Models for Concurrency and Games

Glynn Winskel

The notion of *deterministic/nondeterministic strategy* is potentially as fundamental as the notion of *function/relation*.
A broad enough notion of strategy must be planted firmly within a general model of concurrent/distributed/interactive computation.

The two ingredients of this course

Models for concurrency: Petri nets, event structures, Mazurkiewicz trace languages, transition systems, ... *(we concentrate on event structures)*

Games: 2-party nondeterministic concurrent games between Player (team of players) and Opponent (team of opponents)

This first lecture should give an idea of

- **partial-order models**, a form of model becoming important in a range of areas from security, systems, model checking, systems biology, to proof theory

- why I believe such models will become central in **semantics** of computation and can combine the two approaches, **operational** and **denotational** semantics through the medium of games

Games informally

A **game** G provides constraints on the moves Opponent and Player can make, and often specifies winning conditions.

A **strategy** for Player prescribes moves for Player in answer to moves of Opponent.

Two important operations on games: **parallel composition** of games $G \| H$; **dual** of a game $G^\perp$ (reversing the roles of Player and Opponent)

Joyal after Conway: A strategy **from** a game G **to** a game H, $G \mathrel{\ooalign{$\rightarrow$\cr\hidewidth$+$\hidewidth}} H$, is a strategy in $G^\perp \| H$; strategies compose with identities given by copy-cat. A strategy in G corresponds to a strategy from the empty game $\emptyset$ to G. So

$$\emptyset \mathrel{+\!\!\rightarrow} G \mathrel{+\!\!\rightarrow} H \text{ composes to give } \emptyset \mathrel{+\!\!\rightarrow} H\,,$$

so a strategy in G gives rise to a strategy in H when $G \mathrel{+\!\!\rightarrow} H$.

Games in a model for concurrency

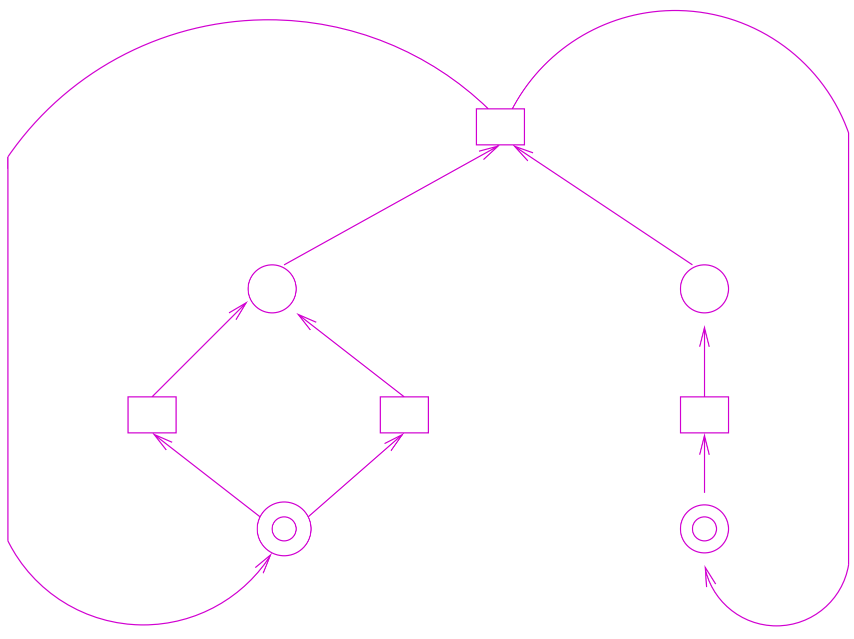
Lead to

- Generalised domain theory (via Joyal-Conway)
- Operations, including higher-order operations, on models for concurrency
- Techniques for Logic (via proofs as concurrent strategies) and (possibly) verification and algorithmics

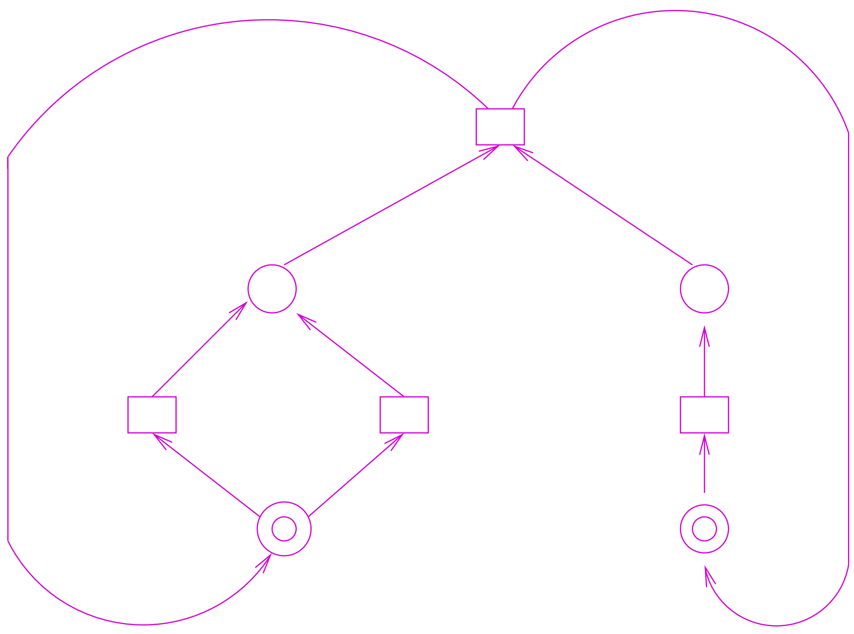
Causal/partial-order models

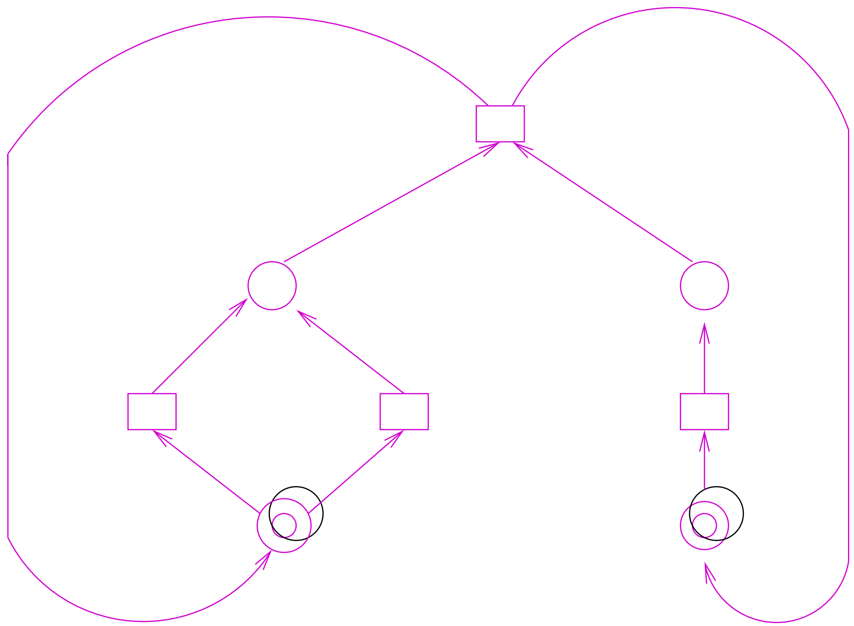
their range and applications ...

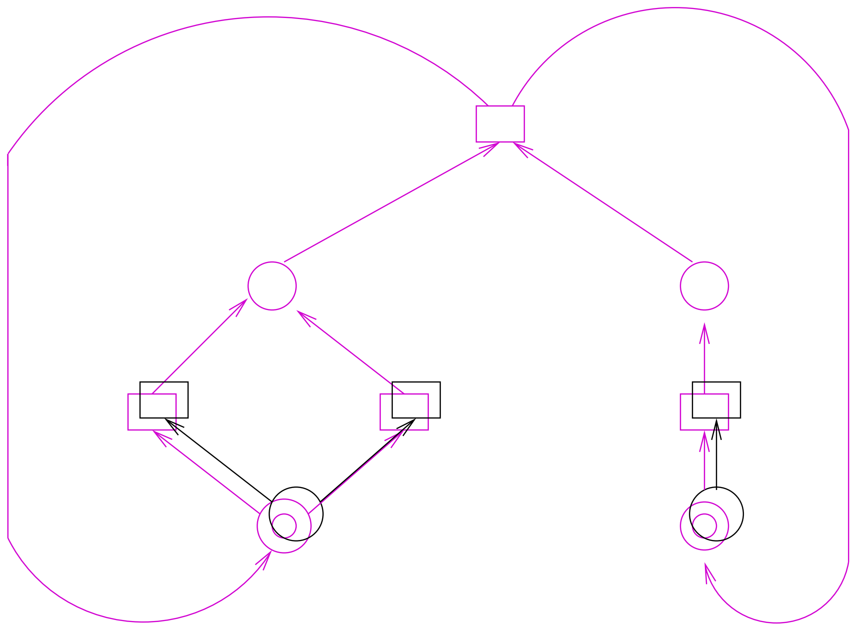
A (safe) Petri net

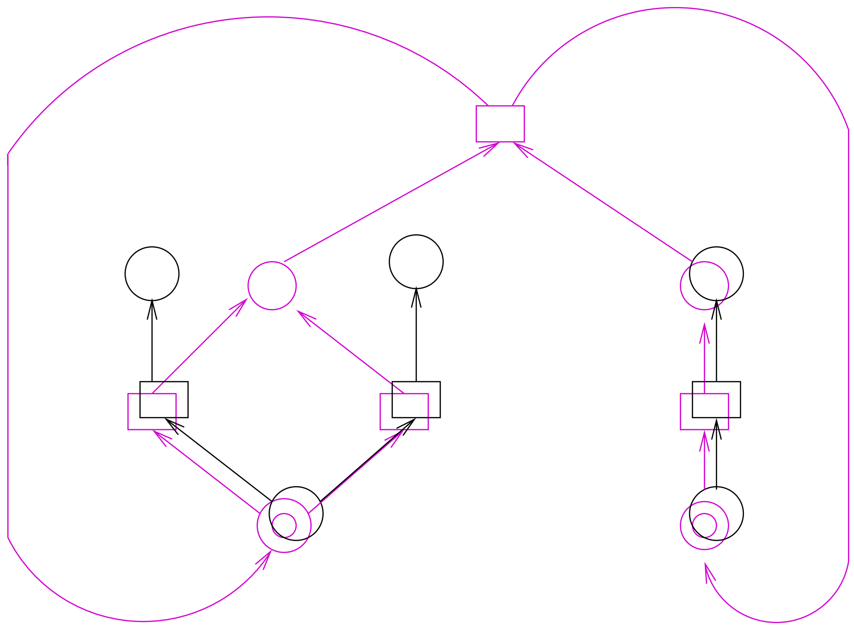


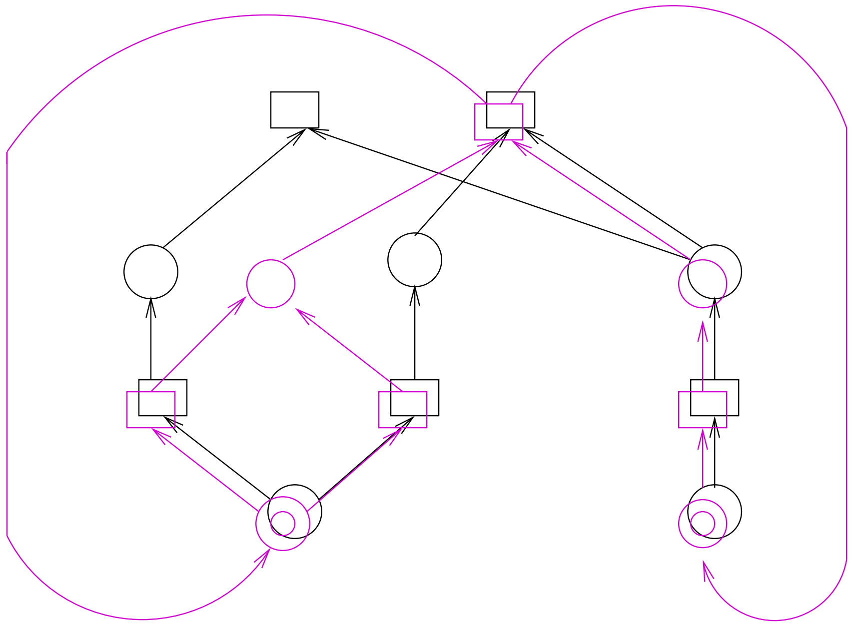
Unfolding a (safe) Petri net:

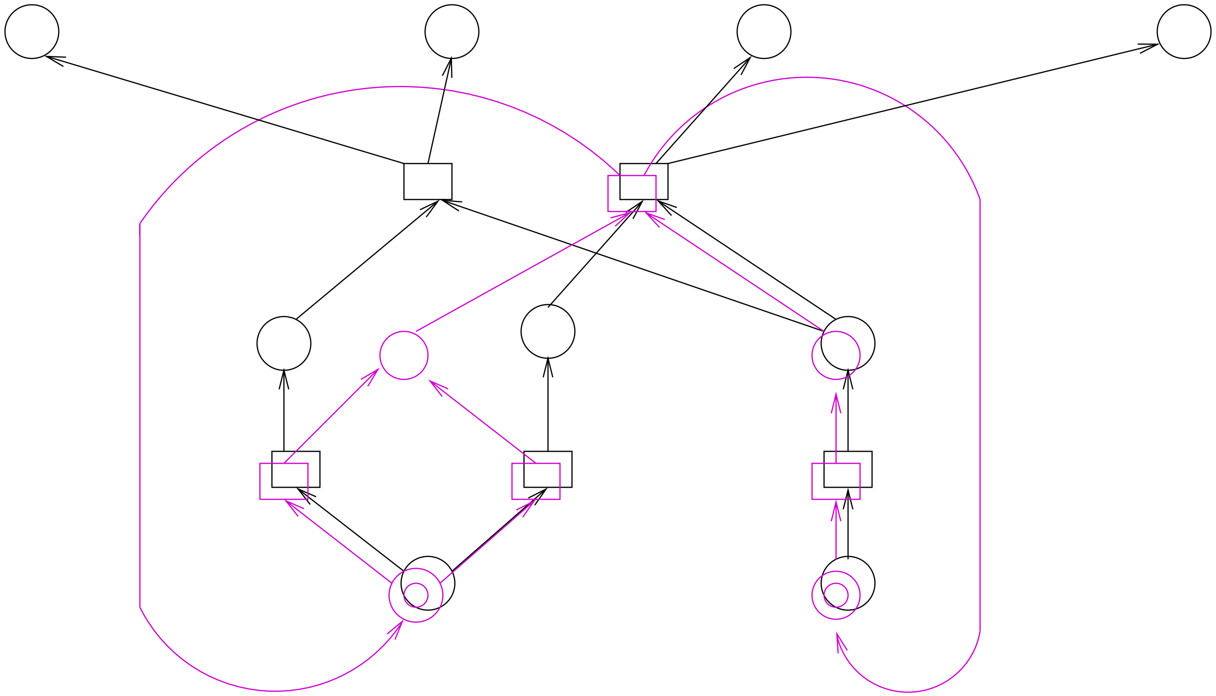


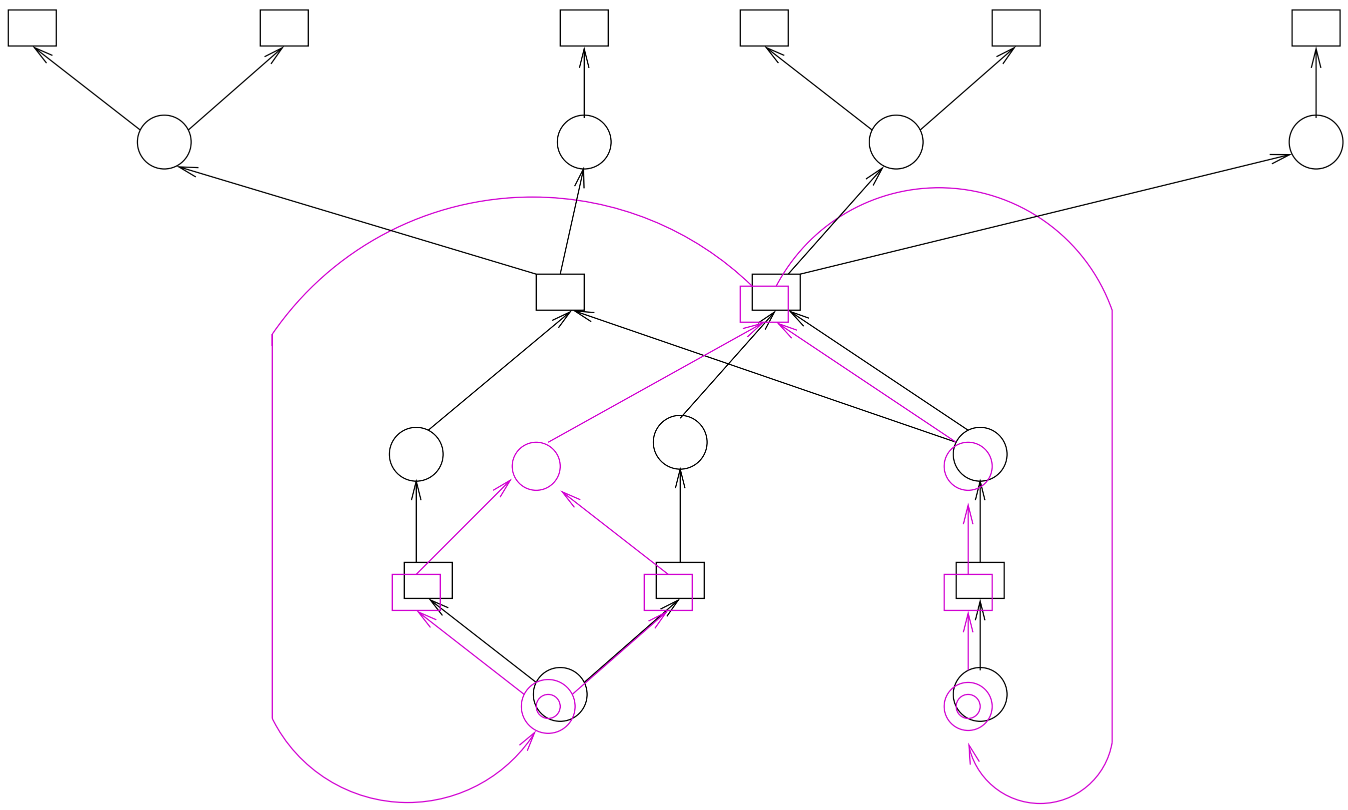


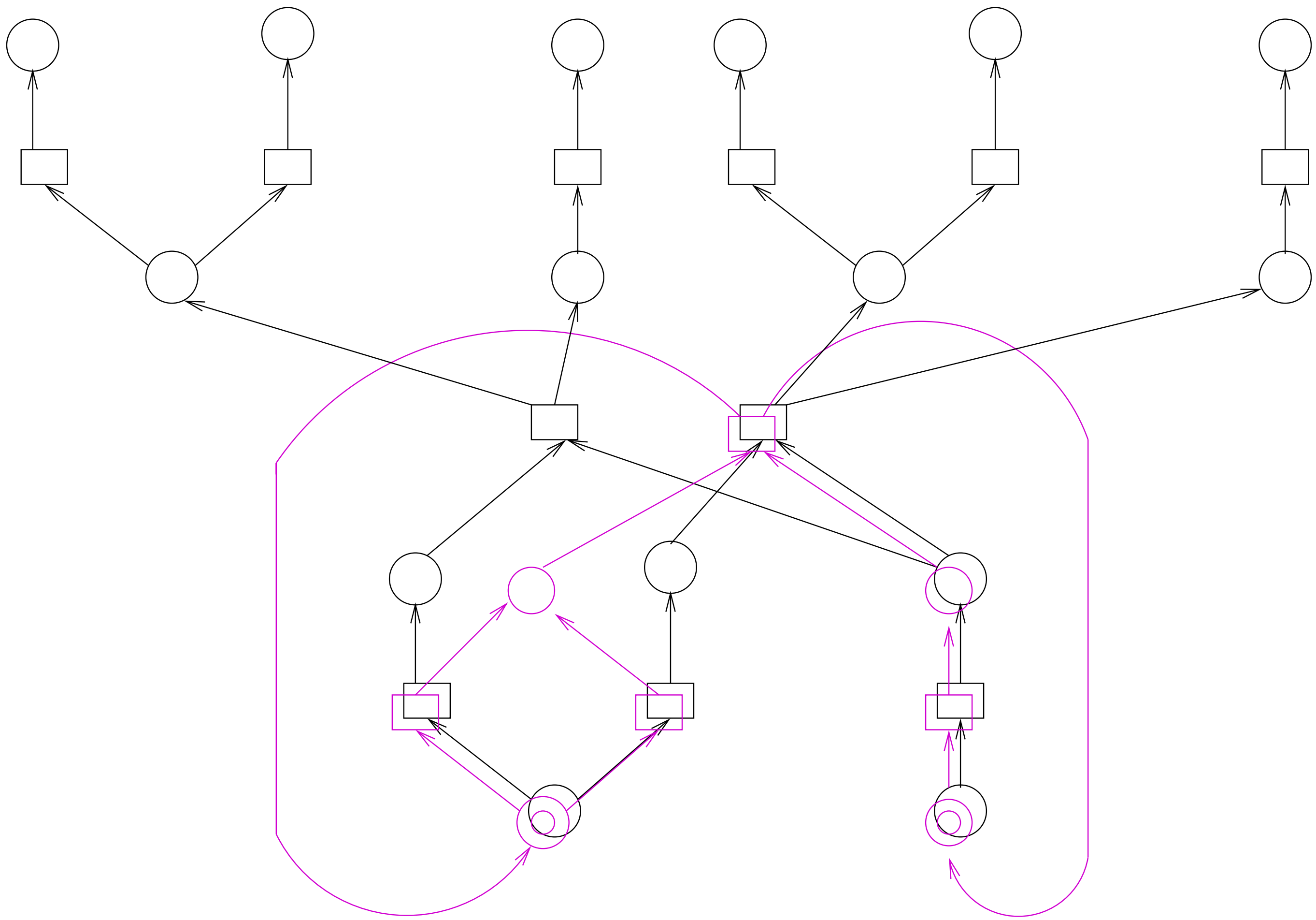


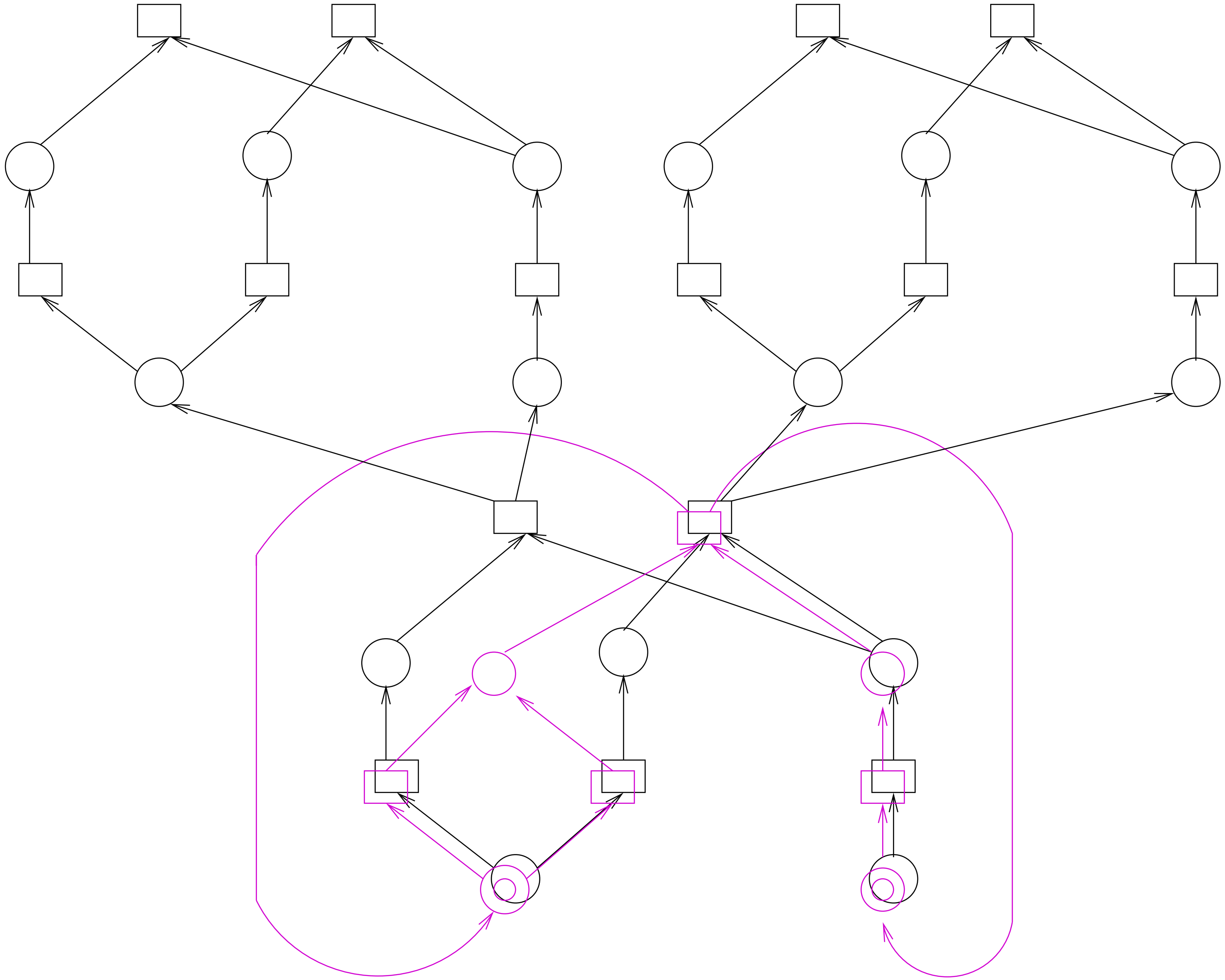


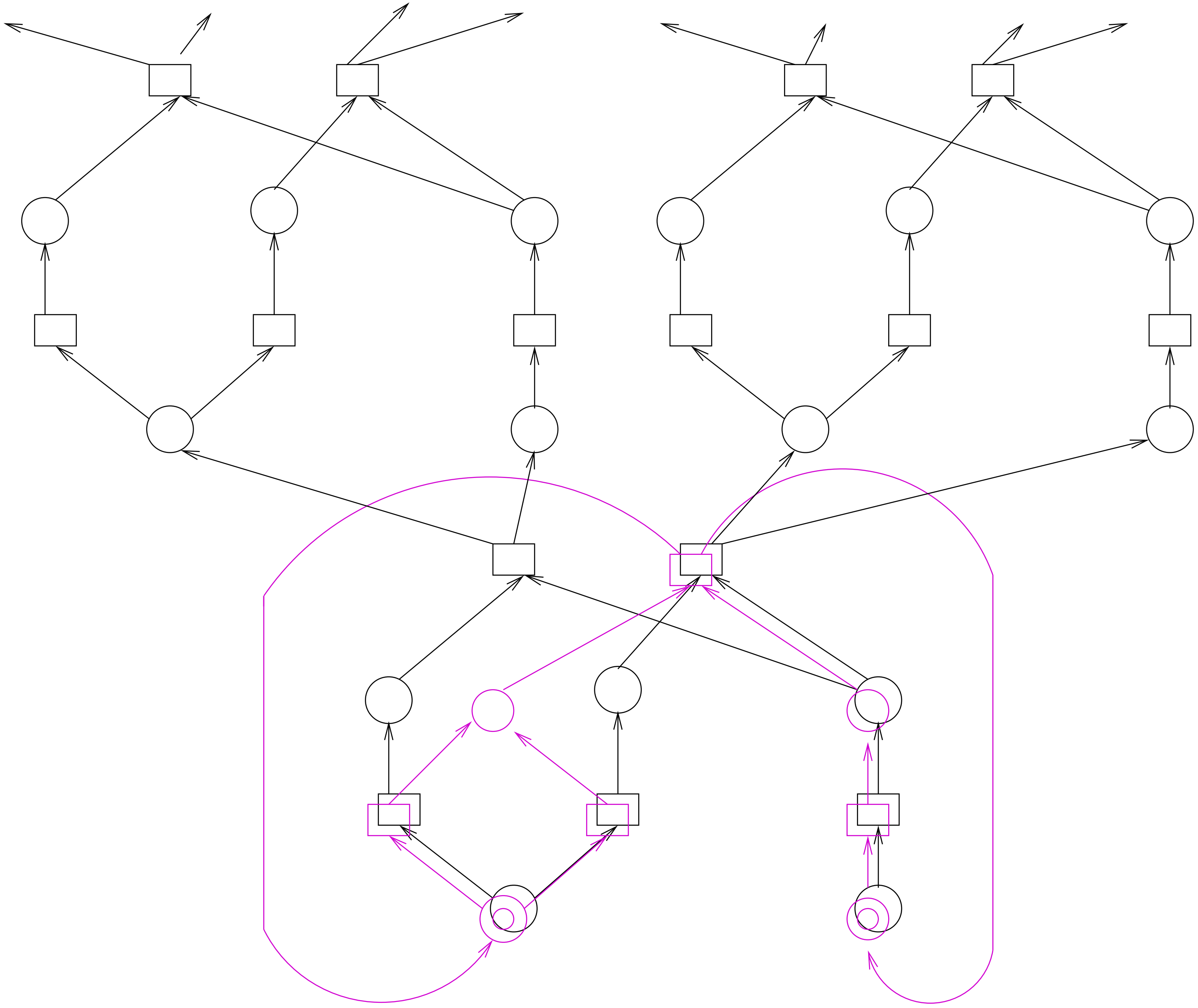


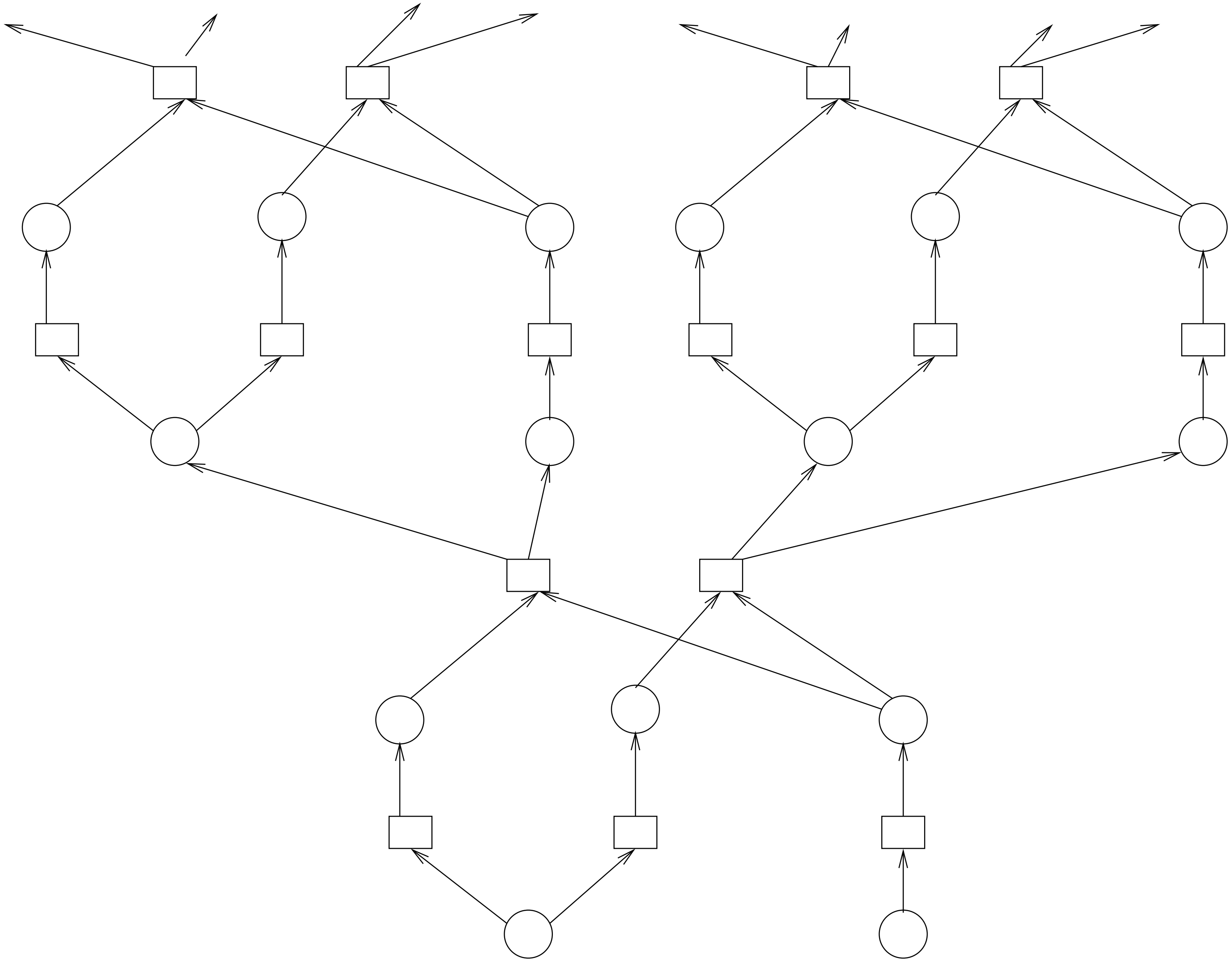


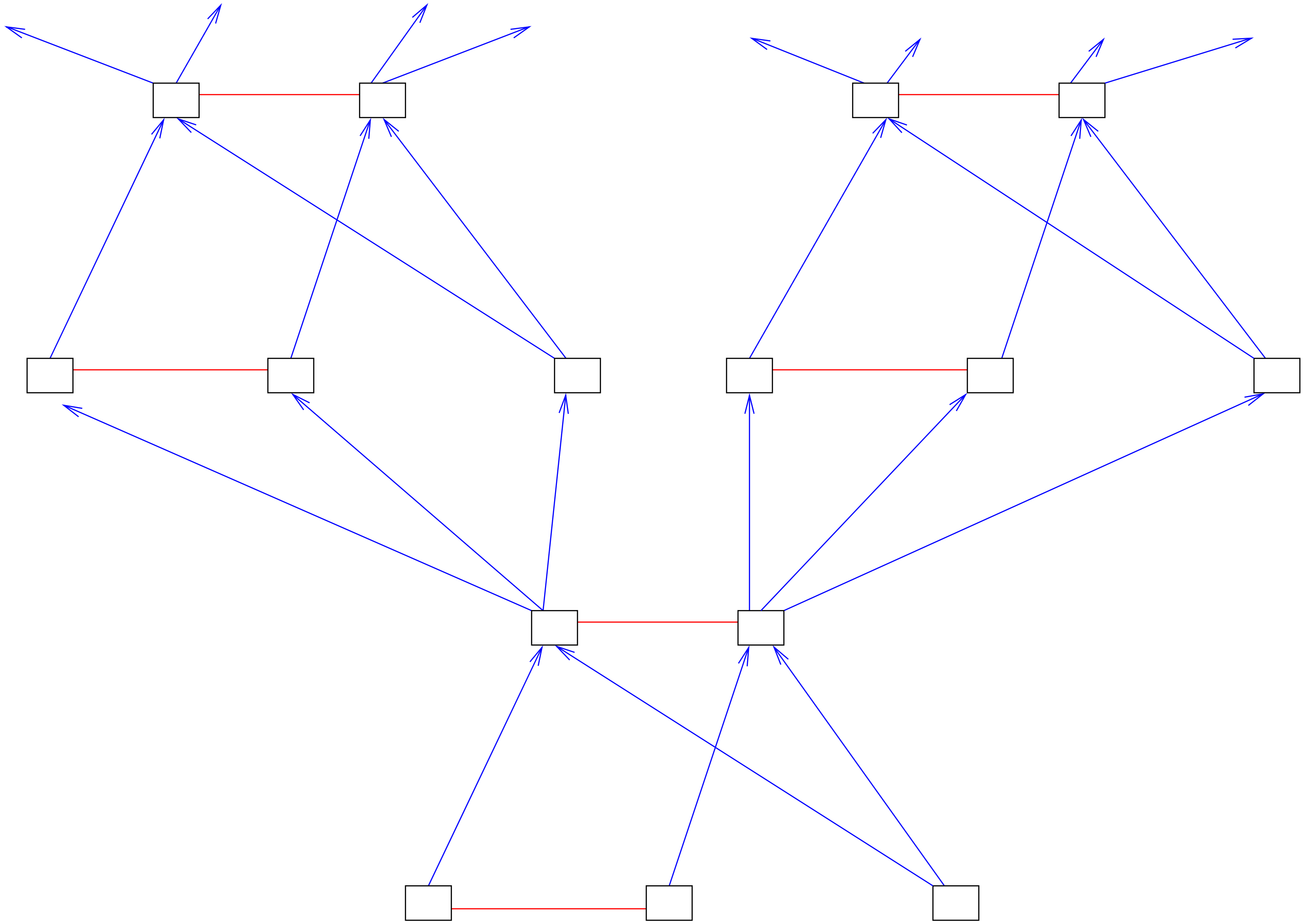












An event structure

Applications of partial-order models

Security protocols, as strand spaces [Guttman et al];
Systems biology, analysis of chemical pathways [Danos-Feret-Fontana-Krivine];
Hardware, in the design of asynchronous circuits [Yakovlev];
Types and proof, domain theory [Berry, Curien-Faggian, Girard];
Nondeterministic dataflow [Jonsson];
Network diagnostics [Benveniste et al];
Logic of programs, in concurrent separation logic;
Partial order model checking [McMillan];
Distributed computation, classically [Lamport] and recently in analysis of trust [Nielsen-Krukow-Sassone].

Domain theory and denotational semantics

Its history and limitations ...

What is a 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.

Basic domain theory

A *domain* is a complete partial order $(D, \sqsubseteq)$: any infinite chain

$$d_0 \sqsubseteq d_1 \sqsubseteq \cdots \sqsubseteq d_n \sqsubseteq \cdots$$

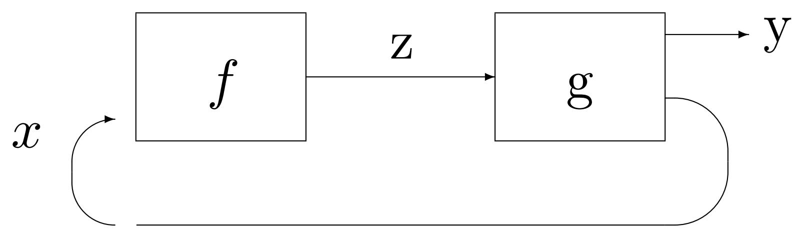
has a least upper bound $\bigsqcup_{n \in \omega} d_n$.

A function $f : D \to E$ is *continuous* if f preserves $\sqsubseteq$ and for all chains $f(\bigsqcup_{n \in \omega} d_n) = \bigsqcup_{n \in \omega} f(d_n)$.

If D has a least element $\perp$ and $f : D \to D$ is continuous, then f has a least fixed point $\bigsqcup_{n \in \omega} f^n(\perp)$. *(Recursive definitions)*

Scott (1969): A nontrivial solution to $D \cong [D \to D]$ *(a recursively defined domain)*, so providing a model of the λ-calculus, and, by the same techniques, the semantics of recursive types.

Deterministic dataflow—Kahn networks



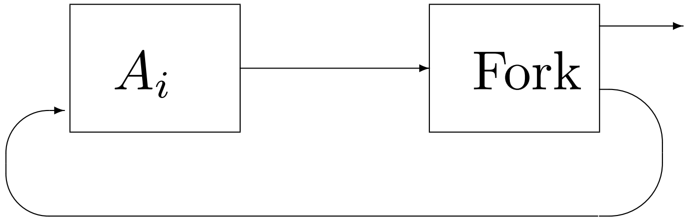
A process built from basic processes connected by channels at which they input and output.

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

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

A solution: generalize relations

A process with input A and output B:

$$
\begin{tikzcd}
 & E \arrow[dl, "\mathrm{dem}"'] \arrow[dr, "\mathrm{out}"] & \\
A & & B
\end{tikzcd}
$$

where A, B and E are event structures,

$\mathrm{out} : E \to B$ is a map expressing the different ways output is produced,

$\mathrm{dem} : E \to A$ is a map expressing the requirement on input for events to occur.

Such 'stable spans' first appeared in semantics of higher-order processes, where in special cases the 'ways' of computing input from output corresponded to derivations in an operational semantics. *They will reappear as a special kinds of strategies.*

Game semantics

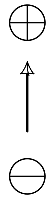
Traditional game semantics of programming languages, starting with the seminal work of Abramsky-Jagadeesan-Malacaria and Hyland-Ong, showed for sequential programs it was very fruitful to regard types as games and programs as strategies. AJM games and HO games are different though both sequential with Player and Opponent moves alternating.

In particular they both achieved *intensional full-abstraction* for the language PCF (the "intensional" is important and often forgotten).

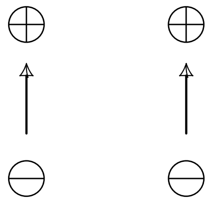
Many subsequent successes ...

Game semantics—a simple example

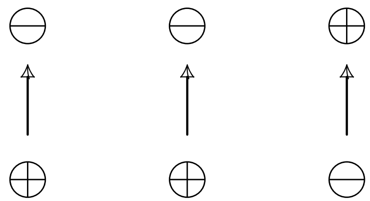
Type with a single value, the game:



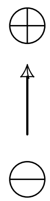
Type with a pair of values, the game:

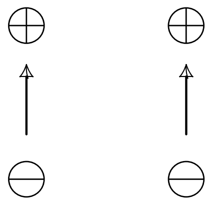


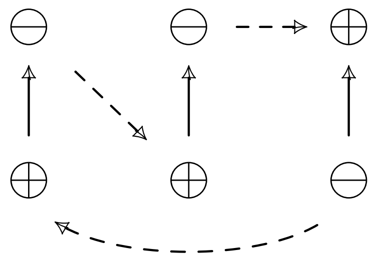
Type of ‘algorithms’ from pairs to value, the game:



Game semantics—a simple example

Type with a single value, the game: 

Type with a pair of values, the game: 

Type of 'algorithms' from pairs to value, the game: 

E.g. "after left then right input yield output"

Logic

The well-known **Curry-Howard correspondence**:

Propositions as types, proofs as programs

Through the denotation of types as games and programs/processes as strategies we obtain the correspondence:

Propositions as games, proofs as strategies

Games and strategies are becoming the denotational semantics of proof. But there are big gaps. Partly because games and strategies as known are not general enough. And there are still conceptual problems in giving a process reading to classical proof.

Other strands: games as a technique in logic, and in the definition of equivalences

Ch 2. EVENT STRUCTURES

Representations of domains

What is the information order? What are the 'units' of information?

('Topological') [Scott]: *Propositions* about finite properties;
more information corresponds to more propositions being true.
Functions are ordered pointwise.
Can represent domains via logical theories. ('Logic of domains')

('Temporal') [Berry]: *Events* (atomic actions);
more information corresponds to more events having occurred.
Intensional 'stable order' on 'stable' functions. ('Stable domain theory')
Can represent Berry's domains as event structures.

Event structures

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

- a set E, of *events*

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

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

which satisfy

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

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

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

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

Say e, e' are *concurrent* if $\{e, e'\} \in \mathrm{Con} \;\&\; e \not\leq e' \;\&\; e' \not\leq e$.

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: $\forall X \subseteq_{\text{fin}} x.\ X \in \text{Con}$ and

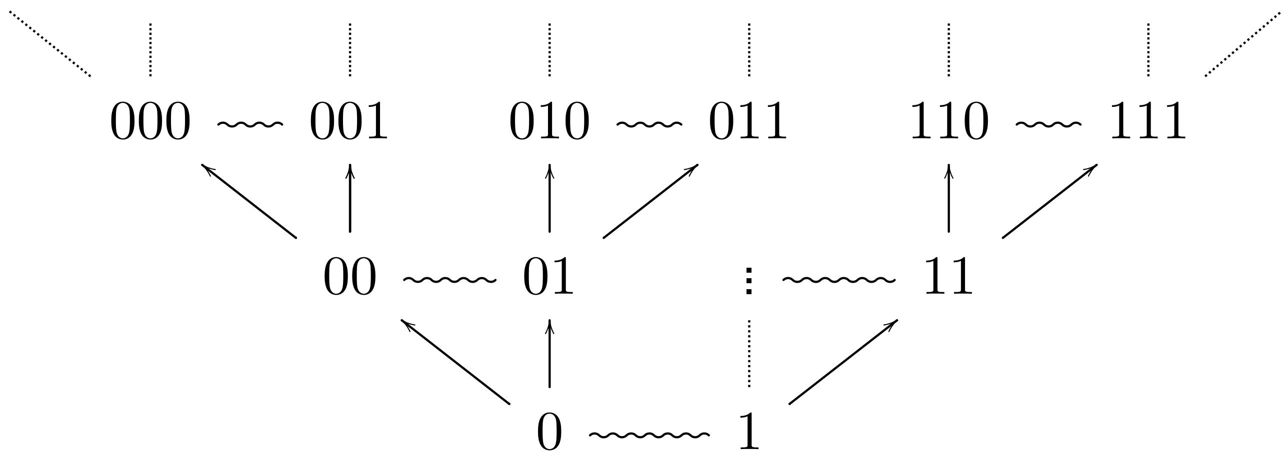
Down-closed: $\forall e, e'.\ e' \leq e \in x \Rightarrow e' \in x$.

For an event e the set $[e] =_{\text{def}} \{e' \in E \mid e' \leq e\}$ is a configuration describing the whole causal history of the event e.

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

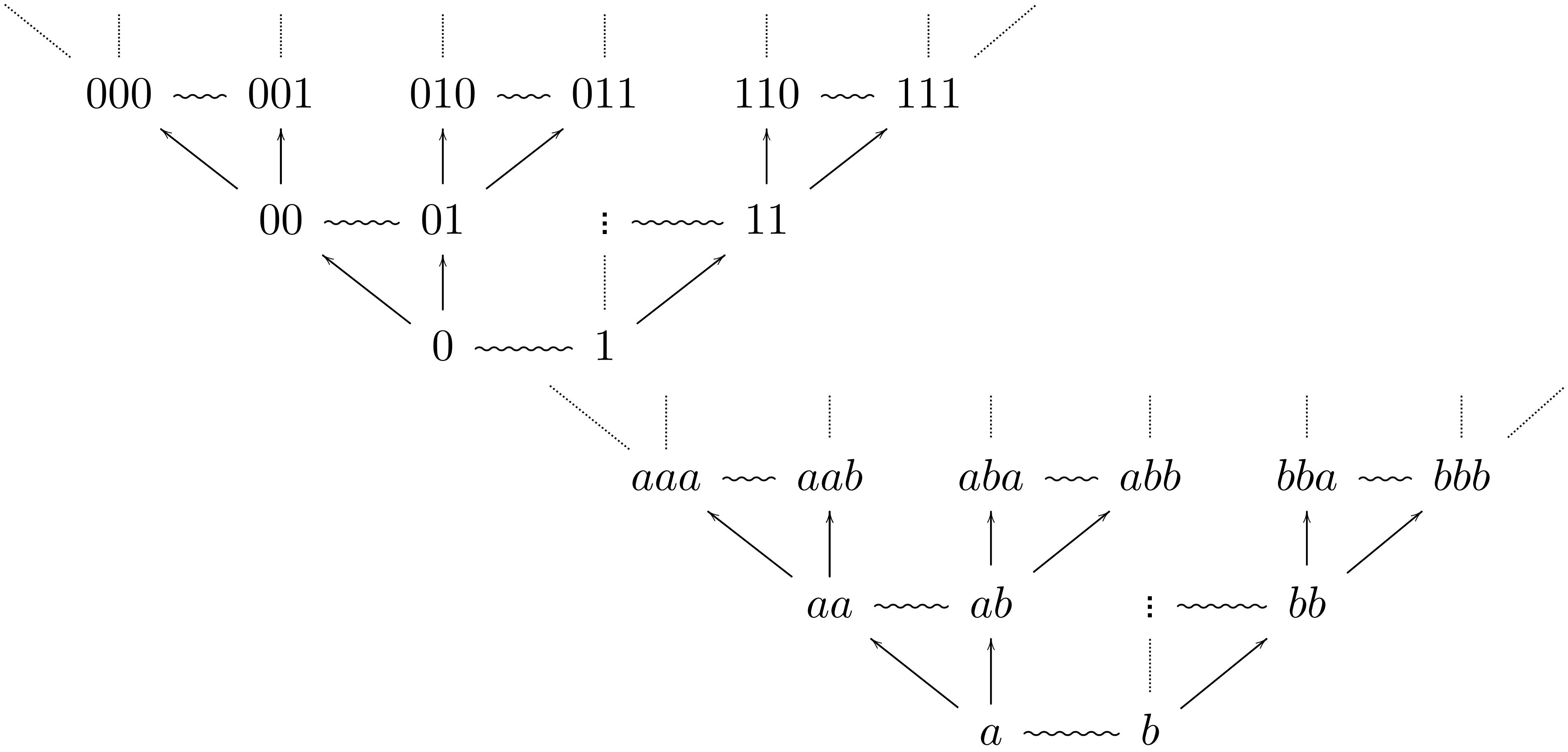
If E is countable, $(\mathcal{C}^\infty(E), \subseteq)$ is a Berry domain (and all such so obtained).
Finite configurations: $\mathcal{C}(E)$.

Example: Streams as event structures



$\rightsquigarrow$ conflict (inconsistency) $\longrightarrow$ causal dependency $\leq$

Simple parallel composition



Maps of event structures

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

In this context, a *simulation 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 $e_1 = e_2$. *('event linearity')*

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

Process constructions on event structures

"Partial synchronous" product: $A \times B$ with projections Π_1 and Π_2, *cf.* CCS synchronized composition where all events of A *can* synchronize with all events of B. ($\mathit{Hard\ to\ construct\ directly\ so\ use\ e.g.\ stable\ families.}$)

Restriction: $E \restriction R$, the restriction of an event structure E to a subset of events R, has events $E' = \{e \in E \mid [e] \subseteq R\}$ with causal dependency and consistency restricted from E.

Synchronized compositions: restrictions of products $A \times B \restriction R$, where R specifies the allowed synchronized and unsynchronized events.

Projection: Let E be an event structure. Let V be a subset of 'visible' events. The *projection* of E on V, $E{\downarrow}V$, has events V with causal dependency and consistency restricted from E.

[$\mathit{Event\ structures\ as\ types}$ and $\mathit{processes?\ Spans}$]

Product—an example

$$
\begin{tikzcd}
b & & & & (b,*) \arrow[rr, squiggly, dash, bend left] & (b,*) \arrow[r, squiggly, dash] & (b,c) \\
a \arrow[u] & \times & c & = & (a,*) \arrow[u] \arrow[r, squiggly, dash] \arrow[urr] & (a,c) \arrow[u] \arrow[r, squiggly, dash] & (*,c) \arrow[u, squiggly, dash]
\end{tikzcd}
$$

Ch 3. STABLE FAMILIES

Stable families—the secret weapon

A *stable family* comprises $\mathcal{F}$, a nonempty family of finite subsets, called *configurations*, satisfying:

Completeness: $\forall Z \subseteq \mathcal{F}.\ Z\!\uparrow\ \Rightarrow \bigcup Z \in \mathcal{F}$;

Stability: $\forall Z \subseteq \mathcal{F}.\ Z \neq \emptyset\ \&\ Z\!\uparrow\ \Rightarrow \bigcap Z \in \mathcal{F}$;

Coincidence-freeness: For all $x \in \mathcal{F}$, $e, e' \in x$ with $e \neq e'$,

$$\exists y \in \mathcal{F}.\ y \subseteq x\ \&\ (e \in y \iff e' \notin y)\,.$$

($Z\!\uparrow$ means $\exists x \in \mathcal{F} \forall z \in Z.\ z \subseteq x$, and expresses the compatibility of Z.)
We call elements of $\bigcup \mathcal{F}$ *events* of $\mathcal{F}$.

Proposition Let x be a configuration of a stable family $\mathcal{F}$. For $e, e' \in x$ define

$$e' \leq_x e \text{ iff } \forall y \in \mathcal{F}.\ y \subseteq x \ \&\ e \in y \Rightarrow e' \in y.$$

When $e \in x$ define the prime configuration

$$[e]_x = \bigcap \{y \in \mathcal{F} \ \mid\ y \subseteq x \ \&\ e \in y\}\ .$$

Then $\leq_x$ is a partial order and $[e]_x$ is a configuration such that

$$[e]_x = \{e' \in x \ \mid\ e' \leq_x e\}.$$

Moreover the configurations $y \subseteq x$ are exactly the down-closed subsets of $\leq_x$.

Proposition Let $\mathcal{F}$ be a stable family. Then, $\mathrm{Pr}(\mathcal{F}) =_{\mathrm{def}} (P, \mathrm{Con}, \leq)$ is an event structure where:

$$P = \{\, [e]_x \;\mid\; e \in x \;\&\; x \in \mathcal{F} \} \;,$$

$$Z \in \mathrm{Con} \text{ iff } Z \subseteq P \;\&\; \bigcup Z \in \mathcal{F} \quad \text{and,}$$

$$p \leq p' \text{ iff } p, p' \in P \;\&\; p \subseteq p' \,.$$

Categories of stable families and event structures

A (partial) map of stable families $f : \mathcal{F} \to \mathcal{G}$ is a partial function f from the events of $\mathcal{F}$ to the events of $\mathcal{G}$ such that for all configurations $x \in \mathcal{F}$,

$$fx \in \mathcal{G} \ \& \ (\forall e_1, e_2 \in x.\ f(e_1) = f(e_2) \Rightarrow e_1 = e_2)\,.$$

Pr is the right adjoint of the "inclusion" functor, taking an event structure E to the stable family $\mathcal{C}(E)$. The unit of the adjunction $E \to \mathrm{Pr}(\mathcal{C}(E))$ takes and event e to the prime configuration $[e] =_{\mathrm{def}} \{e' \in E \mid e' \leq e\}$. The counit $max : \mathcal{C}(\mathrm{Pr}(\mathcal{F})) \to \mathcal{F}$ takes prime configuration $[e]_x$ to e.

Product of stable families

Let $\mathcal{A}$ and $\mathcal{B}$ be stable families with events A and B, respectively. Their product, the stable family $\mathcal{A} \times \mathcal{B}$, has events comprising pairs in
$A \times_* B =_{\text{def}} \{(a, *) \mid a \in A\} \cup \{(a, b) \mid a \in A \ \&\ b \in B\} \cup \{(*, b) \mid b \in B\}$,
the product of sets with partial functions, with (partial) projections π_1 and π_2—treating $*$ as 'undefined'—with configurations

$$x \in \mathcal{A} \times \mathcal{B} \text{ iff}$$

$$x \text{ is a finite subset of } A \times_* B \text{ s.t. } \pi_1 x \in \mathcal{A} \ \&\ \pi_2 x \in \mathcal{B},$$

$$\forall e, e' \in x.\ \pi_1(e) = \pi_1(e') \text{ or } \pi_2(e) = \pi_2(e') \Rightarrow e = e',\ \&$$

$$\forall e, e' \in x.\ e \neq e' \Rightarrow \exists y \subseteq x.\ \pi_1 y \in \mathcal{A} \ \&\ \pi_2 y \in \mathcal{B} \ \&$$
$$(e \in y \iff e' \notin y)\,.$$

Product of event structures

Right adjoints preserve products. Consequently we obtain a product of event structures A and B as

$$A \times B =_{\mathrm{def}} \mathrm{Pr}(\mathcal{C}(A) \times \mathcal{C}(B))$$

and its projections as $\Pi_1 =_{\mathrm{def}} \pi_1 max$ and $\Pi_2 =_{\mathrm{def}} \pi_2 max$.

Hence $\Pi_1 x = \pi_1 \bigcup x$ and $\Pi_2 x = \pi_2 \bigcup x$, for $x \in \mathcal{C}^\infty(A \times B)$.

Ch 4. CONCURRENT GAMES

Concurrent games

Basics

Games and strategies are represented by *event structures with polarity*.

The two polarities $+$ and $-$ express the dichotomy:
player/opponent; process/environment; ally/enemy.

An *event structure with polarity* is one in which all events carry a polarity $+/-$, respected by maps.

Dual, $E^\perp$, of an event structure with polarity E is a copy of the event structure E with a reversal of polarities; $\overline{e} \in E^\perp$ is complement of $e \in E$, and $vice\ versa$.

A (nondeterministic) concurrent *pre-strategy* in game A is a total map $\sigma : S \to A$ of event structures with polarity.

Pre-strategies between games

A *pre-strategy* $\sigma : A \nrightarrow B$ is a total map of event structures with polarity

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

It determines a span of event structures with polarity

$$\begin{tikzcd} & S \arrow[dl, "\sigma_1"'] \arrow[dr, "\sigma_2"] & \\ A^{\perp} & & B \end{tikzcd}$$

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.

Concurrent 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 $\leq_{\mathbb{C}\mathbb{C}_A}$ given as the transitive closure of the relation

$$\leq_{A^\perp \parallel A} \cup \{(\overline{c}, c) \mid c \in A^\perp \parallel A \ \& \ pol_{A^\perp \parallel A}(c) = +\}$$

where $\overline{c} \leftrightarrow c$ is the natural correspondence between $A^\perp$ and A. A finite subset is consistent iff its down-closure in consistent in $A^\perp \| A$. The map γ_A is the identity on the common underlying set of events. Then,

$$x \in \mathcal{C}(\mathbb{C}\mathbb{C}_A) \text{ iff } x \in \mathcal{C}(A^\perp \parallel A) \ \& \ \forall c \in x.\ pol_{A^\perp \parallel A}(c) = + \ \Rightarrow \ \overline{c} \in x\,.$$

Copy-cat—an example

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

$$\begin{tikzcd}
A^{\perp} & & & A \\
\overline{a}_2 & \ominus \arrow[r, dashed] & \oplus & a_2 \\
\overline{a}_1 & \oplus \arrow[u] & \ominus \arrow[l, dashed] \arrow[u] & a_1
\end{tikzcd}$$

Composing pre-strategies

Two pre-strategies $\sigma : A \nrightarrow B$ and $\tau : B \nrightarrow C$ as spans:

$$
\begin{tikzcd}
 & S \arrow[dl, "\sigma_1"'] \arrow[dr, "\sigma_2"] & & & T \arrow[dl, "\tau_1"'] \arrow[dr, "\tau_2"] & \\
A^{\perp} & & B & B^{\perp} & & C\,.
\end{tikzcd}
$$

Their composition

$$
\begin{tikzcd}
 & T \odot S \arrow[dl, "(\tau\odot\sigma)_1"'] \arrow[dr, "(\tau\odot\sigma)_2"] & \\
A^{\perp} & & C
\end{tikzcd}
$$

where $T \odot S =_{\mathrm{def}} (S \times T \upharpoonright \textbf{Syn}) \downarrow \textbf{Vis}$ where ...

$$
\begin{tikzcd}
 & & & S \times T \arrow[dll, "\Pi_1"'] \arrow[drr, "\Pi_2"] & & & \\
 & S \arrow[dl, "\sigma_1"'] \arrow[dr, "\sigma_2"] & & & & T \arrow[dl, "\tau_1"'] \arrow[dr, "\tau_2"] & \\
A^{\perp} & & B & & B^{\perp} & & C
\end{tikzcd}
$$

Their composition: $T \odot S =_{\mathrm{def}} (S \times T \upharpoonright \textbf{Syn}) \downarrow \textbf{Vis}$ where

$$
\begin{aligned}
\textbf{Syn} \;=\; & \{p \in S \times T \;\mid\; \sigma_1 \Pi_1(p) \text{ is defined } \& \ \Pi_2(p) \text{ is undefined}\} \cup \\
& \{p \in S \times T \;\mid\; \sigma_2 \Pi_1(p) = \overline{\tau_1 \Pi_2(p)} \text{ with both defined}\} \cup \\
& \{p \in S \times T \;\mid\; \tau_2 \Pi_2(p) \text{ is defined } \& \ \Pi_1(p) \text{ is undefined}\}\,, \\
\textbf{Vis} \;=\; & \{p \in S \times T \upharpoonright \textbf{Syn} \;\mid\; \sigma_1 \Pi_1(p) \text{ is defined}\} \cup \\
& \{p \in S \times T \upharpoonright \textbf{Syn} \;\mid\; \tau_2 \Pi_2(p) \text{ is defined}\}\,.
\end{aligned}
$$

Theorem characterizing concurrent strategies

Receptivity $\sigma : S \to A^{\perp} \parallel B$ is *receptive* when $\sigma(x) \mathrel{-\!\!\!\subset}^{-} y$ implies there is a *unique* $x' \in \mathcal{C}(S)$ such that $x \mathrel{-\!\!\!\subset} x'$ & $\sigma(x') = y$.

$$\begin{tikzcd}
x \arrow[r, dashed, "\mathrel{-\!\!\!\subset}" description, phantom] & x' \arrow[d, dashed, mapsto] \\
\sigma(x) \arrow[from=u, mapsto] \arrow[r, phantom, "\mathrel{-\!\!\!\subset}^{-}"] & y
\end{tikzcd}$$

Innocence $\sigma : S \to A^{\perp} \parallel B$ is *innocent* when it is

$+$-*Innocence:* If $s \rightarrowtriangle s'$ & $pol(s) = +$ then $\sigma(s) \rightarrowtriangle \sigma(s')$ and

$-$-*Innocence:* If $s \rightarrowtriangle s'$ & $pol(s') = -$ then $\sigma(s) \rightarrowtriangle \sigma(s')$.

[$\rightarrowtriangle$ *stands for immediate causal dependency*]

Theorem Receptivity and innocence are necessary and sufficient for copy-cat to act as identity w.r.t. composition: $\sigma \odot \gamma_A \cong \sigma$ and $\gamma_B \odot \sigma \cong \sigma$ for all $\sigma : A \mathrel{+\!\!\!\!\rightarrow} B$.

Idea of the proof

Necessity Copy-cats γ_A are receptive and innocent. If $\sigma \cong \gamma_B \odot \sigma \odot \gamma_A$ then $\sigma : A \mathrel{+\!\!\!\!\rightarrow} B$ inherits receptivity and innocence from that of copy-cat.

Sufficiency A key lemma for constructing $\theta : \sigma \odot \gamma_A \cong \sigma$ where $\sigma : A \mathrel{+\!\!\!\!\rightarrow} B$:
For σ total, receptive and $-$-innocent, $p : \mathcal{C}(V) \to \mathcal{C}(S)$ monotone,

$$
\begin{tikzcd}
V \arrow[r, dotted, "p"] \arrow[dr, "\upsilon"'] & S \arrow[d, "\sigma"] \arrow[dl, phantom, "{}^{-}\!\supseteq", pos=0.7] \\
& C\,.
\end{tikzcd}
\quad \text{implies} \quad \exists!\theta \text{ s.t.} \quad
\begin{tikzcd}
V \arrow[r, dotted, "p"'{name=P}] \arrow[r, dashed, bend left=60, "\theta"{name=T}] \arrow[dr, "\upsilon"'] & S \arrow[d, "\sigma"] \arrow[dl, phantom, "{}^{-}\!\supseteq", pos=0.7] \\
& C
\arrow[from=T, to=P, phantom, "\rotatebox{90}{$\subseteq^{-}$}"]
\end{tikzcd}
\quad \& \quad \sigma\theta = \upsilon.
$$

Idea of the proof (cont)

Instantiating $p : \mathcal{C}(S \odot \mathbb{C}\!\!\mathbb{C}_A) \to \mathcal{C}(S)$ to the function $p(x) = \Pi_2[x]$,
where $[x]$ is the down-closure in the synchronized composition before projection:

$$
\begin{tikzcd}
S \odot \mathbb{C}\!\!\mathbb{C}_A \arrow[r, dotted, "p"] \arrow[r, dashed, bend left=50, "\theta"] \arrow[dr, "\sigma \odot \gamma_A"'] \arrow[dr, phantom, shift left=4ex, "\supseteq"] & S \arrow[d, "\sigma"] \\
& A^{\perp} \| B
\end{tikzcd}
$$

The bicategory of concurrent games

Definition A *strategy* is a receptive, innocent pre-strategy.

$\rightsquigarrow$ A bicategory, $\mathbf{Games}$, whose

objects are event structures with polarity—the games,

arrows are strategies $\sigma : A \twoheadrightarrow B$

2-cells are maps of spans.

The vertical composition of 2-cells is the usual composition of maps of spans. Horizontal composition is given by the composition of strategies $\odot$ (which extends to a functor on 2-cells via the functoriality of synchronized composition).

Duality

A (pre-)strategy $\sigma : A \mathrel{+\!\!\!\!\!\rightarrow} B$ corresponds to a dual (pre-)strategy $\sigma^\perp : B^\perp \mathrel{+\!\!\!\!\!\rightarrow} A^\perp$:

$$
\begin{tikzcd}
 & S \arrow[dl, "\sigma_1"'] \arrow[dr, "\sigma_2"] & & \longleftrightarrow & & S \arrow[dl, "\sigma_2"'] \arrow[dr, "\sigma_1"] & \\
A^\perp & & B & & (B^\perp)^\perp & & A^\perp
\end{tikzcd}
$$

In fact, the bicategory of concurrent games is *compact-closed*, in particular $(A \| B)^\perp \cong A^\perp \| B^\perp$ and it has a *trace* or feedback operation.

With the addition of the extra features of winning conditions or pay-off, compact closure will weaken to $*$-autonomy.

Strategies—alternative description 1

A strategy S in a game A comprises a total map of event structures with polarity $\sigma : S \to A$ such that

(i) $\sigma x \overset{a}{\longrightarrow\!\!\subset}$ & $pol_A(a) = - \Rightarrow \exists! s \in S.\ x \overset{s}{\longrightarrow\!\!\subset}$ & $\sigma(s) = a$, for all $x \in \mathcal{C}(S)$, $a \in A$.

(ii)(+) If $x \overset{e}{\longrightarrow\!\!\subset} x_1 \overset{e'}{\longrightarrow\!\!\subset}$ & $pol_S(e) = +$ in $\mathcal{C}(S)$ and $\sigma x \overset{\sigma(e')}{\longrightarrow\!\!\subset}$ in $\mathcal{C}(A)$, then $x \overset{e'}{\longrightarrow\!\!\subset}$ in $\mathcal{C}(S)$.

(ii)(−) If $x \overset{e}{\longrightarrow\!\!\subset} x_1 \overset{e'}{\longrightarrow\!\!\subset}$ & $pol_S(e') = -$ in $\mathcal{C}(S)$ and $\sigma x \overset{\sigma(e')}{\longrightarrow\!\!\subset}$ in $\mathcal{C}(A)$, then $x \overset{e'}{\longrightarrow\!\!\subset}$ in $\mathcal{C}(S)$.

Notation $x \overset{e}{\longrightarrow\!\!\subset} y$ iff $x \cup \{e\} = y$ & $e \notin x$, for configurations x, y, event e.
$x \overset{e}{\longrightarrow\!\!\subset}$ iff $\exists y.\ x \overset{e}{\longrightarrow\!\!\subset} y$.

Strategies—alternative description 2

A strategy S in a game A comprises a total map of event structures with polarity $\sigma : S \to A$ 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.*

$$\begin{tikzcd} x \arrow[r, dotted, "\subseteq"] \arrow[d, mapsto, "\sigma"'] & x' \arrow[d, dotted, mapsto, "\sigma"] \\ \sigma x \arrow[r, phantom, "\subseteq^-"] & y , \end{tikzcd}$$

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.*

$$\begin{tikzcd} x' \arrow[r, dotted, "\subseteq"] \arrow[d, dotted, mapsto, "\sigma"'] & x \arrow[d, mapsto, "\sigma"] \\ y \arrow[r, phantom, "\subseteq^+"] & \sigma x . \end{tikzcd}$$

Strategies—alternative description 3

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 $((\mathcal{C}(A), \sqsubseteq_A), \supseteq^-, \subseteq^+)$, $i.e.$ $\exists! z.$

$$\begin{tikzcd} & & y \\ x \arrow[urr, dotted, no head, "\sqsubseteq"] & \supseteq^- & z \arrow[u, phantom, "\cup\!\mid^{+}"] \end{tikzcd}.$$

Proposition $z \in \mathcal{C}(\mathbb{C}_A)$ iff $z_2 \sqsubseteq_A z_1$.

Theorem Strategies $\sigma : S \to A$ correspond to discrete fibrations

$$\sigma" : (\mathcal{C}(S), \sqsubseteq_S) \to (\mathcal{C}(A), \sqsubseteq_A)\,, \quad i.e. \quad \exists! x'. \quad \begin{tikzcd} x' \arrow[d, dotted, mapsto, "\sigma\text{"}"'] \arrow[r, dotted, no head, "\sqsubseteq_S"] & x \arrow[d, mapsto, "\sigma\text{"}"] \\ y & \sqsubseteq_A \sigma"(x) \end{tikzcd},$$

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

Strategies—alternative description 4, via just $+$-moves

A strategy $\sigma : S \to A$ determines
$$\begin{tikzcd} S \arrow[r, "q"'] \arrow[d, "\sigma"'] & S^+ \arrow[dl, dotted, "d"] \arrow[dl, phantom, "{}^-\!\supseteq", near end, shift right=3] \\ A \end{tikzcd}$$
where q is projection and $d : \mathcal{C}(S) \to \mathcal{C}(A)$ s.t. $d(x) = \sigma[x]$. Universal property showing d determines σ:

$$\begin{tikzcd} U \arrow[r, "g"] \arrow[d, "f"'] & S^+ \arrow[dl, dotted, "d"] \arrow[dl, phantom, "{}^-\!\supseteq", near end, shift right=3] \\ A \end{tikzcd} \Rightarrow \exists!\theta \text{ s.t. } \begin{tikzcd} U \arrow[r, dashed, "\theta"] \arrow[rr, bend left, "g"] \arrow[dr, bend right, "f"'] & S \arrow[r, "q"'] \arrow[d, "\sigma"'] & S^+ \arrow[dl, dotted, "d"] \arrow[dl, phantom, "{}^-\!\supseteq", near end, shift right=3] \\ & A \end{tikzcd} \quad \& \ \sigma\theta = f \ \ \& \ q\theta = g.$$

Ch 5. DETERMINISTIC STRATEGIES

Deterministic strategies

Say an event structures with polarity S is *deterministic* iff

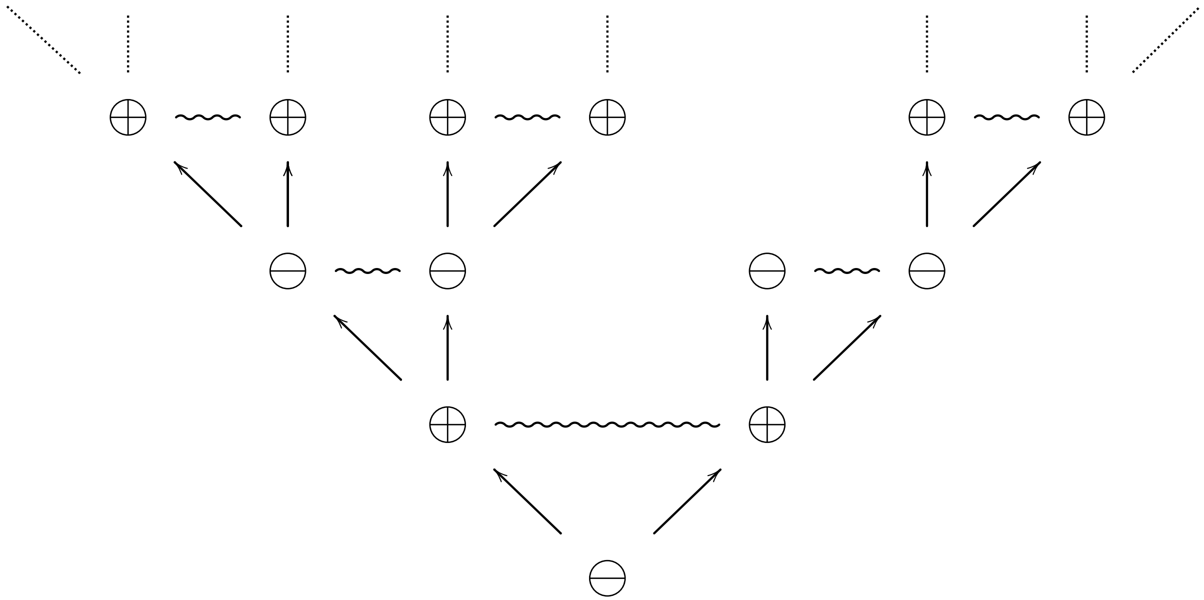
$$\forall X \subseteq_{\rm fin} S.\ Neg[X] \in {\rm Con}_S \Rightarrow X \in {\rm Con}_S\,,$$

where $Neg[X] =_{\rm def} \{s' \in S \mid \exists s \in X.\ pol_S(s') = - \ \&\ s' \leq s\}$.
Say a strategy $\sigma : S \to A$ is deterministic if S is deterministic.

Proposition An event structure with polarity S is deterministic iff $x \stackrel{s}{\longrightarrow\!\!\!\!\!\subset} \ \&\ x \stackrel{s'}{\longrightarrow\!\!\!\!\!\subset} \ \&\ pol_S(s) = +$ implies $x \cup \{s, s'\} \in \mathcal{C}(S)$, for all $x \in \mathcal{C}(S)$.

Notation $x \stackrel{e}{\longrightarrow\!\!\!\!\!\subset} y$ iff $x \cup \{e\} = y \ \&\ e \notin x\,,$ for configurations x, y, event e.
$x \stackrel{e}{\longrightarrow\!\!\!\!\!\subset}$ iff $\exists y.\ x \stackrel{e}{\longrightarrow\!\!\!\!\!\subset} y$.

Example: a tree-like game



$\sim\!\sim\!\sim$ conflict (inconsistency) $\longrightarrow$ immediate causal dependency

$\oplus$ Player move $\ominus$ Opponent move

Nondeterministic copy-cats

(i) Take A to consist of two +ve events and one −ve event, with any two but not all three events consistent. The construction of $\mathbb{C}\mathrm{C}_A$:

$$
\begin{array}{ccccc}
 & \ominus & \rightarrowtriangle & \oplus & \\
A^{\perp} & \ominus & \rightarrowtriangle & \oplus & A \\
 & \oplus & \leftarrowtriangle & \ominus &
\end{array}
$$

(ii) Take A to consist of two events, one +ve and one −ve event, inconsistent with each other. The construction $\mathbb{C}\mathrm{C}_A$:

$$
\begin{array}{ccccc}
A^{\perp} & \ominus & \rightarrowtriangle & \oplus & A \\
 & \oplus & \leftarrowtriangle & \ominus &
\end{array}
$$

Lemma Let A be an event structure with polarity. The copy-cat strategy γ_A is deterministic iff A satisfies

$$\forall x \in \mathcal{C}(A).\ x \xrightarrow{a}\!\subset \ \&\ x \xrightarrow{a'}\!\subset \ \&\ pol_A(a) = + \ \&\ pol_A(a') = - \\ \Rightarrow x \cup \{a, a'\} \in \mathcal{C}(A)\,. \qquad (\textbf{Race-free})$$

Lemma The composition $\tau \odot \sigma$ of two deterministic strategies σ and τ is deterministic.

Lemma A deterministic strategy $\sigma : S \to A$ is injective on configurations (equivalently, $\sigma : S \rightarrowtail A$).

$\rightsquigarrow$ sub-bicategory of race-free games and deterministic strategies, equivalent to an order-enriched category.

Theorem A subfamily $F \subseteq \mathcal{C}(A)$ has the form $\sigma\mathcal{C}(S)$ for a deterministic strategy $\sigma : S \to A$, iff

reachability: $\emptyset \in F$ and if $x \in F$, $\emptyset \stackrel{a_1}{\longrightarrow\!\!\!\subset} x_1 \stackrel{a_2}{\longrightarrow\!\!\!\subset} \cdots \stackrel{a_k}{\longrightarrow\!\!\!\subset} x_k = x$ within F;

determinacy: If $x \stackrel{a}{\longrightarrow\!\!\!\subset}$ and $x \stackrel{a'}{\longrightarrow\!\!\!\subset}$ in F with $pol_A(a) = +$, then $x \cup \{a, a'\} \in F$;

receptivity: If $x \in F$ and $x \stackrel{a}{\longrightarrow\!\!\!\subset}$ in $\mathcal{C}(A)$ and $pol_A(a) = -$, then $x \cup \{a\} \in F$;

+-innocence: If $x \stackrel{a}{\longrightarrow\!\!\!\subset} x_1 \stackrel{a'}{\longrightarrow\!\!\!\subset}$ & $pol_A(a) = +$ in F & $x \stackrel{a'}{\longrightarrow\!\!\!\subset}$ in $\mathcal{C}(A)$, then $x \stackrel{a'}{\longrightarrow\!\!\!\subset}$ in F (receptivity implies $-$-innocence);

1-stable: If $x_1 \stackrel{a}{\longrightarrow\!\!\!\subset} x$ and $x_2 \stackrel{b}{\longrightarrow\!\!\!\subset} x$ in F, then $x_1 \cap x_2 \in F$.

Ch 6. Games people play

Stable spans, profunctors and stable functions The sub-bicategory of $\mathbf{Games}$ where the events of games are purely $+$ve is equivalent to the bicategory of stable spans:

$$
\begin{tikzcd}
& S \arrow[dl, "\sigma_1"'] \arrow[dr, "\sigma_2"] & & \longleftrightarrow & & S^+ \arrow[dl, "\sigma_1^-"'] \arrow[dr, "\sigma_2^+"] & \\
A^\perp & & B & & A & & B\,,
\end{tikzcd}
$$

where S^+ is the projection of S to its $+$ve events; σ_2^+ is the restriction of σ_2 to S^+ is rigid; σ_2^- is a *demand map* taking $x \in \mathcal{C}(S^+)$ to $\sigma_1^-(x) = \sigma_1[x]$.

Composition of stable spans coincides with composition of their associated profunctors. The feedback operation of nondeterministic dataflow is obtained as a special case of the trace on concurrent games.

When deterministic (and event structures are countable) we obtain a sub-bicategory equivalent to Berry's **dI-domains and stable functions**.

Ingenuous strategies Deterministic concurrent strategies coincide with the *receptive ingenuous* strategies of and Melliès and Mimram.

Closure operators A deterministic strategy $\sigma : S \to A$ determines a closure operator φ on $\mathcal{C}^\infty(S)$: for $x \in \mathcal{C}^\infty(S)$,

$$\varphi(x) = x \cup \{s \in S \ \mid \ pol(s) = + \ \& \ Neg[\{s\}] \subseteq x\} \, .$$

The closure operator φ on $\mathcal{C}^\infty(S)$ induces a *partial* closure operator φ_p on $\mathcal{C}^\infty(A)$ and in turn a closure operator $\varphi_p^\top$ on $\mathcal{C}^\infty(A)^\top$ of Abramsky and Melliès.

Simple games *"Simple games"* of game semantics arise when we restrict $\mathbf{Games}$ to objects and deterministic strategies which are 'tree-like'—alternating polarities, with conflicting branches, beginning with opponent moves.

Conway games tree-like, but where only strategies need alternate and begin with opponent moves.

Ch 7. WINNING WAYS

Winning conditions

A *game with winning conditions* comprises

$$G = (A, W)$$

where A is an event structure with polarity and $W \subseteq \mathcal{C}^\infty(A)$ consists of the *winning configurations* for Player.

Define the *losing conditions* to be $L =_{\mathrm{def}} \mathcal{C}^\infty(A) \setminus W$.

Winning strategies

Let $G = (A, W)$ be a game with winning conditions.

A strategy in G is a strategy in A.

A strategy $\sigma : S \to A$ in G is *winning (for Player)* if $\sigma x \in W$, $i.e.$ $\sigma x \notin L$, for all $+$-maximal configurations $x \in \mathcal{C}^{\infty}(S)$.
[A configuration x is $+$-maximal if whenever $x \stackrel{s}{\longrightarrow\!\!\!\!\subset}$ then the event s has $-$ve polarity.]

A winning strategy prescribes moves for Player to avoid ending in a losing configuration, no matter what the activity or inactivity of Opponent.

Characterization via counter-strategies

Informally, a strategy is winning for Player if any play against a counter-strategy *of Opponent* results in *a win for Player.*

A *counter-strategy*, *i.e.* a strategy of Opponent, in a game A is a strategy in the dual game, so $\tau : T \to A^{\perp}$.

What are the *results* $\langle \sigma, \tau \rangle$ of playing strategy σ against counter-strategy τ?

Note $\sigma : \emptyset \dashrightarrow A$ and $\tau : A \dashrightarrow \emptyset$ ...

Composition of pre-strategies without hiding

$$
\begin{tikzcd}[column sep=small]
 & & & S \times T \upharpoonright \mathbf{Syn} \arrow[dll, "\Pi_1"'] \arrow[drr, "\Pi_2"] & & & \\
 & S \arrow[dl, "\sigma_1"'] \arrow[dr, "\sigma_2"] & & & & T \arrow[dl, "\tau_1"'] \arrow[dr, "\tau_2"] & \\
A^\perp & & B & & B^\perp & & C
\end{tikzcd}
$$

where

$$
\begin{aligned}
\textbf{Syn} \;=\; & \{p \in S \times T \mid \sigma_1\Pi_1(p) \text{ is defined } \& \ \Pi_2(p) \text{ is undefined}\} \cup \\
& \{p \in S \times T \mid \sigma_2\Pi_1(p) = \overline{\tau_1\Pi_2(p)} \text{ with both defined}\} \cup \\
& \{p \in S \times T \mid \tau_2\Pi_2(p) \text{ is defined } \& \ \Pi_1(p) \text{ is undefined}\}\,.
\end{aligned}
$$

Special case

$$
\begin{tikzcd}
 & & S \times T \upharpoonright \mathbf{Syn} \arrow[dl, "\Pi_1"'] \arrow[dr, "\Pi_2"] & & \\
 & S \arrow[dl] \arrow[d, "\sigma"] & & T \arrow[d, "\tau"'] \arrow[dr] & \\
\emptyset & A & & A^{\perp} & \emptyset
\end{tikzcd}
$$

where

$$\textbf{Syn} \;=\; \{p \in S \times T \;\mid\; \sigma\Pi_1(p) = \overline{\tau\Pi_2(p)} \text{ with both defined}\}\,.$$

Define **results**, $\langle \sigma, \tau \rangle =_{\mathrm{def}} \{\sigma\Pi_1 z \;\mid\; z \text{ is maximal in } \mathcal{C}^{\infty}(S \times T \upharpoonright \mathbf{Syn})\}\,.$

Characterization of winning strategies

Lemma Let $\sigma : S \to A$ be a strategy in a game (A, W). The strategy σ is winning for Player iff $\langle \sigma, \tau \rangle \subseteq W$ for all (deterministic) strategies $\tau : T \to A^\perp$.

Its proof uses a key lemma:

Lemma Let $\sigma : S \to A^\perp \| B$ and $\tau : B^\perp \| C$ be receptive pre-strategies. Then,

$$z \in \mathcal{C}^\infty(S \times T \restriction \mathbf{Syn}) \text{ is } + \text{-maximal iff}$$
$$\Pi_1 z \in \mathcal{C}^\infty(S) \text{ is } + \text{-maximal } \& \ \Pi_2 z \in \mathcal{C}^\infty(T) \text{ is } + \text{-maximal.}$$

Ex.1. $\ominus \leftrightsquigarrow \oplus$ has a winning strategy only if W comprises all configurations.

Ex.2.

$$\begin{tikzcd} \ominus \arrow[r, no head, squiggly] & \oplus \\ & \oplus \arrow[ul] \arrow[u] \end{tikzcd}$$

the empty strategy is winning if $\emptyset \in W$.

Ex.3. $\ominus \qquad \ominus \qquad \oplus$, with $x \in W$ iff $pol\, x \cap \{-\} \neq \emptyset \Rightarrow pol\, x \cap \{+\} \neq \emptyset$, has a winning nondeterministic strategy, but no winning deterministic strategy.

Ex.4. $\ominus \qquad \oplus \longrightarrow \oplus \longrightarrow \cdots \longrightarrow \oplus \longrightarrow \cdots$ with $x \in W$ iff $(\ominus \in x \Leftrightarrow x \text{ finite})$ has no winning strategy or counterstrategy.

Ex.5.

$$\begin{tikzcd} & \oplus \arrow[d, no head, squiggly] & \oplus \arrow[d, no head, squiggly] & \cdots & \oplus \arrow[d, no head, squiggly] & \cdots \\ \ominus & \oplus \arrow[r] & \oplus \arrow[r] & \cdots \arrow[r] & \oplus \arrow[r] & \cdots \end{tikzcd}$$

with $x \in W$ iff $(\ominus \in x \Leftrightarrow x \text{ finite})$ has a winning strategy.

Operations on games with winning conditions

Dual $G^\perp = (A^\perp, W_{G^\perp})$ where, for $x \in \mathcal{C}^\infty(A)$,

$$x \in W_{G^\perp} \text{ iff } \overline{x} \notin W_G \,.$$

Parallel composition For $G = (A, W_G)$, $H = (B, W_H)$,

$$G \| H =_{\mathrm{def}} (A \| B,\ W_G \| \mathcal{C}^\infty(B) \cup \mathcal{C}^\infty(A) \| W_H)$$

where $X \| Y = \{\{1\} \times x \cup \{2\} \times y \mid x \in X \;\&\; y \in Y\}$ when X and Y are subsets of configurations. To win is to win in either game. Unit of $\|$ is $(\emptyset, \emptyset)$.

Derived operations

Tensor Defining $G \otimes H =_{\rm def} (G^\perp \| H^\perp)^\perp$ we obtain a game where to win is to win in both games G and H—so to lose is to lose in either game. More explicitly,

$$(A, W_A) \otimes (B, W_B) =_{\rm def} (A \| B,\ W_A \| W_B)\,.$$

The unit of $\otimes$ is $(\emptyset, \{\emptyset\})$.

Function space With $G \multimap H =_{\rm def} G^\perp \| H$ a win in $G \multimap H$ is a win in H conditional on a win in G:

Proposition Let $G = (A, W_G)$ and $H = (B, W_H)$ be games with winning conditions. Write $W_{G \multimap H}$ for the winning conditions of $G \multimap H$. For $x \in \mathcal{C}^\infty(A^\perp \| B)$,

$$x \in W_{G \multimap H} \ \text{ iff } \ \overline{x_1} \in W_G \Rightarrow x_2 \in W_H\,.$$

The bicategory of winning strategies

Lemma Let σ be a winning strategy in $G^{\perp} \| H$ and τ be a winning strategy in $H^{\perp} \| K$. Their composition $\tau \odot \sigma$ is a winning strategy in $G^{\perp} \| K$.

But copy-cat need not be winning: Let A consist of $\oplus \sim\!\sim\!\sim \ominus$. The event structure $\mathbb{C}\!\mathbb{C}_A$:

$$\begin{array}{ccccc} A^{\perp} & \ominus & \longrightarrow & \oplus & A \\ & \wr & & \wr & \\ & \oplus & \longleftarrow & \ominus & \end{array}$$

Taking $x = \{\ominus, \ominus\}$, $\overline{x}_1 \in W$ while $x_2 \notin W$.

A robust sufficient condition for copy-cat to be winning: the game is race-free. The notes give a necessary and sufficient condition.
$\rightsquigarrow$ bicategory of games with winning strategies.

Applications

Total strategies: To pick out a subcategory of *total* strategies (where Player can always answer Opponent) within simple games.

Determinacy of concurrent games: A necessary and sufficient condition on a well-founded game A for (A, W) to be determined for all winning conditions: that A is race-free. (A game A is well-founded if all its configurations are finite)

A concurrent-game semantics for PC: W.r.t. a model, a closed formula of Predicate Calculus denotes a concurrent game which has a winning strategy iff the formula is true.

There are a growing number—see Ch. 9.

Ch 8, 9. APPLICATIONS AND EXTENSIONS

Predicate Calculus

The syntax for predicate calculus: formulae are given by

$$\phi, \psi, \cdots ::= R(x_1, \cdots, x_k) \ \mid\ \phi \wedge \psi \ \mid\ \phi \vee \psi \ \mid\ \neg\phi \ \mid\ \exists x.\ \phi \ \mid\ \forall x.\ \phi$$

where R ranges over basic relation symbols of a fixed arity and $x, x_1, x_2, \cdots, x_k$ over variables.

A model M for the predicate calculus comprises a non-empty universe of values V_M and an interpretation for each of the relation symbols as a relation of appropriate arity on V_M. Write

$$\rho \models_M \phi$$

iff formula ϕ is true in M w.r.t. environment ρ; we take an environment to be a function from variables to values.

As concurrent games

The denotation as a game is defined by structural induction:

$$[\![R(x_1, \cdots, x_k)]\!]_M \rho = \begin{cases} (\emptyset, \{\emptyset\}) & \text{if } \rho \models_M R(x_1, \cdots, x_k)\,, \\ (\emptyset, \emptyset) & \text{otherwise.} \end{cases}$$

$$[\![\phi \wedge \psi]\!]_M \rho = [\![\phi]\!]_M \rho \otimes [\![\psi]\!]_M \rho$$

$$[\![\phi \vee \psi]\!]_M \rho = [\![\phi]\!]_M \rho \| [\![\psi]\!]_M \rho$$

$$[\![\neg \phi]\!]_M \rho = ([\![\phi]\!]_M \rho)^\perp$$

$$[\![\exists x.\ \phi]\!]_M \rho = \bigoplus_{v \in V_M} [\![\phi]\!]_M \rho[v/x]$$

$$[\![\forall x.\ \phi]\!]_M \rho = \bigodot_{v \in V_M} [\![\phi]\!]_M \rho[v/x]\,.$$

Prefixed sums

The prefixed game $\oplus.(A, W)$ comprises the event structure with polarity $\oplus.A$ in which all the events of A are made to causally depend on a fresh +ve event $\oplus$. Its winning conditions are those configurations $x \in \mathcal{C}^\infty(\oplus.A)$ of the form $\{\oplus\} \cup y$ for some $y \in W$.

The game $\bigoplus_{v \in V}(A_v, W_v)$ has underlying event structure with polarity the sum (=coproduct) $\sum_{v \in V} \oplus.A_v$ with a configuration winning iff it is the image of a winning configuration in a component under the injection to the sum. The game $\bigominus_{v \in V} G_v$ is defined dually.

Theorem For all predicate-calculus formulae ϕ and environments ρ, $\rho \models_M \phi$ iff the game $[\![\phi]\!]_M \rho$ has a winning strategy.

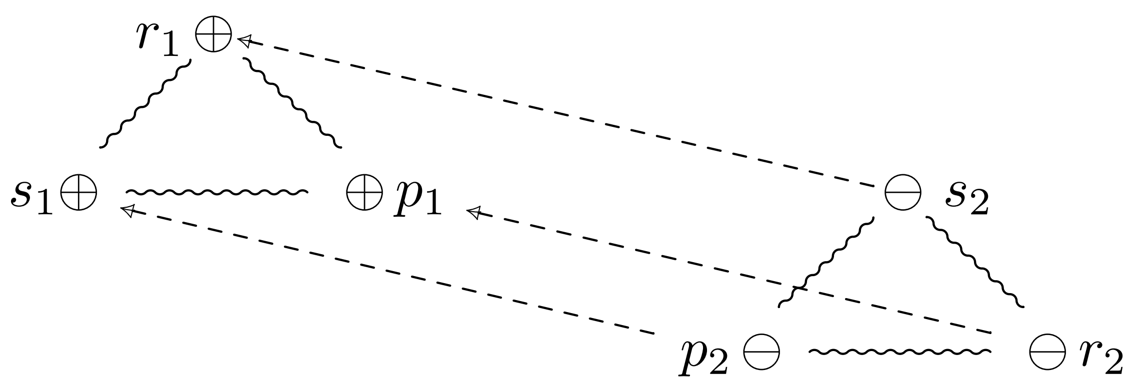
Games with imperfect information

The game "rock, scissors, paper":

The *losing* configurations (for Player):

$$\{s_1, r_2\},\ \{p_1, s_2\},\ \{r_1, p_2\}$$

A cheating strategy



Games with imperfect information

A fixed preorder of *access levels* $(\Lambda, \preceq)$.

An Λ-game (G, l) comprises a game $G = (A, W, L)$ with winning/losing conditions together with a *level function* $l : A \to \Lambda$ such that

$$a \leq_A a' \Rightarrow l(a) \preceq l(a')$$

for all $a, a' \in A$. A Λ-strategy in the Λ-game (G, l) is a strategy $\sigma : S \to A$ for which

$$s \leq_S s' \Rightarrow l\sigma(s) \preceq l\sigma(s')$$

for all $s, s' \in S$.

The bicategory of Λ-games

For a Λ-game (G, l_G), define its dual $(G, l_G)^\perp$ to be $(G^\perp, l_{G^\perp})$ where $l_{G^\perp}(\overline{a}) = l_G(a)$, for a an event of G.

For Λ-games (G, l_G) and (H, l_H), define their parallel composition $(G, l_G)\|(H, l_H)$ to be $(G\|H, l_{G\|H})$ where $l_{G\|H}((1, a)) = l_G(a)$, for a an event of G, and $l_{G\|H}((2, b)) = l_H(b)$, for b an event of H.

A strategy between Λ-games from (G, l_G) to (H, l_H) is a strategy in $(G, l_G)^\perp\|(H, l_H)$.

Proposition (i) Let (G, l_G) be a Λ-game where G satisfies **(Cwins)**. The copy-cat strategy on G is a Λ-strategy. (ii) The composition of Λ-strategies is a Λ-strategy.

Application: Hintikka's IF Logic

Strategies on categories

A rooted factorization system $(\mathbb{C}, \mathcal{L}, \mathcal{R}, 0)$ comprises a small category $\mathbb{C}$ on which there is a factorization system $(\mathbb{C}, \mathcal{L}, \mathcal{R})$ with an object 0 s.t.

$$0 \leftarrow_L \cdot \rightarrow_R \cdots \leftarrow_L \cdot \rightarrow_R c$$

for all objects c in $\mathbb{C}$, and ...

Example: $((\mathcal{C}(A), \sqsubseteq), \supseteq^-, \subseteq^+, \emptyset)$

Strategies

A strategy on a rooted factorization system $(\mathbb{C}, \mathcal{L}, \mathcal{R}, 0)$ is a fibration from another rooted factorization system which preserves $\mathcal{L}$, $\mathcal{R}$ maps and 0.

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

Operations

$(\mathbb{C}, \mathcal{L}, \mathcal{R}, 0)^\perp =_{\mathrm{def}} (\mathbb{C}^{\mathrm{op}}, \mathcal{R}^{\mathrm{op}}, \mathcal{L}^{\mathrm{op}}, 0)$

$(\mathbb{B}, \mathcal{L}_B, \mathcal{R}B, 0_B) \| (\mathbb{C}, \mathcal{L},_C \mathcal{R}_C, 0_C) =_{\mathrm{def}} (\mathbb{B} \times \mathbb{C}, \mathcal{L}_B \times \mathcal{L}_C, \mathcal{R}_B \times \mathcal{R}_C, (0_B, 0_C))$

Composition: 'reachable part of' profunctor composition.

Extensions

Borel determinacy of race-free, bounded-concurrent games

Games with pay-off

Games with back-tracking, via games with symmetry

Probabilistic and quantum games

A language for strategies, regarded as concurrent processes

See `www.cl.cam.ac.uk/~gw104`.