathrm{Con} \text{ for all } e \in E,$$

$$Y \subseteq X \in \mathrm{Con} \Rightarrow Y \in \mathrm{Con}, \quad \text{and}$$

$$X \in \mathrm{Con} \;\&\; e \leq e' \in X \Rightarrow X \cup \{e\} \in \mathrm{Con}.$$

In games the relation of **immediate dependency** $e \rightarrowtriangle e'$, meaning e and e' are distinct with $e \leq e'$ and no event in between, will play an important role.

Configurations of an event structure

The **configurations**, $\mathcal{C}^\infty(E)$, of an event structure E consist of those subsets $x \subseteq E$ which are

Consistent: $\forall X \subseteq_{\rm fin} x.\ X \in \mathrm{Con}$ and

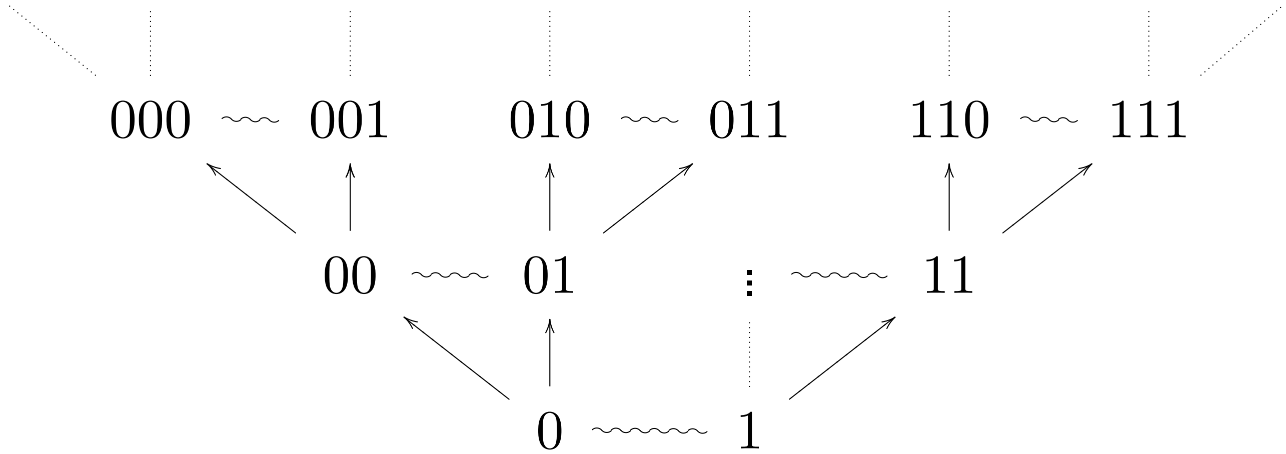
Down-closed: $\forall e, e'.\ e' \leq e \in x \Rightarrow e' \in x$.

$x \subseteq x'$, *i.e.* x is a sub-configuration of x', means that x is a sub-history of x'.

Often concentrate on the **finite configurations** $\mathcal{C}(E)$.

[Berry's dI-domains are exactly $(\mathcal{C}^\infty(E), \subseteq)$, when E is countable.]

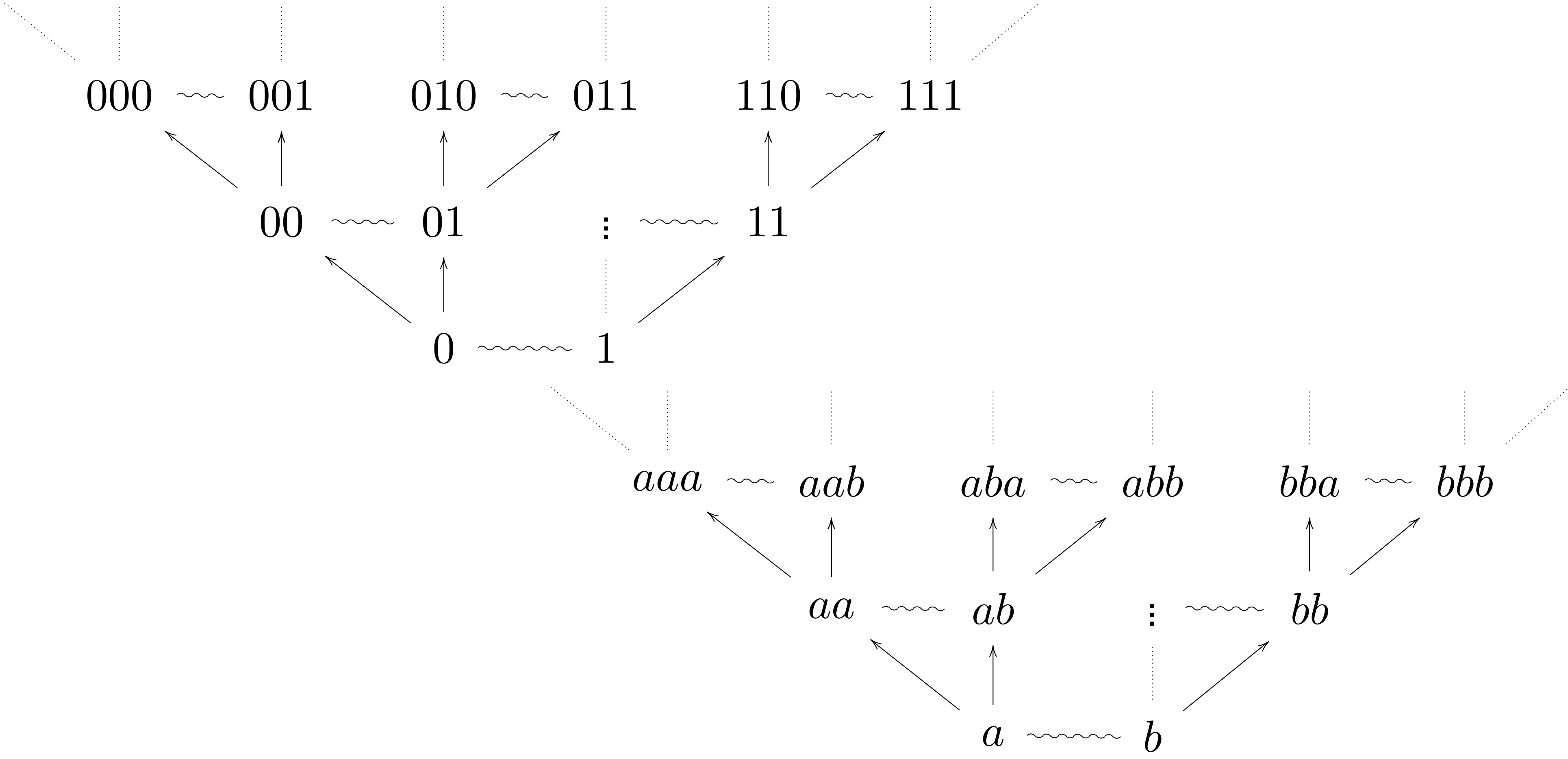
Example: Streams as event structures



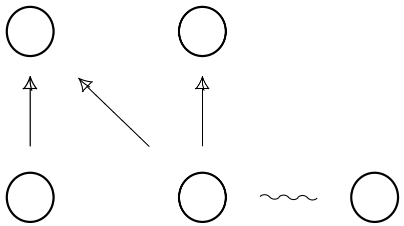
 conflict (inconsistency)

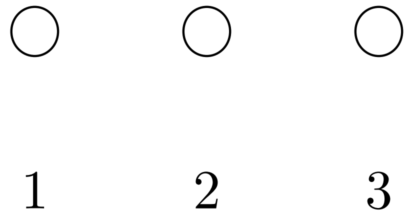
 immediate causal dependency

Simple parallel composition



Other examples





$$\mathrm{Con} = \{\ \emptyset,\ \ \{1\},\{2\},\{3\},\ \ \{1,2\},\{1,3\},\{2,3\}\ \}$$

Structural maps of event structures

A **map** of event structures $f : E \to E'$
is a partial function on events $f : E \rightharpoonup E'$ such that for all $x \in \mathcal{C}(E)$

$fx \in \mathcal{C}(E')$ and

if $e_1, e_2 \in x$ and $f(e_1) = f(e_2)$, then $e_1 = e_2$. *(local injectivity)*

Idea: *the occurrence of an event e in E induces the coincident occurrence of the event $f(e)$ in E' whenever it is defined.*

$\leadsto$

- Semantics of synchronising processes [Hoare, Milner] can be expressed in terms of universal constructions on event structures.
- Relations between models via adjunctions.

Distributed games

Games and strategies are represented by **event structures with polarity**, an event structure $(E, \leq, \mathrm{Con})$ where events E carry a polarity $+/-$ (Player/Opponent), respected by maps.

(Simple) Parallel composition: $A \| B$, by juxtaposition.

Dual, $B^{\perp}$, of an event structure with polarity B is a copy of the event structure B with a reversal of polarities; this switches the roles of Player and Opponent.

Distributed plays and strategies

A **nondeterministic play** in a game A is represented by a total map

$$\sigma : S \to A$$

preserving polarity; S is the event structure with polarity describing the moves played.

A **strategy in** a game A is a (**special**) nondeterministic play $\sigma : S \to A$.

A **strategy from** A **to** B is a strategy in $A^\perp \parallel B$, so $\sigma : S \to A^\perp \parallel B$.

[Conway, Joyal]

Strategies as spans

A strategy from A to B is a (special) total map of event structures with polarity

$$\sigma : S \to A^{\perp} \parallel B\,.$$

It corresponds to a *span* of event structures with polarity

$$\begin{array}{ccc} & S & \\ {}^{\sigma_1}\swarrow & & \searrow^{\sigma_2} \\ A^{\perp} & & B \end{array}$$

where σ_1, σ_2 are *partial* maps of event structures with polarity; one and only one of σ_1, σ_2 is defined on each event of S.

Example: Copy-cat

Identities on games A are given by copy-cat strategies

$$\gamma_A : \mathbb{C}\mathbb{C}_A \to A^\perp \parallel A$$

—strategies for player based on copying the latest moves made by opponent.

$\mathbb{C}\mathbb{C}_A$ has the same events and polarity as $A^\perp \parallel A$ but with causal dependency $\leq$ given as the transitive closure of the relation $\leq_{A^\perp \parallel A}$ extended by

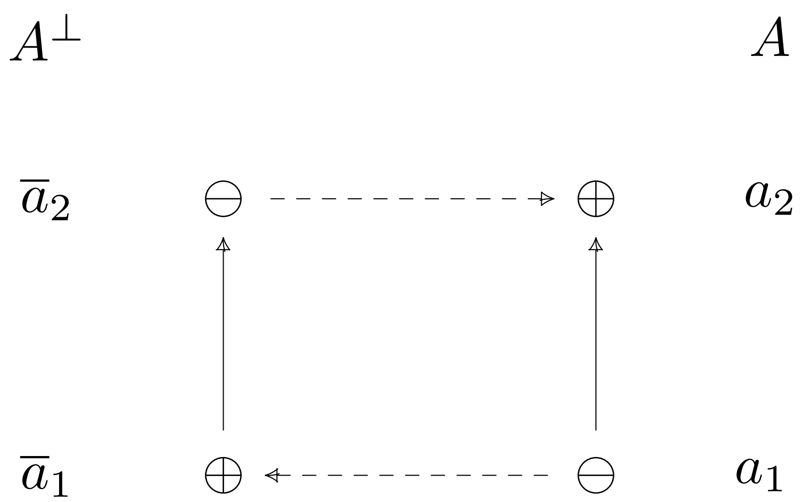
$$\overline{c} \leq c \text{ if } pol_{A^\perp \parallel A}(c) = +\,,$$

where $\overline{c} \leftrightarrow c$ is the natural correspondence between $A^\perp$ and A.
A finite set is consistent if its $\leq$-down-closure is consistent in $A^\perp \parallel A$.

The map γ_A is the identity on the common underlying set of events.

Copy-cat—an example

$$\mathbb{C}\mathbb{C}_A$$



Composing strategies

Two strategies $\sigma : A \mathrel{\,+\!\!\!\!\!\longrightarrow} B$ and $\tau : B \mathrel{\,+\!\!\!\!\!\longrightarrow} C$ as spans:

$$
\begin{array}{ccccc}
 & & S & & \\
 & \sigma_1 \swarrow & & \searrow \sigma_2 & \\
A^{\perp} & & & & B
\end{array}
\qquad
\begin{array}{ccccc}
 & & T & & \\
 & \tau_1 \swarrow & & \searrow \tau_2 & \\
B^{\perp} & & & & C\,.
\end{array}
$$

Their composition

$$
\begin{array}{ccccc}
 & & T \odot S & & \\
 & (\tau \odot \sigma)_1 \swarrow & & \searrow (\tau \odot \sigma)_2 & \\
A^{\perp} & & & & C
\end{array}
$$

where $T \odot S =_{\mathrm{def}}$ *their synchronized composition with hiding...*

For copy-cat to be identity w.r.t. composition

a strategy in a game A has to be $\sigma : S \to A$, a total map of event structures with polarity, such that
(i) whenever $\sigma x \subseteq^- y$ in $\mathcal{C}(A)$ there is a unique $x' \in \mathcal{C}(S)$ so that
$x \subseteq x'$ & $\sigma x' = y$, *i.e.*

$$\begin{array}{ccc} x & \subseteq & x' \\ \sigma \downarrow & & \downarrow \sigma \\ \sigma x & \subseteq^- & y \,, \end{array}$$

and
(ii) whenever $y \subseteq^+ \sigma x$ in $\mathcal{C}(A)$ there is a (necessarily unique) $x' \in \mathcal{C}(S)$ so that
$x' \subseteq x$ & $\sigma x' = y$, *i.e.*

$$\begin{array}{ccc} x' & \subseteq & x \\ \sigma \downarrow & & \downarrow \sigma \\ y & \subseteq^+ & \sigma x \,. \end{array}$$

The only immediate causal dependencies a strategy can introduce: $\ominus \rightarrowtriangle \oplus$

A bicategory of games and strategies

in which **objects are games**, and **maps are strategies**; it is monoidal-closed with **tensor** $A \| B$ and **function space** $A \multimap B =_{\mathrm{def}} A^{\perp} \| B$.

A strategy is **deterministic** iff consistent behaviour of Opponent implies consistent behaviour of Player. $\rightsquigarrow$ A sub-category where maps are **deterministic strategies** and objects are 'race-free' games. [Melliès & Mimram's *receptive ingenuous* strategies]

When games comprise purely Player-moves, obtain the **stable spans used in nd dataflow**. Includes Berry's **dI-domains and stable functions** as the deterministic subcategory.

Includes **simple games** at the basis of **sequential algorithms**, **AJM** and **HO games** of traditional game semantics.

Consequences and extensions

1. An affine higher-order language of distributed strategies

2. Winning conditions, and Borel determinacy (iff race-free and bounded-concurrent); and games with pay-off

3. Probabilistic and Quantum strategies

4. Imperfect information

5. Symmetry and back-tracking in games

6. Games as factorization systems ($\rightsquigarrow$ Berry's bidomains as games)

1. An affine higher-order language of distributed strategies

Below, $\sigma : A$ means σ *is a strategy in game A.*

Composition $\sigma \odot \tau : A \| C$, if $\sigma : A \| B$ and $\tau : B^{\perp} \| C$.

Simple parallel composition $\sigma \| \tau : A \| B$, if $\sigma : A$ and $\tau : B$.

Pullback $f^* \sigma : A$, if $\sigma : B$ and $f : A \to B$ (subsumes prefixing $\oplus.\sigma$, $\ominus.\sigma$).

Relabelling $f\sigma : B$, if $\sigma : A$ and $f : A \to B$ is a strategy.

Sum $\sum_{i \in I} \sigma_i : A$, if $\sigma_i : A$ for all $i \in I$.

Lambda-abstraction $\lambda x : A.\sigma : A^{\perp} \| B$, if $x : A \vdash \sigma : B$.

Recursion on types and processes.

2. Winning strategies

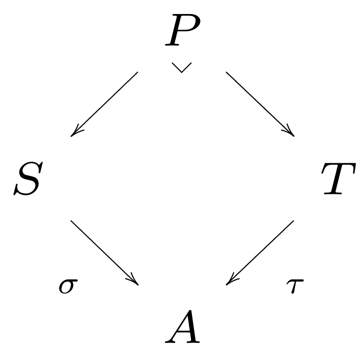
A strategy $\sigma : S \to A$, with $\mathbf{W} \subseteq \mathcal{C}^\infty(\mathbf{A})$, is **winning** (for Player)

iff

any maximal play against a counter-strategy $\tau : T \to A^\perp$ results in a win for Player

iff

$\sigma x \in W$, for all $+$-maximal configurations $x \in \mathcal{C}^\infty(S)$.



A **winning strategy from** (A, W_A) **to** (B, W_B) is a winning strategy in $(A, W_A)^\perp \| (B, W_B)$ where

$(A, W_A)^\perp = (A^\perp, W_{A^\perp})$ where $W_{A^\perp}$ is the complement of W_A.

$(A, W_A) \| (B, W_B) = (A \| B, W_{A\|B})$ where

$$x \in W_{A\|B} \iff x_A \in W_A \text{ or } x_B \in W_B \,.$$

To win in $G \| H$ is to win in either game G or H.

Winning strategies compose $\rightsquigarrow$ *a bicategory of winning strategies.*
Extends to zero-sum games with payoff.

3. Probabilistic strategies

A **probabilistic event structure** is an event structure E and a **configuration-valuation** $v : \mathcal{C}(E) \to [0,1]$ such that ..., which is equivalent to a continuous valuation on the open sets of $\mathcal{C}^{\infty}(E)$, so yields a probability measure on Borel subsets. [Samy Abbes, Daniele Varacca]

A **probabilistic strategy** is a strategy which 'projects to' a probabilistic event structures on the Player-moves. They form a bicategory on 'race-free' games.

In a **quantum event structure** and **quantum game** events e are interpreted as unitary or projection operators Q_e on Hilbert space so that concurrent events are associated with commuting operators. A consequence: given a start state, a density operator ρ, a quantum event structure determines a probabilistic event structure, on compatible configurations—*cf. consistent-history approach to quantum theory [Griffiths, Omnès].*

4. Imperfect information

W.r.t. a preorder of **access levels** $(\Lambda, \preceq)$,

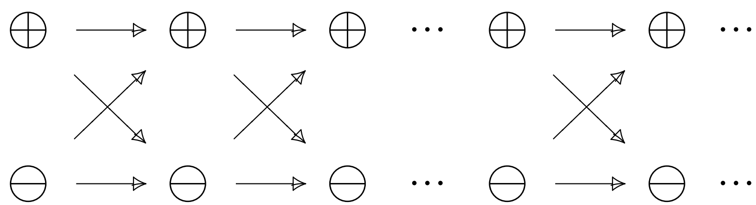
a Λ-**game** comprises a game A with a *level function* $l : A \to \Lambda$ such that

$$\forall a, a' \in A.\ a \leq_A a' \Rightarrow l(a) \preceq l(a')\,;$$

a Λ-**strategy** is a strategy $\sigma : S \to A$ for which

$$\forall s, s' \in S.\ s \leq_S s' \Rightarrow l\sigma(a) \preceq l\sigma(a')\,.$$

$E.g.$ $(\Lambda, \preceq)$ for Blackwell games (aka. ‘concurrent games’):



In conclusion

Arguably, the concept of strategy is potentially as fundamental, broadly applicable and precise as that of relation.

To realize this the concept needs to be developed in sufficient generality, outside the confines of traditional sequential games.

At the junction of algorithmics, logic and semantics, distributed strategies and games would appear to combine algebraic structure with algorithmic expressivity.

But can they support the tight contextual reasoning demanded by algorithmics?

Needs further investigation!

In conclusion

Arguably, the concept of strategy is potentially as fundamental, broadly applicable and precise as that of relation.

To realize this the concept needs to be developed in sufficient generality, outside the confines of traditional sequential games.

At the junction of algorithmics, logic and semantics, distributed strategies and games would appear to combine algebraic structure with algorithmic expressivity.

But can they support the tight contextual reasoning demanded by algorithmics?

Needs further investigation!

THANK YOU!

5. Games with symmetry

A **game with symmetry** comprises a game A and a bisimulation equivalence on A.

Maps respect symmetry and can now be considered "equal up to symmetry".

$\rightsquigarrow$ great range of (co)monads up to symmetry.

Strategies as (special) presheaves/profunctors

Defining a partial order — *the Scott order* — on configurations of A

$$x \sqsubseteq_A y \;\text{ iff }\; x \supseteq^- \cdot \subseteq^+ \cdot \supseteq^- \cdots \supseteq^- \cdot \subseteq^+ y\,.$$

we obtain a factorization system $((\mathcal{C}(A), \sqsubseteq_A), \supseteq^-, \subseteq^+)$, *i.e.* $\exists! z.$

$$\begin{array}{ccc} & & y \\ & \sqsubseteq & \subseteq^+ \\ x & \supseteq^- & z\,. \end{array}$$

Proposition $z \in \mathcal{C}(\mathbb{C}\!\mathbb{C}_A)$ iff $z_2 \sqsubseteq_A \overline{z}_1$.

Theorem *Strategies in A correspond to total $\sigma : S \to A$ which yield discrete fibrations*

$$\sigma``: (\mathcal{C}(S), \sqsubseteq_S) \to (\mathcal{C}(A), \sqsubseteq_A)\,, \quad i.e. \quad \exists! x'. \quad \begin{array}{ccc} x' & \sqsubseteq_S & x \\ \sigma`` \downarrow & & \downarrow \sigma`` \\ y & \sqsubseteq_A & \sigma``(x)\,, \end{array}$$

and preserve $\supseteq^-$, $\subseteq^+$ *and* $\emptyset$.

6. Strategies on ‘rooted’ factorization systems

A strategy on a ‘rooted’ factorization system $(\mathbb{A}, L_A, R_A, 0_A)$ is a discrete fibration

$$F : (\mathbb{S}, L_S, R_S, 0_S) \to (\mathbb{A}, L_A, R_A, 0_A)\,,$$

from another rooted factorization system $(\mathbb{S}, L_S, R_S, 0_S)$, which preserves L, R maps and 0.

Example: The map $\sigma\text{``} : ((\mathcal{C}(S), \sqsubseteq_S), \supseteq^-, \subseteq^+, \emptyset) \to ((\mathcal{C}(A), \sqsubseteq_A), \supseteq^-, \subseteq^+, \emptyset)$ induced by a strategy $\sigma : S \to A$.

Operations $\quad (\mathbb{C}, L, R, 0)^\perp =_{\rm def} (\mathbb{C}^{\rm op}, R^{\rm op}, L^{\rm op}, 0)$

$$(\mathbb{B}, L_B, R_B, 0_B) \| (\mathbb{C}, L_C, R_C, 0_C) =_{\rm def} (\mathbb{B} \times \mathbb{C}, L_B \times L_C, R_B \times R_C, (0_B, 0_C))$$

Composition: *reachable part of* profunctor composition.

Games and strategies embed faithfully in rooted factorization systems.