Models of concurrency, categories, and games

Pierre Clairambault and Glynn Winskel

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

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

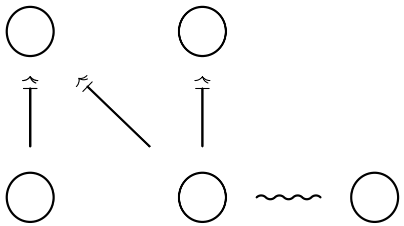
$$\{e\} \in \mathrm{Con} \text{ for all } e \in E,$$

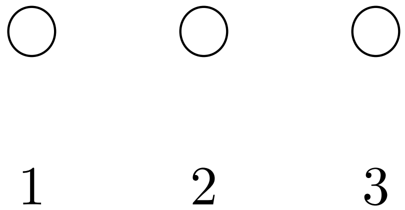
$$Y \subseteq X \in \mathrm{Con} \Rightarrow Y \in \mathrm{Con}, \quad \text{and}$$

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

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

Event structures - two simple examples





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

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

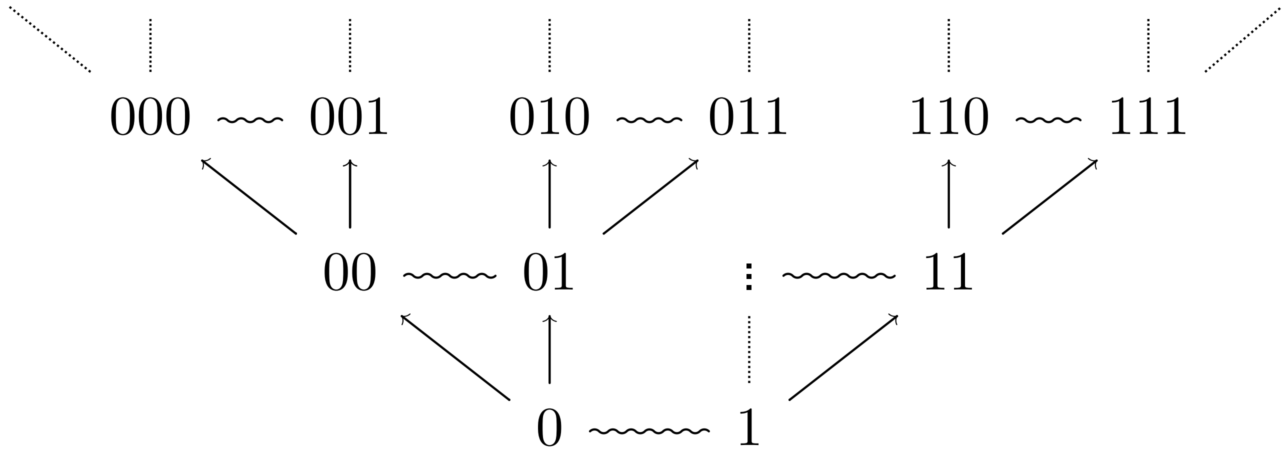
Down-closed: $\forall e, e'.\ e' \leq e \in x \Rightarrow e' \in x$.

For an event e the set $[e] =_{\mathrm{def}} \{e' \in E \mid e' \leq e\}$ 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'.

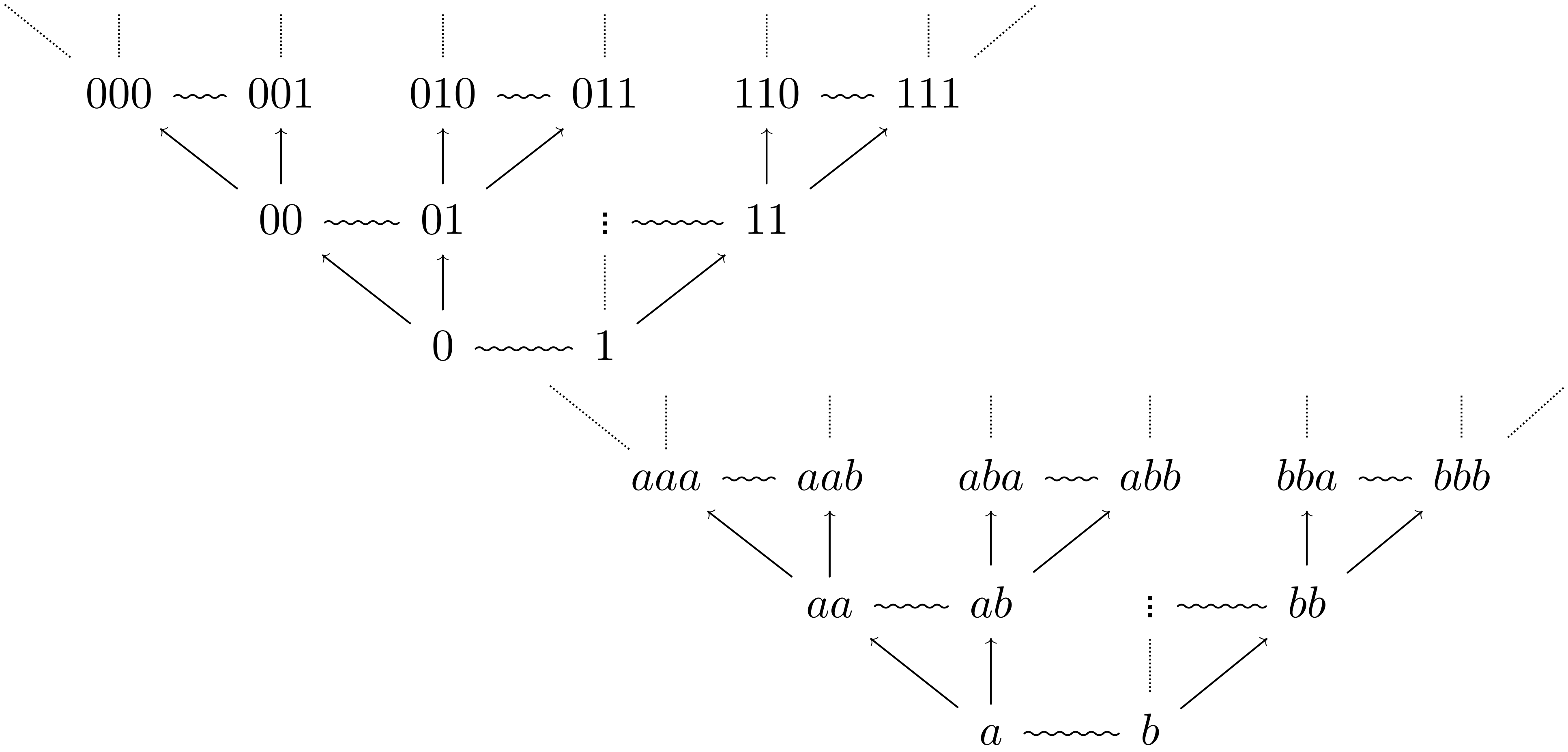
If E is countable, $(\mathcal{C}^\infty(E), \subseteq)$ is a Berry dI domain (and all such so obtained). Finite configurations: $\mathcal{C}(E)$.

Example: Streams as event structures



$\rightsquigarrow$ conflict (inconsistency) $\qquad \longrightarrow$ 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 $\mathit{preserve\ concurrency}$, and $\mathit{locally\ reflect\ causal\ dependency\ i.e.}$

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

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. An *equaliser* ...

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

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

Product—an example

$$
\begin{tikzcd}
b \\
a \arrow[u]
\end{tikzcd}
\quad \times \quad
c
\quad = \quad
\begin{tikzcd}
(b,*) \arrow[rr, bend left, no head, squiggly] & (b,*) \arrow[r, no head, squiggly] & (b,c) \\
(a,*) \arrow[u] \arrow[urr] \arrow[r, no head, squiggly] & (a,c) \arrow[u] \arrow[r, no head, squiggly] & (*,c) \arrow[u, no head, squiggly]
\end{tikzcd}
$$

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.

A factorisation system ...

The factorisation is characterised to within isomorphism by the following universal characterisation: for any factorisation

$$f : E \xrightarrow{g_0} E_1 \xrightarrow{g_1} E'$$

where g_0 is partial and g_1 is total there is a (necessarily total) unique map $h : E{\downarrow}V \to E_1$ such that

$$\begin{tikzcd} E \arrow[r, "f_0"] \arrow[dr, "g_0"'] & E{\downarrow}V \arrow[r, "f_1"] \arrow[d, dashed, "h"] & E' \\ & E_1 \arrow[ur, "g_1"'] & \end{tikzcd}$$

commutes.

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:

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

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

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

Lemma $(*)$ $\quad [e]_x \subseteq z \iff [e]_x = [e]_z$, *whenever* $e \in x$ *and* z *in* $\mathcal{F}$.
Proof. 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 $[e]_x = [e]_z$. '$\Leftarrow$' Trivial. $\square$

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 $x \in \mathcal{F}$,

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

Two significant maps:

The map of event structures $E \to \mathrm{Pr}(\mathcal{C}(E))$ takes an event e to the prime configuration $[e] =_{\mathrm{def}} \{e' \in E \ \mid \ e' \leq e\}$ — it is an isomorphism.

The map of stable families $top : \mathcal{C}(\mathrm{Pr}(\mathcal{F})) \to \mathcal{F}$ 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.

$top : \mathcal{C}(\mathrm{Pr}(\mathcal{F})) \to \mathcal{F}$ **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

Lemma $(*)$ $\quad [e]_x \subseteq z \iff [e]_x = [e]_z$, whenever $e \in x$ and z in $\mathcal{F}$.

"$y \subseteq rhs$": $[e]_x \in y \Rightarrow [e]_x \subseteq \bigcup y \Rightarrow [e]_x = [e]_{\bigcup y} \in rhs$, by $(*)$.

"$rhs \subseteq y$: 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 $[e]_x \in y$. Therefore $[e]_x \subseteq \bigcup y$ so by $(*)$ we obtain $p = [e]_{\bigcup y} = [e]_x$, giving $p \in y$.

Adjunctions via free objects

One way to present an adjunction between two categories $\mathcal{A}$ and $\mathcal{B}$ is by

- a functor $G : \mathcal{B} \to \mathcal{A}$

- an operation F from objects of $\mathcal{A}$ to objects of $\mathcal{B}$

- for each $A \in \mathcal{A}$, a *unit* map $\eta_A : A \to GF(A)$ satisfying for any $B \in \mathcal{B}$ and any map $f : A \to G(B)$ there is a unique map $g : F(A) \to B$ s.t. $f = G(g) \circ \eta_A$.

$$\begin{tikzcd} \mathcal{A} \arrow[rr, dashed, bend right=10, "F"'] & & \mathcal{B} \arrow[ll, bend right=10, "G"'] \end{tikzcd}$$

$$\begin{tikzcd} A \arrow[r, "\eta_A"] \arrow[dr, "f"'] & GF(A) \arrow[d, dotted, "G(g)"] \\ & G(B) \end{tikzcd} \qquad \begin{tikzcd} F(A) \arrow[d, dotted, "g"] \\ B \end{tikzcd}$$

Adjunctions via cofree objects

A dual way to present an adjunction between two categories $\mathcal{A}$ and $\mathcal{B}$ is by

• a functor $F : \mathcal{A} \to \mathcal{B}$

• an operation G from objects of $\mathcal{B}$ to objects of $\mathcal{A}$

• for each $B \in \mathcal{B}$, a *counit* map $\epsilon_B : FG(B) \to B$ satisfying for any $A \in \mathcal{A}$ and any map $g : F(A) \to B$ there is a unique map $f : A \to G(B)$ s.t. $g = \epsilon_B \circ F(f)$.

$$\begin{tikzcd} \mathcal{A} \arrow[r, bend right, "F"'] & \mathcal{B} \arrow[l, dashed, bend right, "G"'] \end{tikzcd}$$

$$\begin{tikzcd} G(B) & & B & FG(B) \arrow[l, "\epsilon_B"'] \\ A \arrow[u, dotted, "f"] & & & F(A) \arrow[ul, "g"] \arrow[u, dotted, "F(f)"'] \end{tikzcd}$$

Adjunctions via natural bijections

Another way to present an adjunction between two categories $\mathcal{A}$ and $\mathcal{B}$ is by a pair of functors $F : \mathcal{A} \to \mathcal{B}$ and $G : \mathcal{B} \to \mathcal{A}$ for which there is a bijection

$$\theta_{A,B} : \mathcal{B}(F(A), B) \cong \mathcal{A}(A, G(B))\,,$$

natural in $A \in \mathcal{A}, B \in \mathcal{B}$.

The unit, $\eta_A = \theta_{A,F(A)}(\mathrm{id}_{F(A)})$.

The counit, $\epsilon_B = \theta^{-1}_{G(B),B}(\mathrm{id}_{G(B)})$.

F is called the *left adjoint* and G the *right adjoint* of the adjunction.
Often the adjunction is denoted by $F \dashv G$.

An adjunction from event structures to stable families

$$\mathcal{E} \underset{\mathcal{C}(\_)}{\overset{\Pr}{\rightleftarrows}} \mathcal{F}am$$

$\Pr$ is right adjoint to the functor, taking an event structure E to the stable family $\mathcal{C}(E)$.

The unit of the adjunction $E \to \Pr(\mathcal{C}(E))$ takes an event e to the prime configuration $[e] =_{\mathrm{def}} \{e' \in E \mid e' \leq e\}$ — it is an isomorphism.

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

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
$A \times_* B =_{\mathrm{def}} \{(a, *) \mid a \in A\} \cup \{(a, b) \mid a \in A \ \& \ b \in B\} \cup \{(*, b) \mid b \in B\}$, the product of sets with partial functions, with (partial) projections π_1 and π_2—treating $*$ as 'undefined'—with configurations

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

Product of event structures

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

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

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

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 $D =_{\rm def} \{(a,b) \in A \times B \mid f(a) = g(b)\}$ with projections π_1 and π_2 to the left and right components. Define a family of configurations of the *pullback* to consist of

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

(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.

...... (Adjunctions compose.)

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:

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

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 $(a, b) \le (a', b')$ if $a \le_A a'$ or $b \le_B b'$ 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 **pre-strategy**, or *nondeterministic play*, in a game A is a total map

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

preserving polarity; S is the event structure with polarity describing the moves played according to the pre-strategy.

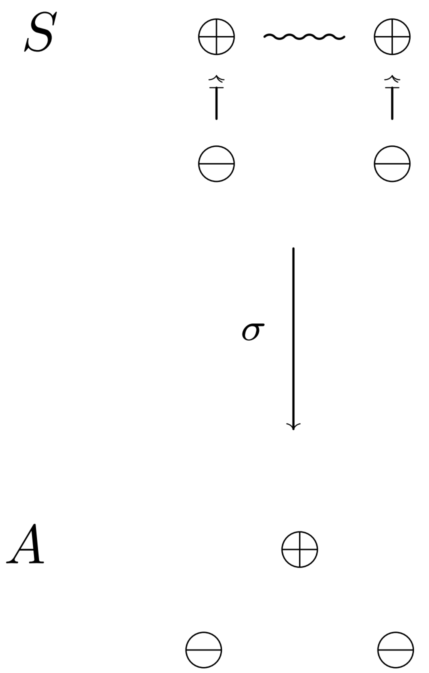
A **strategy in** a game A is a (**special**) pre-strategy $\sigma : S \to A$.

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

[Conway, Joyal]

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

A strategy - an example



configurations of S = *"states of play"*

configurations of A = *"positions of the game"*

The strategy: answer either move of Opponent by the Player move.

When are two pre-strategies the same?

A map between pre-strategies:

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

which commutes.

When f is an isomorphism we regard the two pre-strategies as essentially the same.

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

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

$$\begin{tikzcd}
A^{\perp} & & & & A \\
\overline{a}_2 & \ominus \arrow[rr, dashed, |->] & & \oplus & a_2 \\
\overline{a}_1 & \oplus \arrow[u, |->] & & \ominus \arrow[ll, dashed, |->] \arrow[u, |->] & a_1
\end{tikzcd}$$

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

Via pullback. Ignoring polarities, the composite partial map

$$
\begin{tikzcd}
 & T \circledast S \arrow[dl, "\Pi_1"'] \arrow[dr, "\Pi_2"] \arrow[dd, phantom, "\vee", pos=0.1] & \\
S \| C \arrow[dr, "\sigma \| C"'] & & A \| T \arrow[dl, "A \| \tau"] \\
 & A \| B \| C \arrow[d] & \\
 & A \| C &
\end{tikzcd}
$$

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 (a.k.a. Courtesy) $\sigma : S \to A^{\perp} \parallel B$ 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

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.*

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

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.*

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

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
$$\begin{tikzcd} A \arrow[r, bend left=40, "\sigma", "\shortmid" marking] \arrow[r, bend right=40, "\sigma'"', "\shortmid" marking] \arrow[r, phantom, "\Downarrow f"] & B \end{tikzcd}$$
are maps $f : S \to S'$ such that
$$\begin{tikzcd} S \arrow[r, "f"] \arrow[dr, "\sigma"'] \arrow[dr, phantom, bend left=10, "="] & S' \arrow[d, "\sigma'"] \\ & A^\perp \| B \end{tikzcd}.$$

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.

DETERMINISTIC STRATEGIES

Deterministic strategies

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

$$\forall X \subseteq_{\mathrm{fin}} S.\ [X]^- \in \mathrm{Con}_S \Rightarrow X \in \mathrm{Con}_S\,,$$

where $[X]^- =_{\mathrm{def}} \{s' \in S \mid \exists s \in X.\ pol_S(s') = - \ \&\ s' \le s\}$.
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 $\oplus \rightsquigarrow \ominus$. The construction $\mathbb{CC}_A$:

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

To see $\mathbb{CC}_A$ 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

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

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

$\leadsto$ 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 $x \in F$, $\emptyset \overset{a_1}{\longrightarrow\!\!\!\subset} x_1 \overset{a_2}{\longrightarrow\!\!\!\subset} \cdots \overset{a_k}{\longrightarrow\!\!\!\subset} x_k = x$ within F;

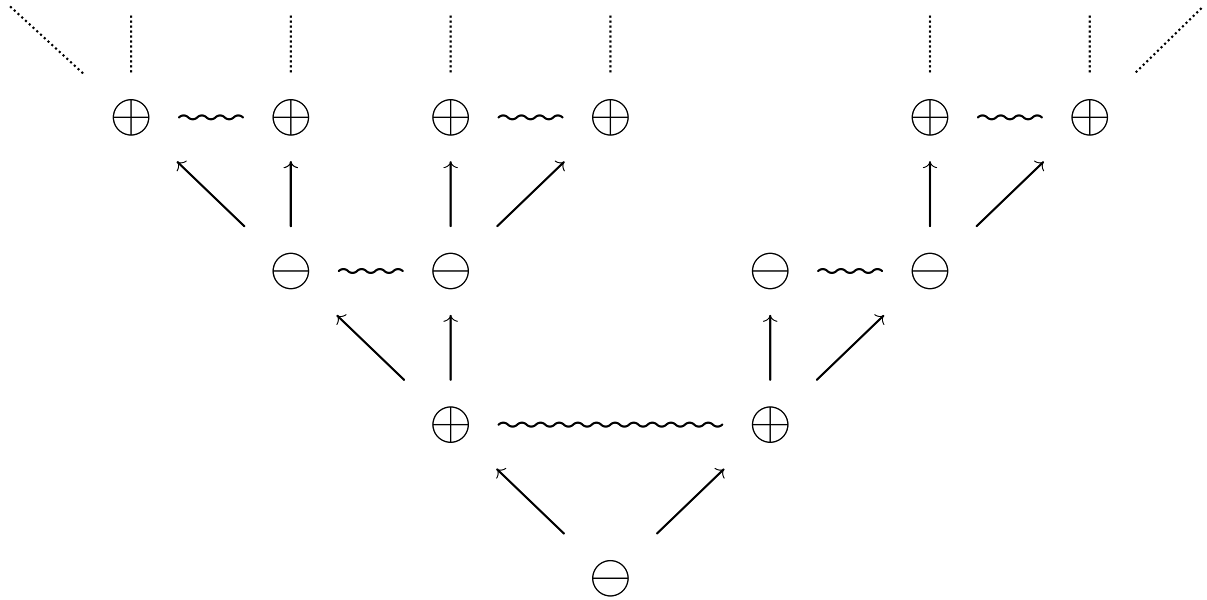
determinacy: If $x \overset{a}{\longrightarrow\!\!\!\subset}$ and $x \overset{a'}{\longrightarrow\!\!\!\subset}$ in F with $pol_A(a) = +$, then $x \cup \{a, a'\} \in F$;

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

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

1-stable: If $x_1 \overset{a}{\longrightarrow\!\!\!\subset} x$ and $x_2 \overset{b}{\longrightarrow\!\!\!\subset} x$ 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

Special cases

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 $x \in \mathcal{C}^\infty(S)$,

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

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.

Winning strategies in a game with winning conditions

A *game with winning conditions*: a game A with $W \subseteq \mathcal{C}^\infty(A)$

Let $\sigma : S \to A$ a strategy. $Informally,$

σ *is* winning *iff against a counter-strategy, the resulting plays give a win for Player.*

I.e., for any strategy, $\tau : T \to A^\perp$, the composition before hiding results in $\langle \sigma, \tau \rangle = \{(\tau \circledast \sigma)z \mid z \text{ maxl in } \mathcal{C}^\infty(T \circledast S)\} \subseteq W$.

$$\begin{tikzcd}
 & T \circledast S \arrow[dl] \arrow[dr] \arrow[dd, dotted, "\tau \circledast \sigma"] & \\
S \arrow[dr, "\sigma"'] & & T \arrow[dl, "\tau"] \\
 & A &
\end{tikzcd}$$

Lemma $\sigma : S \to A$ is winning iff $\sigma x \in W$, for all $+$-maximal $x \in \mathcal{C}^\infty(S)$.

Operations on games with winning conditions

Dual $(A, W)^\perp = (A^\perp, \mathcal{C}^\infty(A) \setminus W)$. To win is to not win in (A, W).

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

$$W_{A\|B} = W_A \| \mathcal{C}^\infty(B) \ \cup \ \mathcal{C}^\infty(A) \| W_B \,.$$

To win is to win in one game or the other.

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

Lemma The composition of winning strategies is winning.

$\rightsquigarrow$ a bicategory winning strategies.

From strategies to probabilistic strategies

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

Idea:

(1) Endow S with conditional probability (*via* valuation on configurations), 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, cf. Judea Pearl.

Concurrent strategies at the crossroads

Extend to *winning conditions* or *payoff*, *imperfect information*, *probabilistic*, *quantum* strategies, *parallel causes*. "Structural Game Theory." Determinacy.

But the strategies are *linear*, so disallow backtracking. One reason to introduce *symmetry* in games and strategies. Then can support (co)monads *up to symmetry* for copying, to break linearity. There are relations with *homotopy*.

A *semantics of classical proofs* as strategies; dependency between existential witnesses and universally quantified variables is revealed as causal dependency of Player on Opponent moves. $\rightsquigarrow$ A compositional proof of Herbrand's theorem.

Strategies as concurrent processes: A *language* and *structural operational semantics* for strategies. Subtle differences with the traditional process calculi.

Programs as strategies ...