Distributed Games and Strategies

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

A model for distributed computation: *Event structures*, central within models for concurrency, Petri nets, Mazurkiewicz trace languages, transition systems, ...

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

Motivation

Originally as **foundation for semantics of computation**. So as a successor to Domain Theory, the mathematical foundations of Denotational Semantics.
In the course we will see game semantics for non-deterministic dataflow, probability with nondeterminism and higher types - all bugbears of traditional domain theory.

A **structural game theory** in which one can program games and (optimal) strategies.

More distantly, there is a hope that the generality of distributed games can help bridge the **big divide** in CS between Algorithmics and Semantics. At the very least they go some way to providing a common vocabulary.

On the course

- Course notes

- Regular hand-ins and, for evaluation, a final take-away exam.

- Although you will be expected to carry out proofs and show competence in several techniques you will not be expected to reproduce, or even necessarily read, some of the hard proofs of the notes. However you will be expected to understand the theorems. The slides will delimit what is examinable.

This course is a bridge between Andy Pitts' Category Theory and Peter Sewell's Multicore Semantics and Programming in ACS. Simon Castellan will guest lecture on weak memory models via event structures.

Related Part II courses: Denotational Semantics; Topics in Concurrency.

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, weak memory models, model checking, systems biology, to proof theory;

- why such models are becoming important in **semantics** of computation and can combine the two approaches, **operational** and **denotational** semantics through the medium of games;

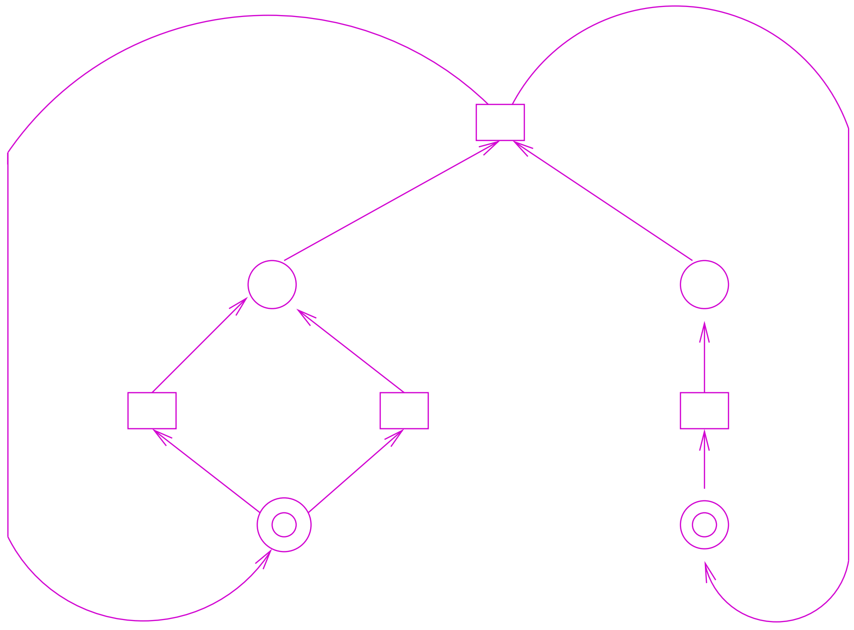
- the **underlying mathematics** of event structures and distributed games;

- the **range** of distributed games.

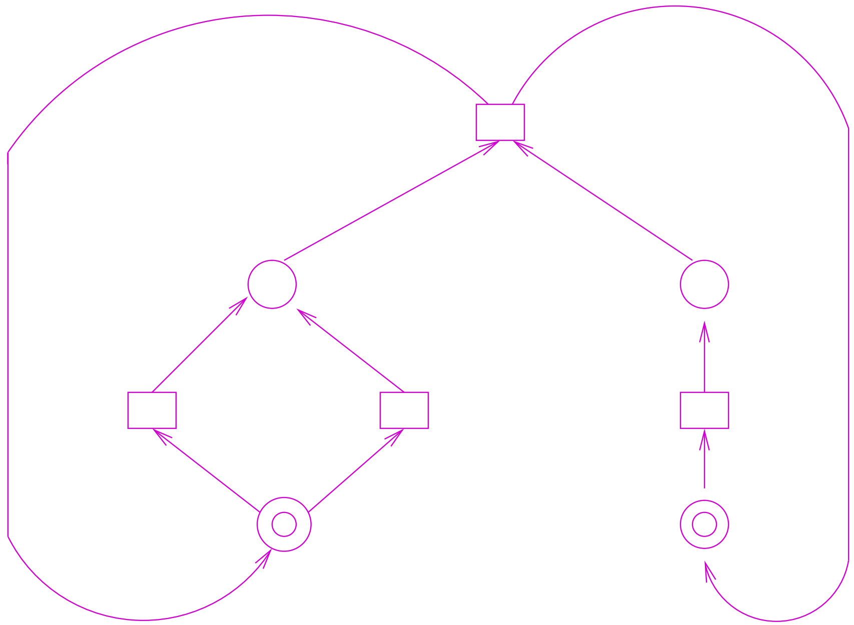
Causal/partial-order models

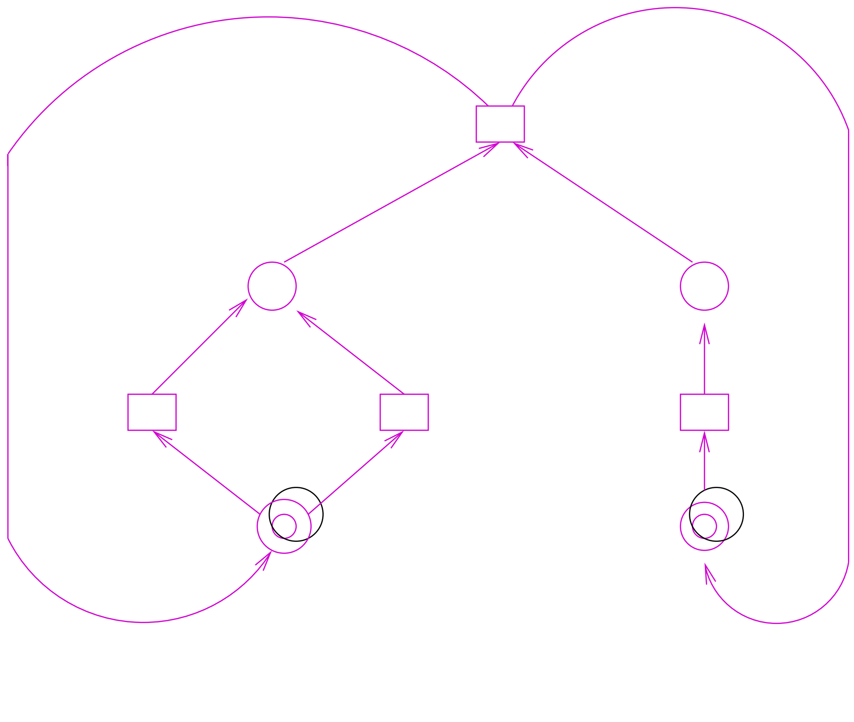
their range and applications ...

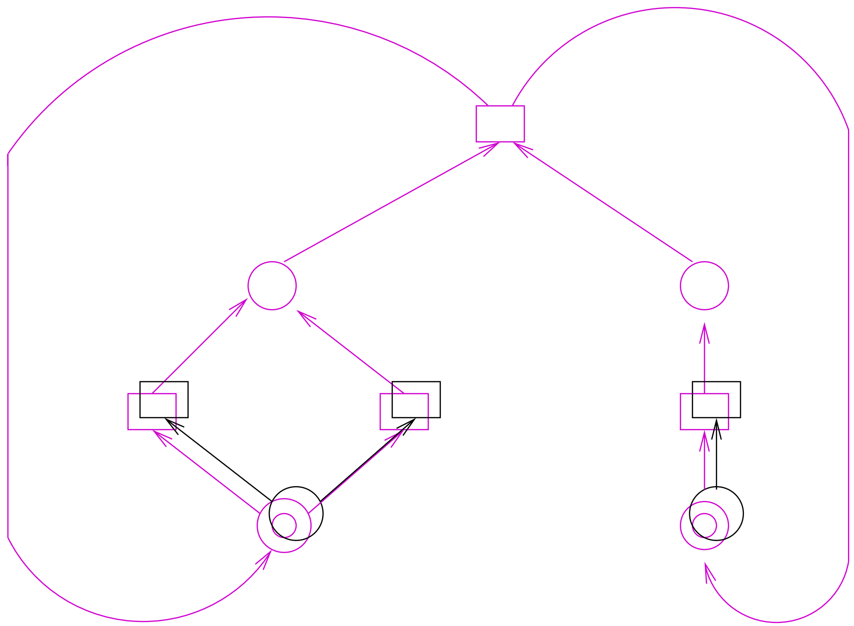
A (safe) Petri net

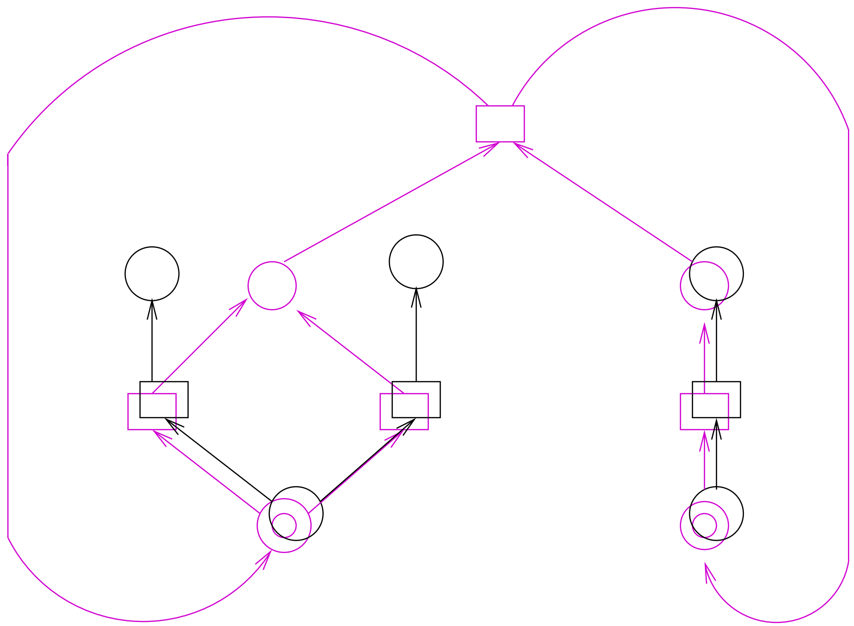


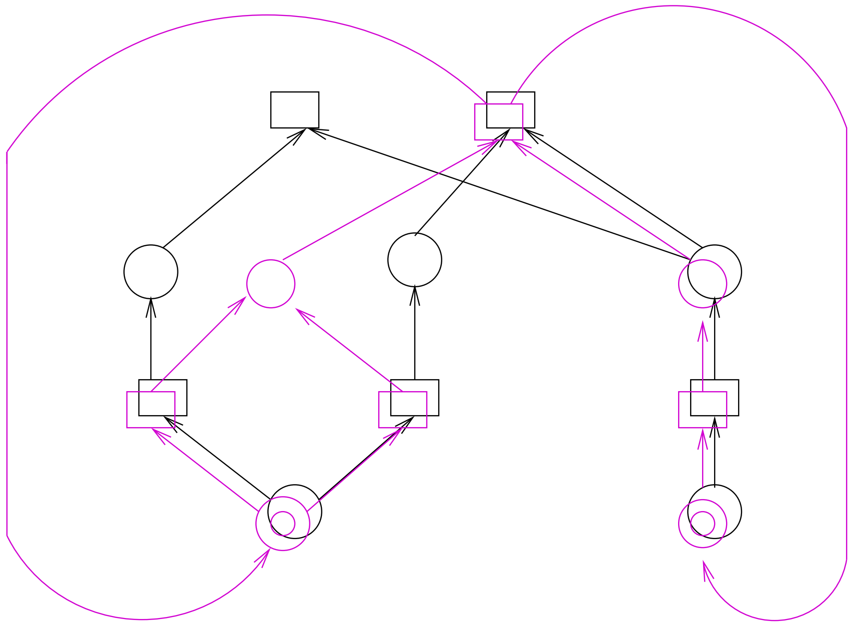
Unfolding a (safe) Petri net:

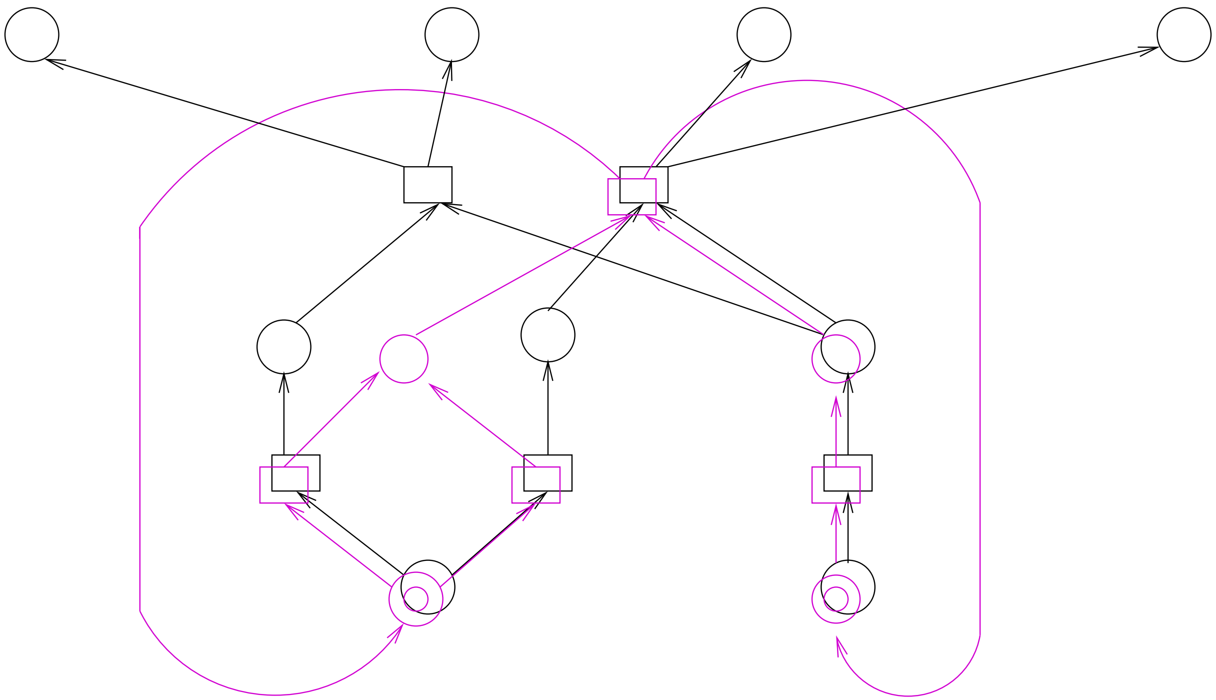


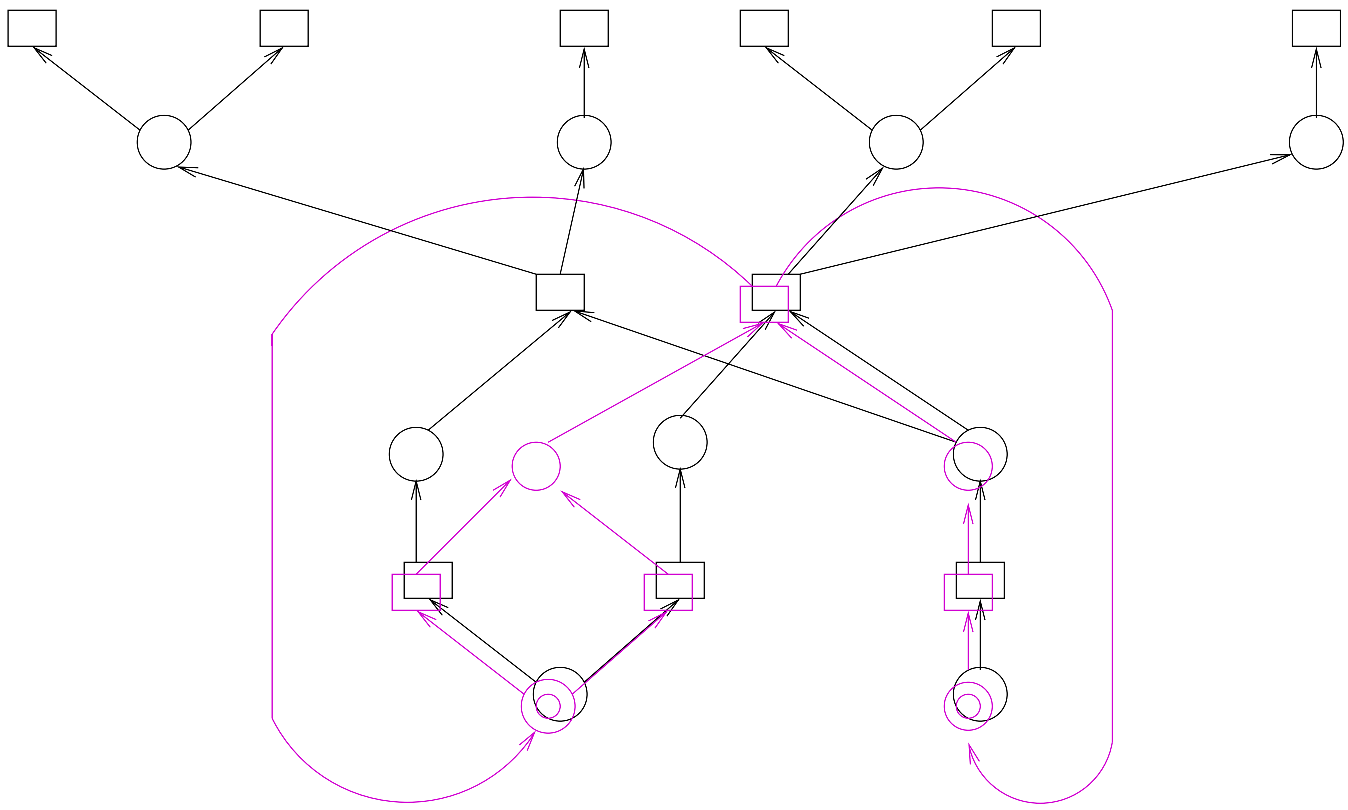


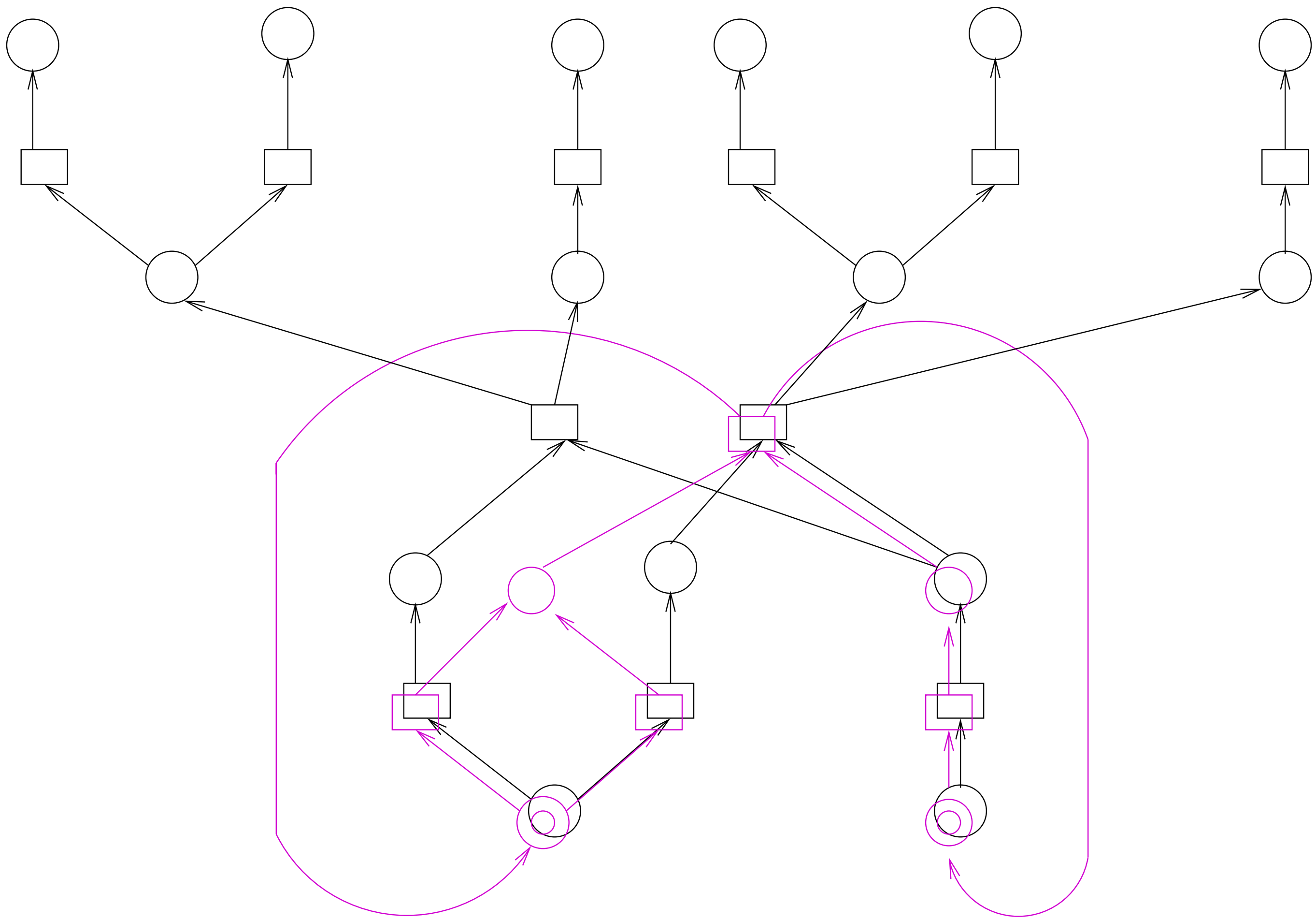


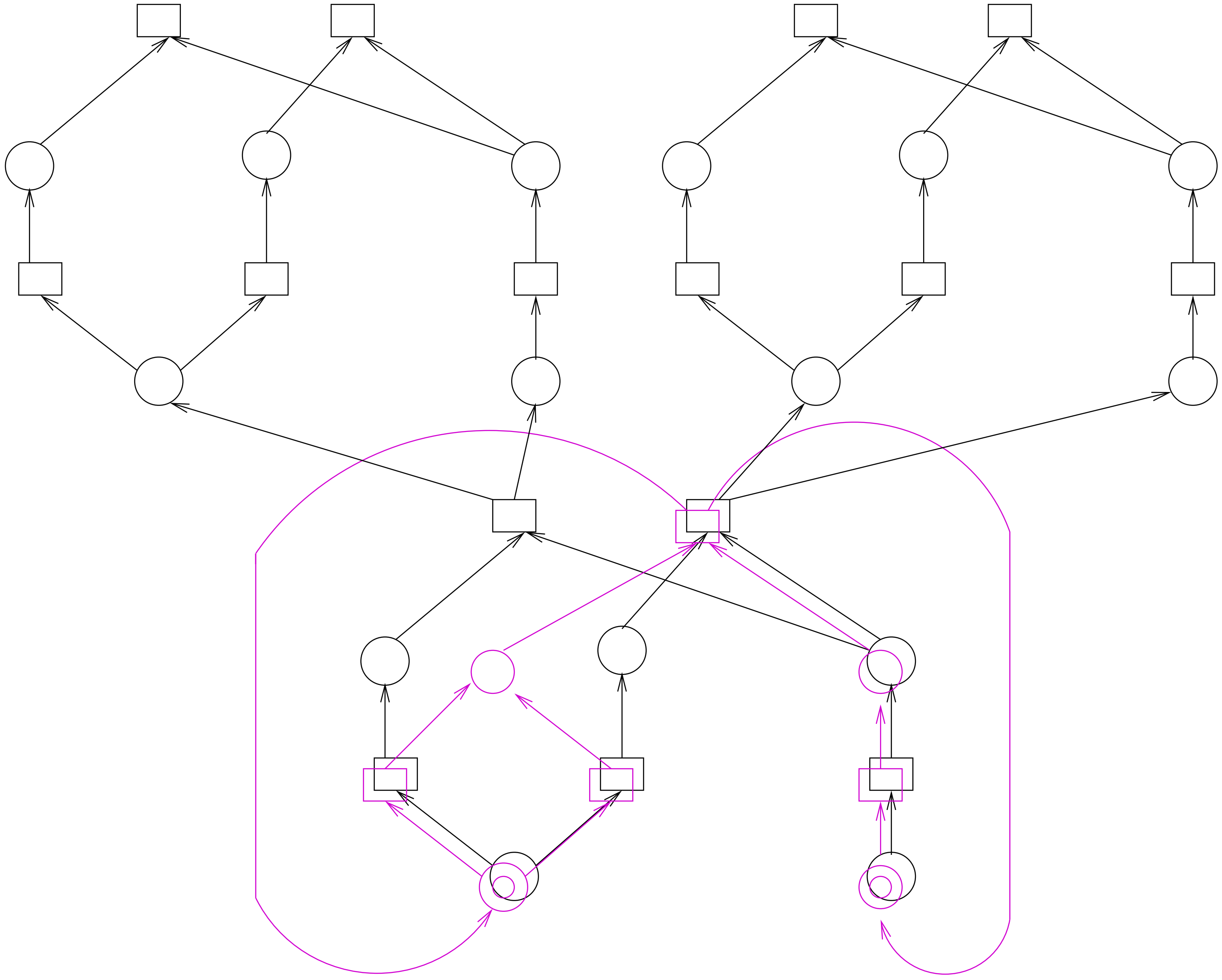


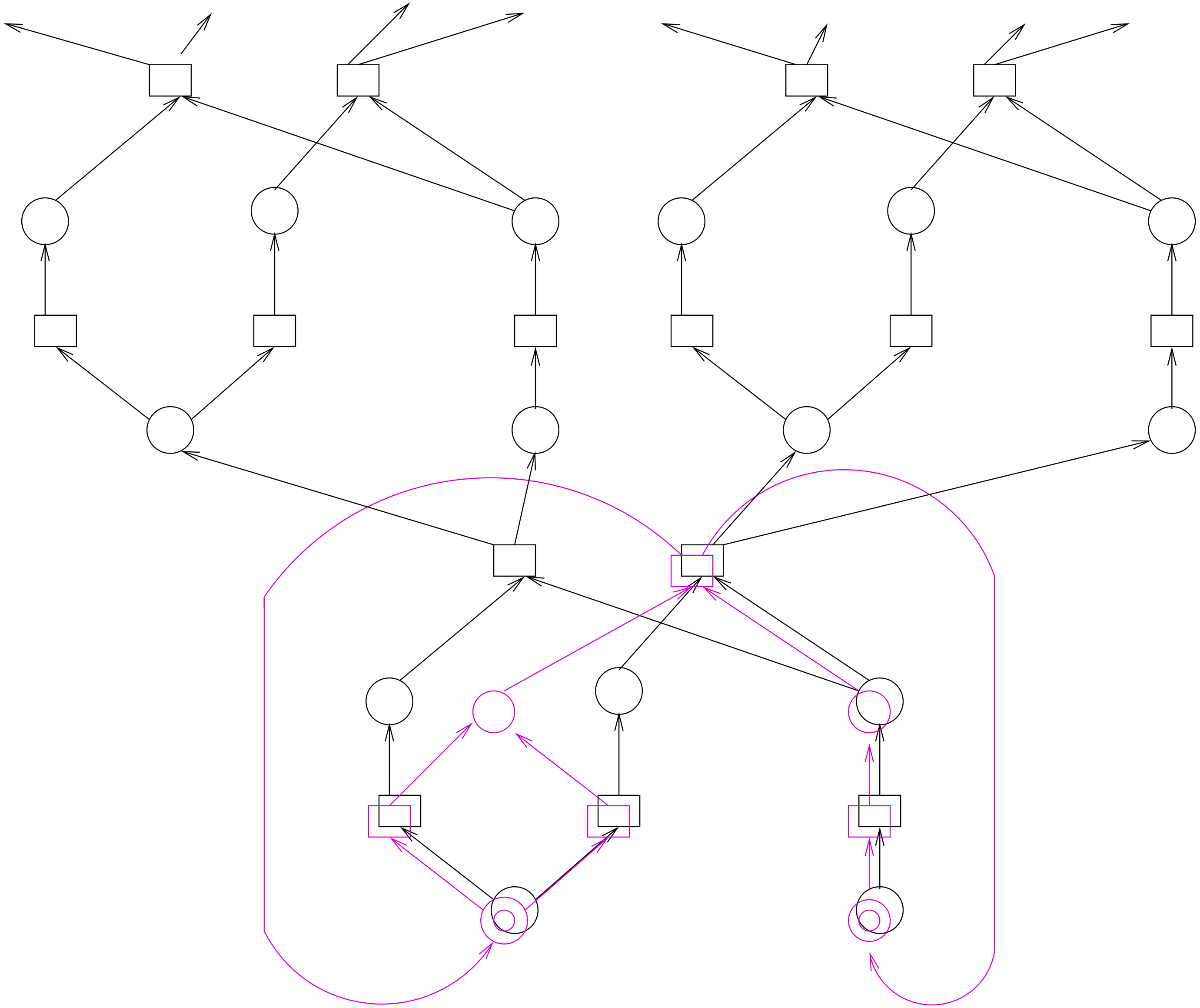


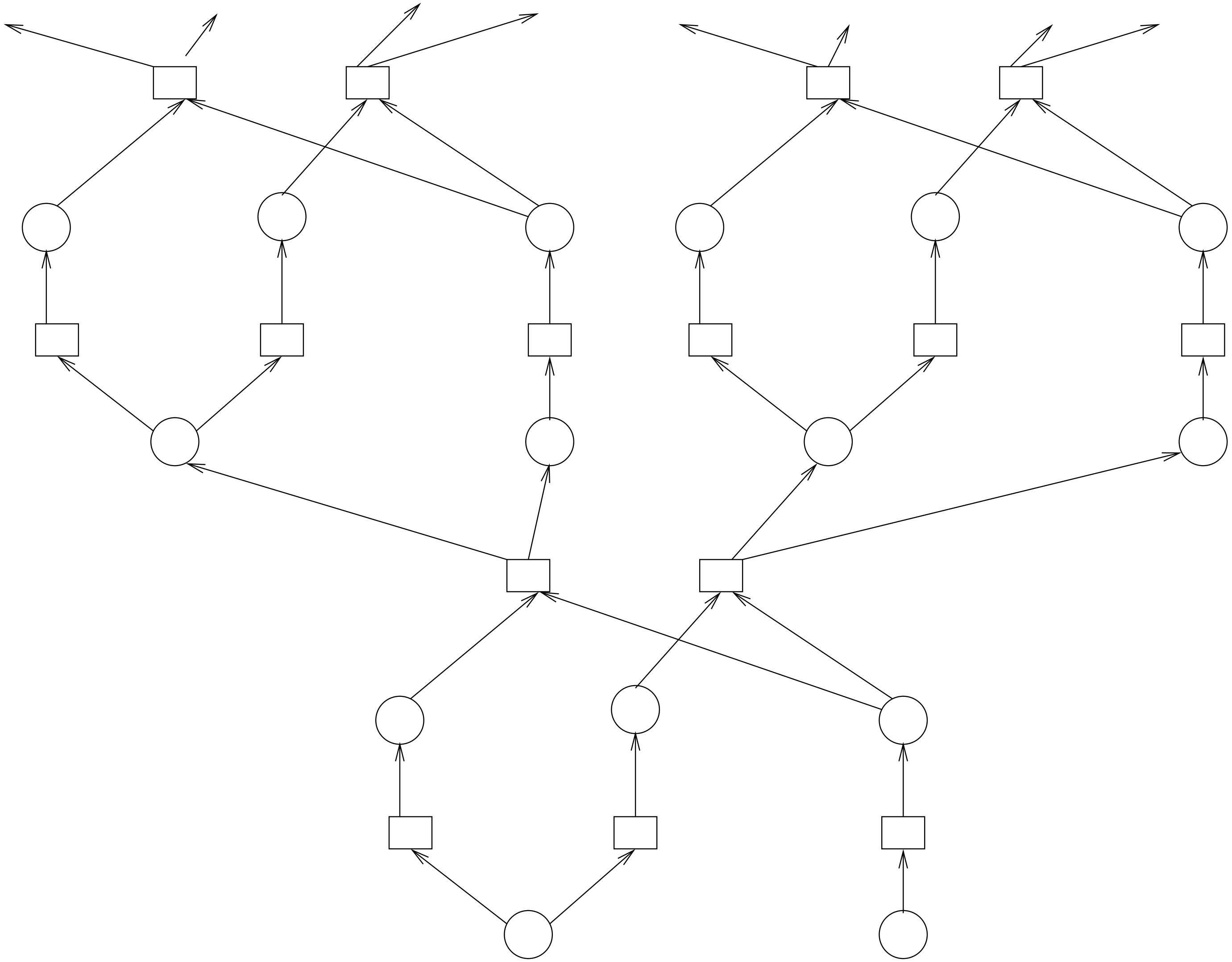


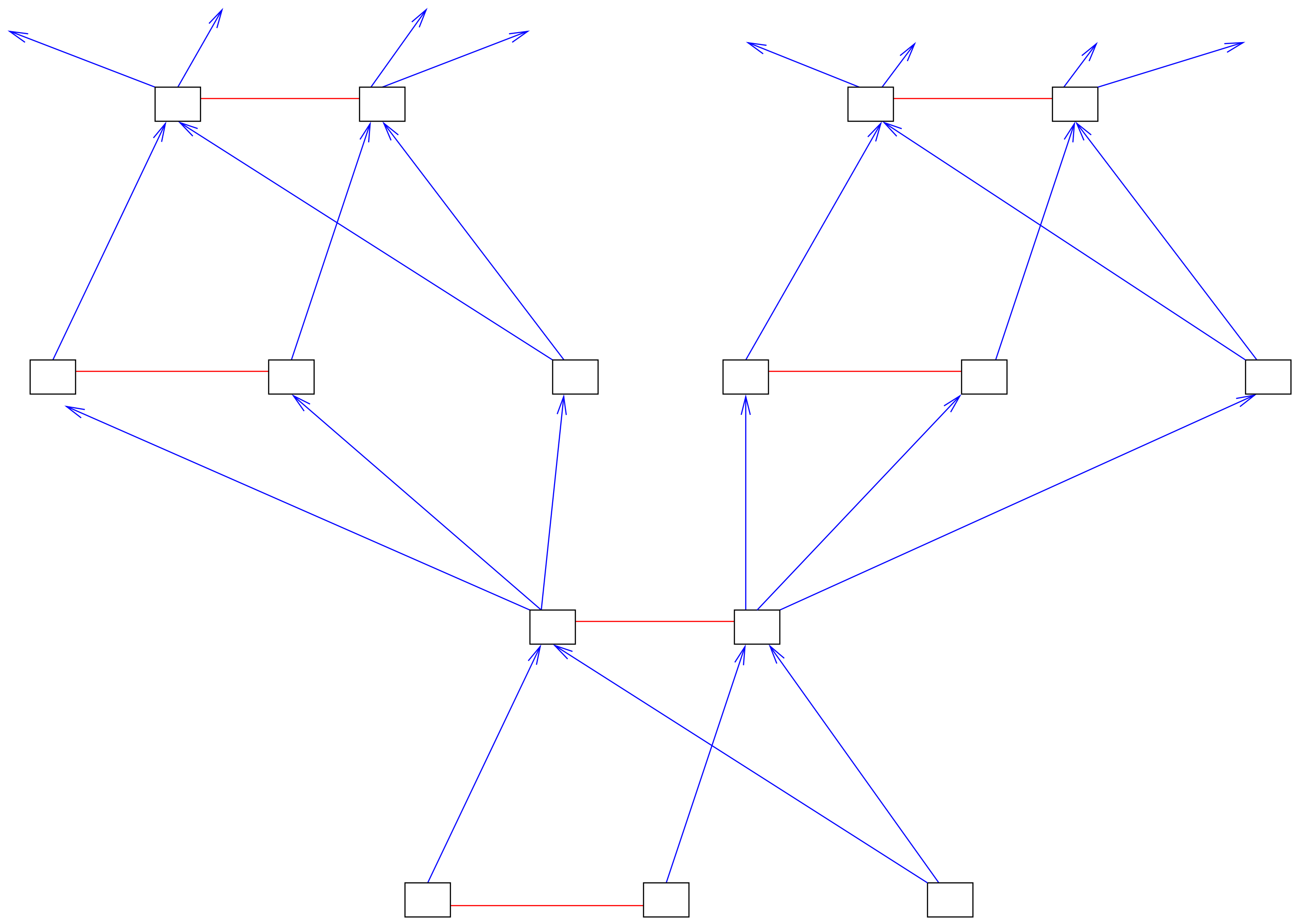












An event structure

Applications of partial-order models

Security protocols, as strand spaces, event strs [Guttman et al, Basin, Constable]; *Systems biology*, analysis of chemical pathways [Danos-Feret-Fontana-Krivine]; *Hardware*, in the design of asynchronous circuits [Yakovlev]; *Relaxed/weak memory*, event structures [Jeffrey, Pichon, Castellan]; *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 *e.g.* 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 $\bot$ and $f : D \to D$ is continuous, then f has a least fixed point $\bigsqcup_{n\in\omega} f^n(\bot)$. $\quad$ *(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.

But ...

Although denotational semantics and its mathematical foundation, domain theory, have had tremendous successes, amongst them functional programming, it suffers from certain anomalies:

- Nondeterministic dataflow;

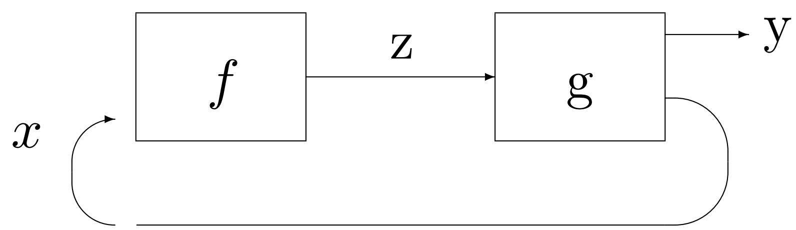
- Although it can address probabilistic computation to some extent, it has difficulties with computation which combines probability with nondeterminism or higher types;

- Concurrent/distributed computation is often captured too crudely;

- Issues of full-abstraction.

In summary, traditional domain theory has abstracted too early from operational concerns.

Deterministic dataflow—Kahn networks



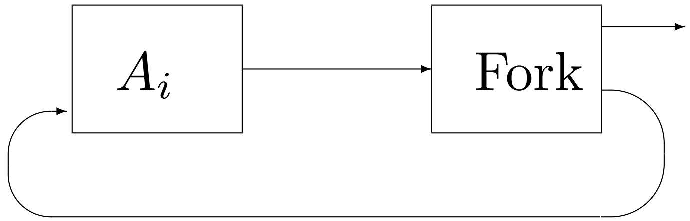
A process built from basic processes connected by channels at which they input and output.

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' will reappear as a special kinds of distributed strategies.

Traditional game semantics

Arose in the 90's as a partial answer to the quest for a more operational "domain theory."

Why games - informally and generally

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

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, $G \nrightarrow H$, is a strategy in $G^{\perp} \| H$; strategies compose with identities given by copy-cat. A strategy in H corresponds to a strategy from the empty game $\emptyset$ to H. So

$$\emptyset \nrightarrow G \nrightarrow H \text{ composes to give } \emptyset \nrightarrow H \,,$$

so a strategy in G gives rise to a strategy in H when $G \nrightarrow H$.

Game semantics of sequential programs

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:

$$\begin{tikzcd} \oplus \\ \ominus \arrow[u] \end{tikzcd}$$

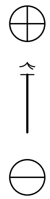
Type with a pair of values, the game:

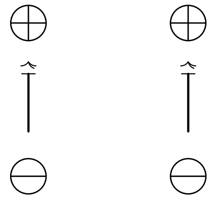
$$\begin{tikzcd} \oplus & \oplus \\ \ominus \arrow[u] & \ominus \arrow[u] \end{tikzcd}$$

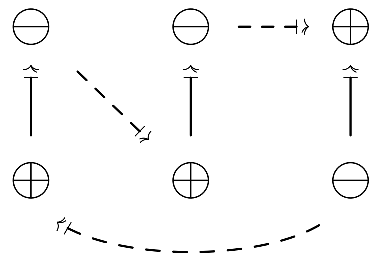
Type of ‘algorithms’ from pairs to value, the game:

$$\begin{tikzcd} \ominus & \ominus & \oplus \\ \oplus \arrow[u] & \oplus \arrow[u] & \ominus \arrow[u] \end{tikzcd}$$

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”

Game semantics of 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 gaps. *E.g.* there are conceptual problems in giving a process reading to classical proof. Partly because traditional games and strategies are not general enough. *In the course we shall see an interpretation of classical proofs as winning distributed strategies.*

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

In the course you will see:

- event structures, their techniques and constructions;
- games with winning conditions and payoff;
- games of imperfect information;
- how to put probability on event structures;
- probabilistic (and possibly quantum) strategies;
- the issue of parallel causes;
- language(s) for distributed strategies;
- applications from: weak memory models; classical proofs as strategies; ...

Ch 2. EVENT STRUCTURES

Event structures are the concurrent analogue of trees in which 'branches' are partial orders of event occurrences. Just as a transition system unfolds to a tree, so a Petri net unfolds to an occurrence net and from this to an event structure.

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 dI domains as event structures.

Event structures

An *(prime) 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: $\forall X \subseteq_{\text{fin}} x.\ X \in \mathrm{Con}$ and

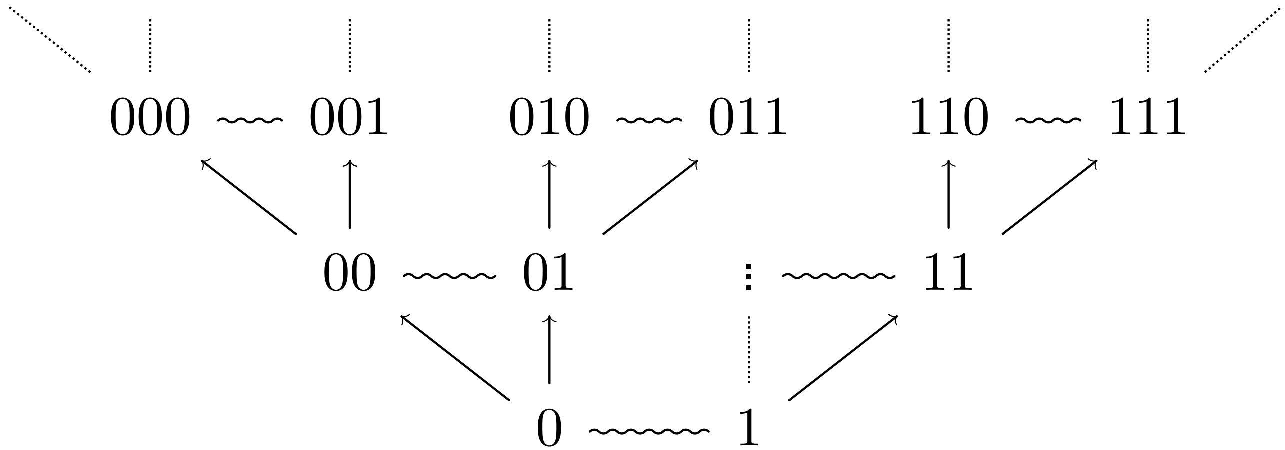
Down-closed: $\forall e, e'.\ e' \leq e \in x \Rightarrow e' \in x$.

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'.

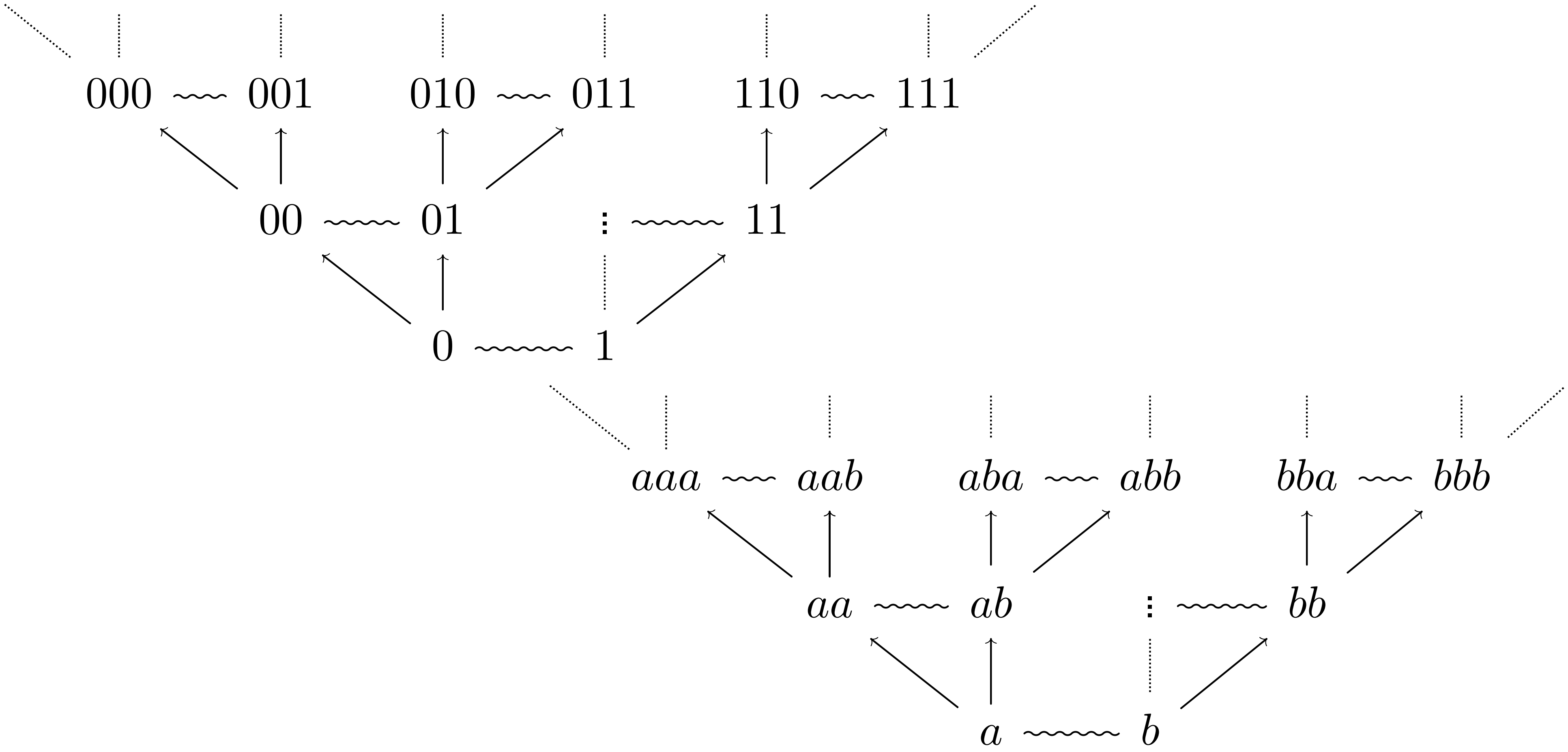
If E is countable, $(\mathcal{C}^\infty(E), \subseteq)$ is a Berry dI domain (and all such so obtained).
Finite configurations: $\mathcal{C}(E)$.

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 **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)*

The map f is **rigid** if total and preserves $\leq$.
Maps $\textit{preserve concurrency}$, and $\textit{locally reflect causal dependency i.e.}$

$$e_1, e_2 \in x \ \&\ f(e_1) \leq f(e_2) \Rightarrow e_1 \leq e_2\,.$$

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. (*Hard to construct directly so use e.g. stable families.*)

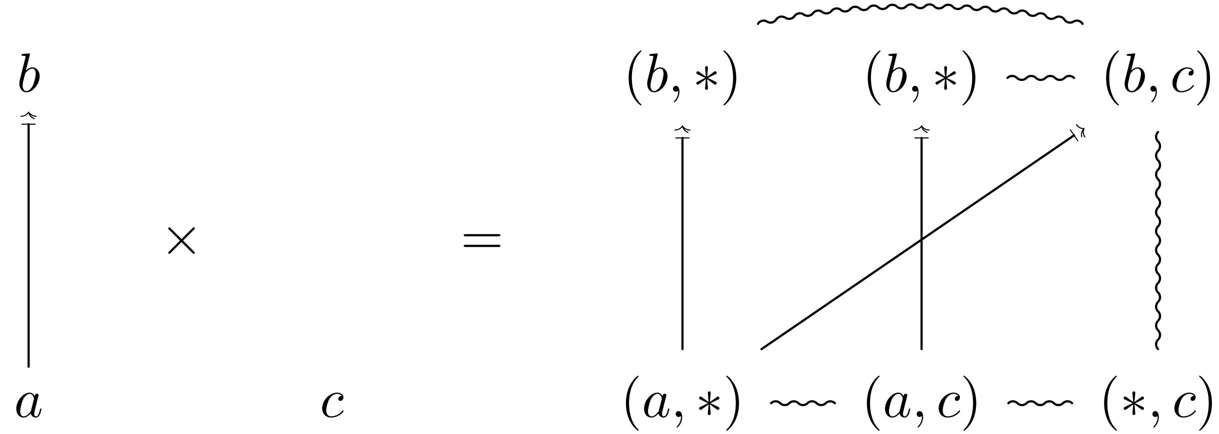
Restriction: $E \upharpoonright R$, the restriction of an event structure E to a subset of events R, has events $E' = \{e \in E \mid [e] \subseteq R\}$ with causal dependency and consistency restricted from E.

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

Pullback: Given $f : A \to C$ and $g : B \to C$ their pullback is obtained as the restriction of the product $A \times B$ to events

$$\{e \mid \text{ if } f\Pi_1(e) \ \&\ g\Pi_2(e) \text{ defined, } f\Pi_1(e) = g\Pi_2(e)\}\,.$$

Product—an example



The duplication of events with common images under the projections, as in the two events carrying $(b, *)$ *can be troublesome!*

Recursively-defined event structures

An approximation order $\trianglelefteq$ on event structures:

$$
\begin{aligned}
(E', \leq', \mathrm{Con}') \trianglelefteq (E, \leq, \mathrm{Con}) \iff & E' \subseteq E \ \& \\
& \forall e' \in E'.\ [e']' = [e'] \ \& \\
& \forall X' \subseteq E'.\ X' \in \mathrm{Con}' \iff X \in \mathrm{Con}\,.
\end{aligned}
$$

The order $\trianglelefteq$ forms a 'large cpo,' with bottom the empty event structure, and lubs of an ω-chains given by unions.

Constructions on event structures can be ensured to be continuous w.r.t. $\trianglelefteq$;
it suffices to check that they are $\trianglelefteq$-monotonic and continuous on event sets,
$i.e.\ A \trianglelefteq B \Rightarrow Op(A) \trianglelefteq Op(B)$ and
$a \in Op(\bigcup_{i\in\omega} A_i) \Rightarrow a \in \bigcup_{i\in\omega} Op(A_i)$ on ω-chains.
$\leadsto$ *recursive definition via least fixed points.*

Hiding - via a factorization system

A partial map

$$f : E \to E'$$

of event structures has **partial-total factorization** as a composition

$$E \xrightarrow{p} E{\downarrow}V \xrightarrow{t} E'$$

where $V =_{\mathrm{def}} \{e \in E \mid f(e) \text{ is defined}\}$ is the domain of definition of f;

the **projection** $E{\downarrow}V =_{\mathrm{def}} (V, \leq_V, \mathrm{Con}_V)$, where

$v \leq_V v'$ iff $v \leq v'$ & $v, v' \in V$ and $X \in \mathrm{Con}_V$ iff $X \in \mathrm{Con}$ & $X \subseteq V$;

the *partial* map $p : E \to E{\downarrow}V$ acts as identity on V and is undefined otherwise;

and the *total* map $t : E{\downarrow}V \to E'$, called the **defined part** of f, acts as f.

Ch 3. STABLE FAMILIES

A technique for working with event structures. They generalise the configurations of an event structure to allow the same event to occur is several incompatible ways. Nevertheless they determine event structures.

Stable families

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 $\mathcal{F}$.

Stable families - alternative characterisation

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

Completeness: $\emptyset \in \mathcal{F} \;\&\; \forall x, y \in \mathcal{F}.\ x \uparrow y \Rightarrow x \cup y \in \mathcal{F}$;

Stability: $\forall x, y \in \mathcal{F}.\ x \uparrow y \Rightarrow x \cap y \in \mathcal{F}$;

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

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

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:

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

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

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

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 an event e to the prime configuration $[e] =_{\mathrm{def}} \{e' \in E \mid e' \leq e\}$ — it is an isomorphism.

The counit $top : \mathcal{C}(\mathrm{Pr}(\mathcal{F})) \to \mathcal{F}$ takes $[e]_x$ to e;
it induces an order-isomorphism between $(\mathcal{C}(\mathrm{Pr}(\mathcal{F})), \subseteq)$ and $(\mathcal{F}, \subseteq)$
given by $y \mapsto top\, y = \bigcup y$. Details on the next slide.

$top : \mathcal{C}(\Pr(\mathcal{F})) \to \mathcal{F}$ **with** $[e]_x \mapsto e$ **induces an order iso:**
$\theta(y) = top\ y = \bigcup y$ with mutual inverse $\phi(x) = \{[e]_x \ \mid \ e \in x\}$.
Clearly, both θ and ϕ preserve $\subseteq$.

$\theta\phi(x) = \bigcup\{[e]_x \ \mid \ e \in x\} = x$.

$\phi\theta(y) = \{[e]_{\bigcup y} \ \mid \ e \in \bigcup y\}$. To show $rhs = y$ use

$[e]_x \subseteq z \iff [e]_x = [e]_z$, whenever $e \in x$ and z in $\mathcal{F}$:
From $e \in [e]_x \subseteq z$ we get $[e]_z \subseteq [e]_x$. Hence $e \in [e]_z \subseteq x$ ensuring the converse inclusion $[e]_x \subseteq [e]_z$, so $[e]_x = [e]_z$.

"$y \subseteq rhs$": $[e]_x \in y \Rightarrow [e]_x \subseteq \bigcup y \Rightarrow [e]_x = [e]_{\bigcup y} \in rhs$.

"$rhs \subseteq y$: Assume $p \in rhs$. Then $p = [e]_{\bigcup y}$ with $e \in \bigcup y$. We have $e \in [e']_x \in y$ for some e', x with $e' \in x$. So $[e]_x \subseteq [e']_x \in y$ ensuring $[e]_x \in y$. As $[e]_x \subseteq \bigcup y$ we obtain $p = [e]_{\bigcup y} = [e]_x$, so $p \in y$.

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

$$
\begin{aligned}
& x \in \mathcal{A} \times \mathcal{B} \text{ iff} \\
& x \text{ is a finite subset of } A \times_* B \text{ s.t. } \ \pi_1 x \in \mathcal{A} \;\&\; \pi_2 x \in \mathcal{B}, \\
& \forall e, e' \in x.\ \pi_1(e) = \pi_1(e') \text{ or } \pi_2(e) = \pi_2(e') \Rightarrow e = e'\,, \& \\
& \forall e, e' \in x.\ e \neq e' \Rightarrow \exists y \subseteq x.\ \pi_1 y \in \mathcal{A} \;\&\; \pi_2 y \in \mathcal{B} \;\& \\
& \qquad\qquad\qquad\qquad\qquad\qquad (e \in y \iff e' \notin y)\,.
\end{aligned}
$$

Product of event structures

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

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

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

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

Pullbacks of stable families with total maps

Let $f : \mathcal{A} \to \mathcal{C}$ and $g : \mathcal{B} \to \mathcal{C}$ be total maps of stable families. Assume $\mathcal{A}$ and $\mathcal{B}$ have underlying sets A and B. Define $D =_{\text{def}} \{(a, b) \in A \times B \mid f(a) = g(b)\}$ with projections π_1 and π_2 to the left and right components. Define a family of configurations of the *pullback* to consist of

$$
\begin{aligned}
&x \in \mathcal{D} \text{ iff} \\
&x \text{ is a finite subset of } D \text{ such that } \pi_1 x \in \mathcal{A} \ \&\ \pi_2 x \in \mathcal{B}, \\
&\forall e, e' \in x.\ e \neq e' \Rightarrow \exists y \subseteq x.\ \pi_1 y \in \mathcal{A} \ \&\ \pi_2 y \in \mathcal{B}\ \& \\
&\qquad\qquad\qquad\qquad\qquad\qquad (e \in y \iff e' \notin y)\,.
\end{aligned}
$$

(Local injectivity of π_1, π_2 follows automatically.)

Pullbacks of stable families with total maps - a characterisation

Proposition Finite configurations of $\mathcal{D}$ correspond to the composite bijections

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

between configurations $x \in \mathcal{A}$ and $y \in \mathcal{B}$ s.t. $fx = gy$ for which the transitive relation generated on θ by

$$(a, b) \leq_\theta (a', b') \text{ if } a \leq_x a' \text{ or } b \leq_y b'$$

is a partial order.

Consequently finite configurations of the pullback of event structures correspond to "secure bijections" as above.

Other adjunctions between models for concurrency

Many models for concurrency naturally form categories, related by adjunctions:

- The 'inclusion' of Event Structures in Stable Families has a right adjoint, Pr;

- The inclusion of (the category of) Trees in Event Structures has a right adjoint, serialising an event structure to a tree;

- The 'inclusion' functor from Trees to Transition Systems has a right adjoint, that of unfolding a transition sytem to a tree;

- The inclusion of Occurrence Nets in (1-Safe) Petri Nets has a right adjoint, unfolding a net to its occurrence net;

- The forgetful functor from Occurrence Nets to Event Structures has a left adjoint.

......

Ch 4. DISTRIBUTED GAMES

In which games and strategies are represented by event structures.

Structural maps of event structures - recap

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

$$fx \in \mathcal{C}(E') \text{ and } e_1, e_2 \in x \ \&\ f(e_1) = f(e_2) \Rightarrow e_1 = e_2\,.$$

Note that when f is total it restricts to a bijection $x \cong fx$, for any $x \in \mathcal{C}(E)$.
A total map is **rigid** when it preserves causal dependency.

Maps *preserve concurrency*, and *locally reflect causal dependency:*

$$e_1, e_2 \in x \ \&\ f(e_1) \leq f(e_2) \text{ (both defined) } \Rightarrow e_1 \leq e_2\,.$$

Pullbacks of total maps of event structures *(For composition)*

Total maps $f : A \to C$ and $g : B \to C$ have pullbacks in the category of event structures:

$$\begin{tikzcd} & P \arrow[dl, "\pi_1"'] \arrow[dr, "\pi_2"] & \\ A \arrow[dr, "f"'] & & B \arrow[dl, "g"] \\ & C\,. & \end{tikzcd}$$

Finite configurations of P correspond to the composite bijections

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

between configurations $x \in \mathcal{C}(A)$ and $y \in \mathcal{C}(B)$ s.t. $fx = gy$ for which the transitive relation generated on θ by $\quad (a, b) \leq (a', b')$ if $a \leq_A a'$ or $b \leq_B b'$
is a partial order.

Defined part of a map *(For hiding)*

A partial map

$$f : E \to E'$$

of event structures has **partial-total factorization** as a composition

$$E \xrightarrow{p} E{\downarrow}V \xrightarrow{t} E'$$

where $V =_{\mathrm{def}} \{e \in E \mid f(e) \text{ is defined}\}$ is the domain of definition of f;

the **projection** $E{\downarrow}V =_{\mathrm{def}} (V, \leq_V, \mathrm{Con}_V)$, where

$v \leq_V v'$ iff $v \leq v'$ & $v, v' \in V$ and $X \in \mathrm{Con}_V$ iff $X \in \mathrm{Con}$ & $X \subseteq V$;

the *partial* map $p : E \to E{\downarrow}V$ acts as identity on V and is undefined otherwise;

and the *total* map $t : E{\downarrow}V \to E'$, called the **defined part** of f, acts as f.

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**, or **pre-strategy**, in a game A is a total map

$$\begin{array}{c} S \\ \Big\downarrow \sigma \\ A \end{array}$$

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]

NB: A strategy in a game A is a strategy for Player;
a strategy for Opponent - a counter-strategy - is a strategy in $A^\perp$*.*

When are two nd plays/strategies the same?

A map between nd plays:

$$
\begin{tikzcd}
S \arrow[r, "f"] \arrow[dr, "\sigma"'] & S'. \arrow[d, "\sigma'"] \\
& A
\end{tikzcd}
$$

which commutes.

When f is an isomorphism we regard the two nd plays/strategies as essentially the same.

Example of a strategy: copy-cat strategy from A to A

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

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

Copy-cat in general

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 is consistent in $A^\perp \| A$. The map γ_A is the identity on the common underlying set of events. Then,

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

Composition of strategies $\sigma : S \to A^{\perp} \| B$, $\tau : T \to B^{\perp} \| C$

Via pullback. Ignoring polarities, the composite partial map

$$\begin{tikzcd}
 & T * S \arrow[dl, "\Pi_1"'] \arrow[dr, "\Pi_2"] \arrow[dd, phantom, "\vee" very near start] & \\
S \| C \arrow[dr, "\sigma \| C"'] & & A \| T \arrow[dl, "A \| \tau"] \\
 & A \| B \| C \arrow[d] & \\
 & A \| C &
\end{tikzcd}$$

has defined part, yielding $T \odot S \xrightarrow{\tau \odot \sigma} A^{\perp} \| C$ once reinstate polarities.

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

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

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

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

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

$-$-*Innocent:* 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 $\sigma : A \nrightarrow B$.

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) 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.*

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

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.*

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

Strategies—alternative description 2

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

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

we obtain a factorization system $((\mathcal{C}(A), \sqsubseteq_A), \supseteq^-, \subseteq^+)$, *i.e.* $\exists! z.$
$$\begin{tikzcd} & & x \\ y \arrow[urr, dotted, "\sqsubseteq"] \arrow[r, phantom, "\supseteq^-"] & \; & z \arrow[u, phantom, "\subseteq^+"] \end{tikzcd}\ .$$

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

$$\sigma“ : (\mathcal{C}(S), \sqsubseteq_S) \to (\mathcal{C}(A), \sqsubseteq_A)\,, \quad i.e.\;\; \exists! x'. \quad \begin{tikzcd} x' \arrow[r, dotted, phantom, "\sqsubseteq_S"] \arrow[d, dotted, "\sigma“"'] & x \arrow[d, "\sigma“"] \\ y \arrow[r, phantom, "\sqsubseteq_A"] & \sigma“(x) \end{tikzcd}\,,$$

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

$\rightsquigarrow$ *A lax functor from strategies to profunctors ...*

Strategies—informal alternative description 3

Given a strategy $\sigma : S \to A$ it can be shown (Lemma 8.23) that

$$\{x^+ \cup \sigma x^- \ \mid\ x \in \mathcal{C}(S)\}$$

is a stable family order-isomorphic to $(\mathcal{C}(S), \subseteq)$ under $x \mapsto x^+ \cup \sigma x^-$.

This implies a strategy $\sigma : S \to A$ is got from the game A by adding

- conflicting copies of $+$-events with
- "causal wiring" required of the game and respecting receptivity and innocence.

A bicategory of games

Objects are event structures with polarity—the games, A, B, ... ;
Arrows $\sigma : A \nrightarrow B$ are strategies $\sigma : S \to A^\perp \| B$;

2-Cells $\begin{tikzcd} A \arrow[r, bend left, "\sigma"] \arrow[r, bend right, "\sigma'"'] \arrow[r, phantom, "\Downarrow f"] & B \end{tikzcd}$ are maps $f : S \to S'$ such that $\begin{tikzcd} S \arrow[r, "f"] \arrow[dr, "\sigma"'] \arrow[r, phantom, shift right=3, "="] & S' \arrow[d, "\sigma'"] \\ & A^\perp \| B \end{tikzcd}$.

The vertical composition of 2-cells is the usual composition of maps. Horizontal composition is given by $\odot$ (which extends to a functor via the universality of pb and partial-total factorisation).
Duality: $\sigma : A \nrightarrow B$ *corresponds to* $\sigma^\perp : B^\perp \nrightarrow A^\perp$, *as* $A^\perp \| B \cong (B^\perp)^\perp \| A^\perp$.
The bicategory of strategies is compact-closed (so has a trace, a feedback operation extending that of nondeterministic dataflow)—though with extra features of winning conditions or pay-off, this will weaken to $*$*-autonomy.*

Ch 5. DETERMINISTIC STRATEGIES

Deterministic strategies

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

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

where $Neg[X] =_{\text{def}} \{s' \in S \ \mid \ \exists s \in X.\ pol_S(s') = - \ \&\ s' \leq s\}$.
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 $\exists y.\ x \stackrel{e}{\longrightarrow\!\!\!\!\subset} y$.

Nondeterministic copy-cats

Take A to consist of two events, one $+$ve and one $-$ve event, inconsistent with each other $\oplus \sim\!\sim\!\sim \ominus$. The construction $\mathbb{C}\!\!\mathbb{C}_A$:

$$
\begin{tikzcd}
A^{\perp} & \ominus \arrow[r, "" , rightarrow] \arrow[d, no head, squiggly] & \oplus \arrow[d, no head, squiggly] & A \\
 & \oplus & \ominus \arrow[l] &
\end{tikzcd}
$$

To see $\mathbb{C}\!\!\mathbb{C}_A$ is not deterministic, take x to be the singleton set consisting $e.g.$ of the $-$ve event on the left and s, s' to be the $+$ve and $-$ve events on the right.

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

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

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 (so, $\sigma : S \rightarrowtail A$).

$\rightsquigarrow$ 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;

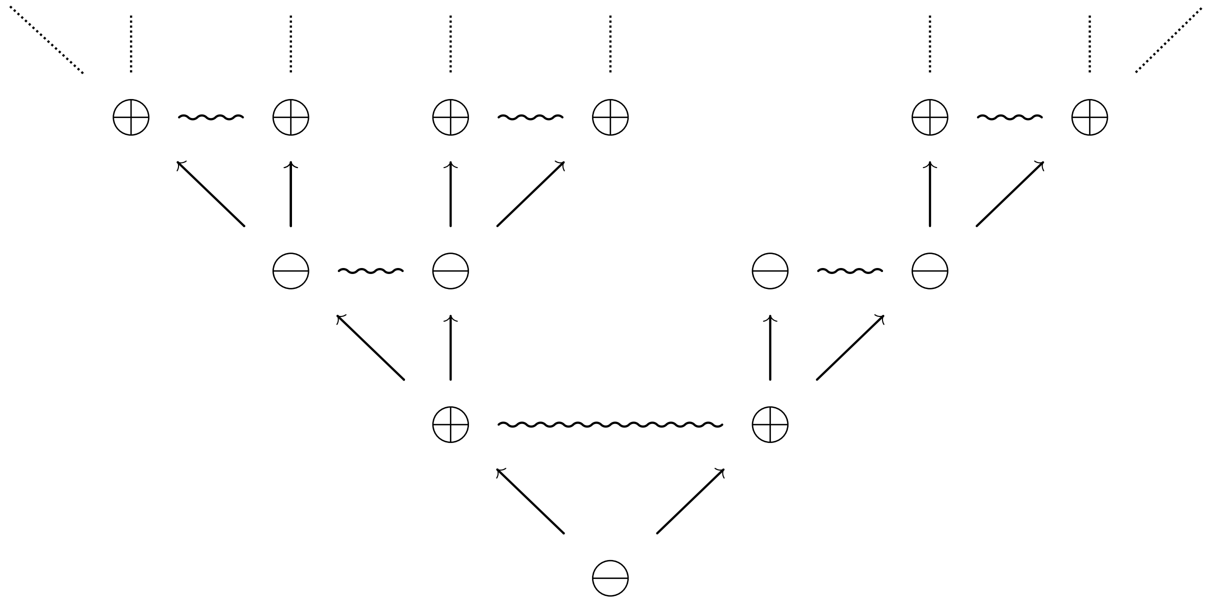
determinacy: If $x \stackrel{a}{\longrightarrow\!\!\!\subset}$ and $x \stackrel{a'}{\longrightarrow\!\!\!\subset}$ in F with $pol_A(a) = +$, then $x \cup \{a, a'\} \in F$;

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 $x_1 \cap x_2 \in F$.

Example: a tree-like game



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

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

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: a strategy $\sigma : S \to A^{\perp} \| B$ corresponds to

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

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 8. 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 $\sigma : \emptyset \nrightarrow A$ and $\tau : A \nrightarrow \emptyset$ ...

Composition of strategies without hiding

Let $\sigma : S \to A^\perp \| B$ and $\tau : T \to B^\perp \| C$ be strategies. Their composition before hiding:

$$\begin{tikzcd}
 & T * S \arrow[dl, "\Pi_1"'] \arrow[dr, "\Pi_2"] \arrow[dd, phantom, "\vee"] & \\
S \| C \arrow[dr, "\sigma \| C"'] & & A \| T \arrow[dl, "A \| \tau"] \\
 & A \| B \| C &
\end{tikzcd}$$

Special case

Let σ be a strategy in B and τ a counterstrategy, a strategy in $B^\perp$. Their composition before hiding:

$$\begin{tikzcd} & T * S \arrow[dl, "\Pi_1"'] \arrow[dr, "\Pi_2"] & \\ S \arrow[dr, "\sigma"'] & & T \arrow[dl, "\tau"] \\ & B & \end{tikzcd}$$

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

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 strategies. Then,

$$z \in \mathcal{C}^{\infty}(T * S) \text{ is } +\text{-maximal iff}$$

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

[Also holds for receptive pre-strategies.]

Ex.1. $\ominus \rightsquigarrow \oplus$ has a winning strategy only if $\{\ominus\} \in W$.

Ex.2.
$$\begin{tikzcd} \ominus \arrow[r, squiggly] & \oplus \\ & \oplus \arrow[ul, rightarrowtail] \arrow[u, rightarrowtail] \end{tikzcd}$$
the empty strategy is winning if $\emptyset \in W$.

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

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

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 $G = (A, W_G)$, $H = (B, W_H)$,

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

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 $G \otimes H =_{\text{def}} (G^{\perp} \| H^{\perp})^{\perp}$ 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,

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

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

Function space With $G \multimap H =_{\text{def}} G^{\perp} \| H$ 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 $\mathbb{C}\!\mathbb{C}_A$:

$$
\begin{tikzcd}
A^\perp & \ominus \arrow[r] \arrow[d, no head, squiggly] & \oplus \arrow[d, no head, squiggly] & A \\
& \oplus & \ominus \arrow[l] &
\end{tikzcd}
$$

With $W = \{\{\oplus\}\}$. Taking $+$-maximal $x = \{\ominus, \ominus\}$, $\overline{x}_1 \in W$ 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.

Ch 8, 9,10. SOME APPLICATIONS AND EXTENSIONS

In the notes ...

Total strategies: To pick out a subcategory of *total* strategies (where Player can always answer Opponent) within simple games.
Determinacy: 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 necessary and sufficient condition on a game for it to be determined w.r.t. Borel winning conditions is that it is race-free and bounded concurrent (in no configuration is an event concurrent with infinitely many events of opposing polarity).
A game semantics for Predicate Calculus: 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. Via games with imperfect information, semantics of Hintikka's IF logic.
Strategies as concurrent processes (not in notes): their 'may-and-must' behaviour via "stopping configurations" (to refine $+$-maximal configurations) gives an accurate analysis of 'must win' and 'may win.'

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:

$$\llbracket R(x_1, \cdots, x_k) \rrbracket_M \rho = \begin{cases} (\emptyset, \{\emptyset\}) & \text{if } \rho \models_M R(x_1, \cdots, x_k)\,, \\ (\emptyset, \emptyset) & \text{otherwise.} \end{cases}$$

$$\llbracket \phi \wedge \psi \rrbracket_M \rho = \llbracket \phi \rrbracket_M \rho \otimes \llbracket \psi \rrbracket_M \rho$$

$$\llbracket \phi \vee \psi \rrbracket_M \rho = \llbracket \phi \rrbracket_M \rho \| \llbracket \psi \rrbracket_M \rho$$

$$\llbracket \neg \phi \rrbracket_M \rho = (\llbracket \phi \rrbracket_M \rho)^\perp$$

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

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

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 $\bigodot_{v \in V} G_v$ 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.

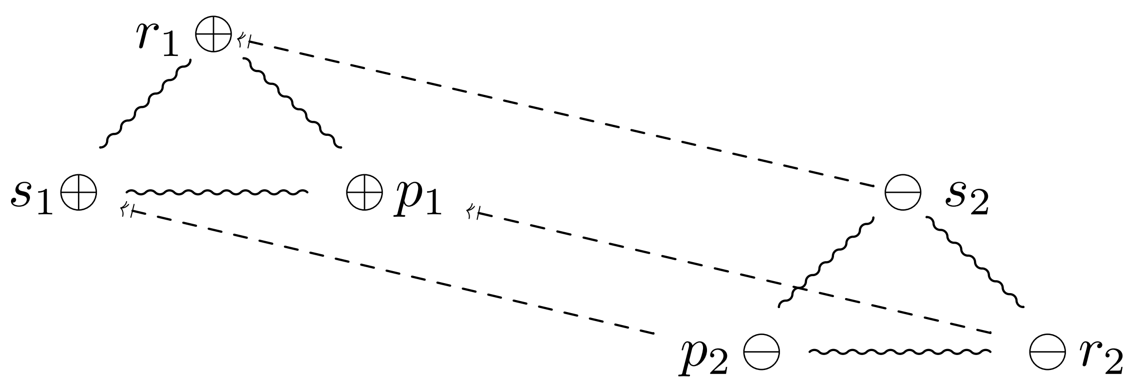
Games with imperfect information

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

PROBABILISTIC EVENT STRUCTURES & STRATEGIES

Ch 11. From strategies to probabilistic strategies

$$\begin{array}{c} S \\ \big\downarrow \sigma \\ A \end{array}$$

Aim

(1) To endow S with probability, while

(2) taking account of the fact that in a strategy Player can't be aware of the probabilities assigned by Opponent. (*E.g.* in 'Matching pennies')

Causal independence between Player and Opponent moves will entail their probabilistic independence. Equivalently, probabilistic dependence of Player on Opponent moves will presuppose their causal dependence.

Probabilistic event structures

A **probabilistic event structure** comprises an event structure $E = (E, \leq, \mathrm{Con})$ together with a (normalized) **continuous valuation**, $i.e.$ a function w from the Scott open subsets of configurations $\mathcal{C}^\infty(E)$ to $[0,1]$ which is

(normalized) $w(\mathcal{C}^\infty(E)) = 1$ **(strict)** $w(\emptyset) = 0$

(monotone) $U \subseteq V \Rightarrow w(U) \leq w(V)$

(modular) $w(U \cup V) + w(U \cap V) = w(U) + w(V)$

(continuous) $w(\bigcup_{i\in I} U_i) = \mathrm{sup}_{i\in I} w(U_i)$ for *directed* unions $\bigcup_{i\in I} U_i$.

Intuition: $w(U)$ *is the probability of the result being in* U.
A cts valuation extends to a probability measure on Borel sets of configurations.

A workable characterization: A **probabilistic event structure** comprises an event str. E with a **configuration-valuation** $v : \mathcal{C}(E) \to [0,1]$ which satisfies

(normalized) $v(\emptyset) = 1$ and

(non $-$ve drop) $d_v^{(n)}[y; x_1, \cdots, x_n] \geq 0$, for all $n \in \omega$, and $y \subseteq x_1, \cdots, x_n$ in $\mathcal{C}(E)$.

For $y \subseteq x_1, \cdots, x_n$ in $\mathcal{C}(E)$,

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

—the index I ranges over $\emptyset \neq I \subseteq \{1, \cdots, n\}$ s.t. $\{x_i \mid i \in I\}$ is compatible.
(*Sufficient to check the 'drop condition' for* $y \longrightarrow\!\!\!\!\subset x_1, \cdots, x_n$)

Theorem. Continuous valuations restrict to configuration-valuations.
A configuration-valuation extends to a unique continuous valuation on open sets, and that to a unique probabilistic measure on Borel subsets of configurations.
(*The result holds in greater generality, for Scott domains*)

Example Two concurrent events a and b, with configuration-valn and probability:

$$\begin{tikzcd}[column sep=small]
 & & & 1/4 & & & \\
 & & & 1/4 & & & \\
 & & & \{a,b\} & & & \\
1/4 & 1/2 & \{a\} \arrow[ur, hook] & & \{b\} \arrow[ul, hook] & 1/4 & 0 \\
 & & & \emptyset \arrow[ul, hook] \arrow[ur, hook] & & & \\
 & & & 1 & & & \\
 & & & 1/2 & & &
\end{tikzcd}$$

Probabilistic event structure with polarities

Let E be an event structure in which (not necessarily all) events carry $+/-$.
Write $x \subseteq^p y$ if $x \subseteq y$ and no event in $y \setminus x$ has polarity $-$.

Now, a **configuration-valuation** is a function $v : \mathcal{C}(E) \to [0,1]$ for which

$$v(\emptyset) = 1, \qquad\qquad x \subseteq^- y \Rightarrow v(x) = v(y) \text{ , for all } x, y \in \mathcal{C}(E),$$

and the "drop condition"

$$d_v^{(n)}[y; x_1, \cdots, x_n] \geq 0$$

for all $n \in \omega$ and $y \subseteq^p x_1, \cdots, x_n$ in $\mathcal{C}(E)$.
(*Sufficient to check the 'drop condition' for* $y {-\!\!\!\subset}^p x_1, \cdots, x_n$)

A **probabilistic event structure with polarity** comprises E an event structure with polarity together with a configuration-valuation $v_E : \mathcal{C}(E) \to [0,1]$.

Probabilistic strategies

Assume games are *race-free*, $i.e.$ there is no immediate conflict between events of opposite polarity.

A **probabilistic strategy** in A comprises S, v_S, a probabilistic event structure with polarity, and a strategy $\sigma : S \to A$.

A race-free game A has a **probabilistic copy-cat** by taking $v_{\mathbb{C}_A}$ constantly 1 —this is a configuration-valuation as $\mathbb{C}_A$ is deterministic for race-free A.

For the **composition** $\tau \odot \sigma$ endow the pb $T * S$ with configuration-valuation $v(x) = v_S(\Pi_1^S x) \times v_T(\Pi_2^T x)$. This forms a configuration-valuation because assuming $\Pi_1^S y \longrightarrow\!\!\!\!\subset^+ \Pi_1^S x_i$ for $1 \le i \le m$ and $\Pi_2^T y \longrightarrow\!\!\!\!\subset^+ \Pi_2^T x_i$ for $m+1 \le i \le n$,

$$d_v^{(n)}[y; x_1, \cdots, x_n] = d_v^{(m)}[\Pi_1^S y; \Pi_1^S x_1, \cdots, \Pi_1^S x_m] \times d_v^{(n-m)}[\Pi_2^T y; \Pi_2^T x_{m+1}, \cdots, \Pi_2^T x_n]\,.$$

$\leadsto$ *a bicategory of probabilistic strategies on race-free games—2-cells?*

A special case of composition without hiding: play-off

Given a probabilistic strategy $v_S, \sigma : S \to A$ and counter-strategy $v_T, \tau : T \to A^\perp$ we obtain

$$\begin{tikzcd} & T * S \arrow[dl, "\Pi_1"'] \arrow[dr, "\Pi_2"] \arrow[dd, phantom, "\vee" near start] & \\ S \arrow[dr, "\sigma"'] & & T \arrow[dl, "\tau"] \\ & A & \end{tikzcd}$$

with valuation $v_S \Pi_1 \times v_T \Pi_2$ on the pullback $T * S$ — a probabilistic event structure, making A a probabilistic event str. too, with probabilty measure μ.

Adding **pay-off** as a random variable X from $\mathcal{C}^\infty(A)$ get **expected pay-off** as the Lebesgue integral

$$\int X(x)\ d\mu(x)\,.$$

Maps between probabilistic strategies (2-cells?)

The push-forward of a configuration-valuation across a map:

Given a map of strategies
$$\begin{tikzcd} S \arrow[r, "f"] \arrow[dr, "\sigma"'] & S' \arrow[d, "\sigma'"] \\ & A \end{tikzcd}$$
and a configuration-valuation v of S,

cannot in general push it forwards to a configuration-valuation fv of S'.

However, **if f is rigid**, defining

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

for $y \in \mathcal{C}(S')$, yields a configuration-valuation fv of S' —the **push-forward** of v.

The rigid image of a probabilistic strategy

A strategy $\sigma : S \to A$ has a **rigid image** comprising
$$\begin{tikzcd} S \arrow[r, two heads, "f_0"] \arrow[dr, "\sigma"'] & S_0 \arrow[d, "\sigma_0"] \\ & A \end{tikzcd}$$
where

f_0 is rigid epi and σ_0 is a strategy with universal property:
$$\begin{tikzcd} S \arrow[r, two heads, "f"] \arrow[dr, "\sigma"'] \arrow[rr, bend left, "f_0"] & S' \arrow[r, dashed] \arrow[d, dashed, "\sigma'"] & S_0 \arrow[dl, "\sigma_0"] \\ & A & \end{tikzcd}$$

A probabilistic strategy $\sigma : S \to A$ with configuration-valuation v of S has rigid image the probabilistic strategy $\sigma_0 : S_0 \to A$ with configuration-valuation the push-forward $f_0 v$.

Remark: Rigid images are not preserved by composition of strategies.

A bicategory of games and probabilistic strategies

Objects are race-free games A, B, C, ... ;

Arrows $\sigma : A \not\rightarrow B$ are probabilistic strategies $\sigma : S \to A^{\perp} \| B$ with configuration valuation $v : \mathcal{C}(S) \to [0,1]$;

2-Cells $\begin{tikzcd} A \arrow[r, bend left=40, "{\sigma,v}", ""{name=U, below}] \arrow[r, bend right=40, "{\sigma',v'}"', ""{name=D}] & B \arrow[Rightarrow, from=U, to=D, "f"] \end{tikzcd}$ are rigid maps $f : S \to S'$ making $\begin{tikzcd} S \arrow[r, "f"] \arrow[dr, "\sigma"'] & S' \arrow[d, "\sigma'"] \\ & A^{\perp} \| B \end{tikzcd}$

commute and $fv \le v'$.

2-cells include rigid embeddings preserving the value assigned by configuration valuations and the approximation order $\trianglelefteq$ on event structures. Taking rigid images (they're 2-cells) yields a functor to an order-enriched category.

Constructions on (probabilistic) strategies

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 $f : A \to B$ reflects $+$-consistency (subsumes prefixing $\ominus.\sigma$ and $\oplus.\sigma$).

Conjunction $\sigma_1 \wedge \sigma_2$, if $\sigma_1 : A$ and $\sigma_2 : A$, conjoined via their pullback.

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

Probabilistic sum $\sum_{i \in I} p_i \sigma_i : A$, if $\sigma_i : A$ for all $i \in I$, where I is countable with sub-probability distribution $\langle p_i \rangle_{i \in I}$.

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

Recursion on types and processes. **[Is a profunctor-based metalanguage]**

Extensions in the notes

• **Payoff and value theorems:** endow games with a measurable payoff function on configurations. Optimal strategies, Nash equilibria and a value theorem.

• **Imperfect information:** where games also carry a preorder of access levels to restrict the causal dependencies of strategies. (Optimal strategies and value theorems?)

• **Simple Quantum games:** Interpret moves of a game as projection and unitary operators on a Hilbert space s.t. concurrent moves are associated with commuting operators. The play-off of a probabilistic strategy against a probabilistic counterstrategy results in a probabilistic quantum experiment, where, assuming the game has an initial quantum state, each particular experiment determines a probability distribution over end positions of the game.

Recent advances

- **Intensional full-abstraction results.**

- **Probabilistic strategies with parallel causes.**

- **A compositional proof of Herbrand's theorem via an interpretation of classical proofs as concurrent strategies.**

- **Truly Quantum strategies.**

Thanks to the ERC, Aurore Alcolei, Nathan Bowler, Simon Castellan, Pierre Clairambault, Julian Gutierrez, Jonathan Hayman, Martin Hyland, Alex Katovsky, Tamas Kispeter, Marc Lasson, Hugo Paquet, Silvain Rideau, Frank Roumen, Marc de Visme.