Models of concurrency, categories, and games
Lecture 3

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

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

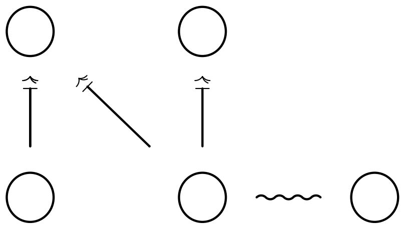
$$\{e\} \in \mathrm{Con} \text{ for all } e \in E,$$

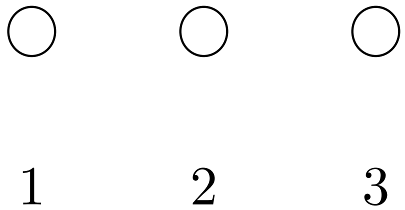
$$Y \subseteq X \in \mathrm{Con} \Rightarrow Y \in \mathrm{Con}, \quad \text{and}$$

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

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

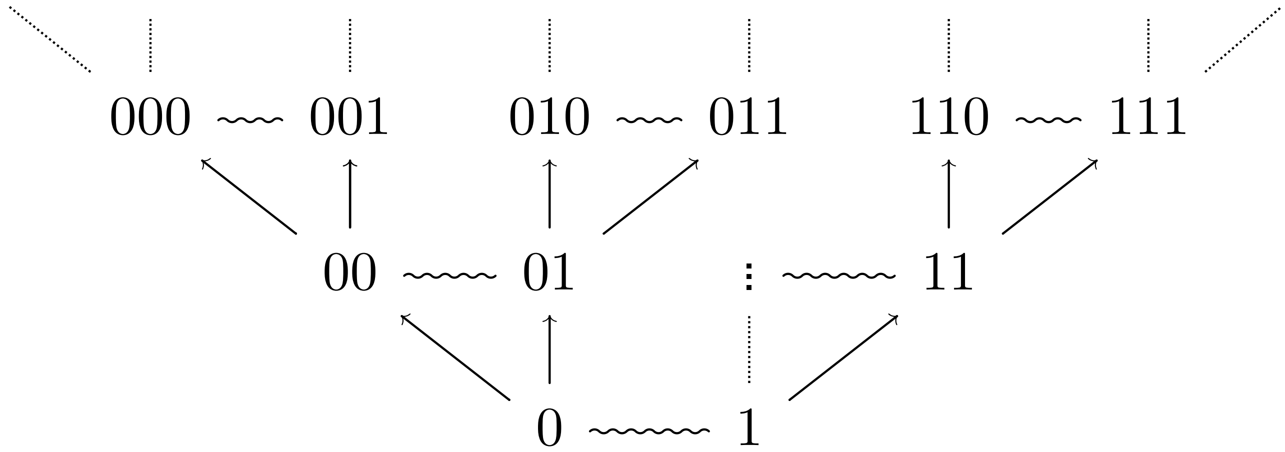
Down-closed: $\forall e, e'.\ e' \leq e \in x \Rightarrow e' \in x$.

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

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

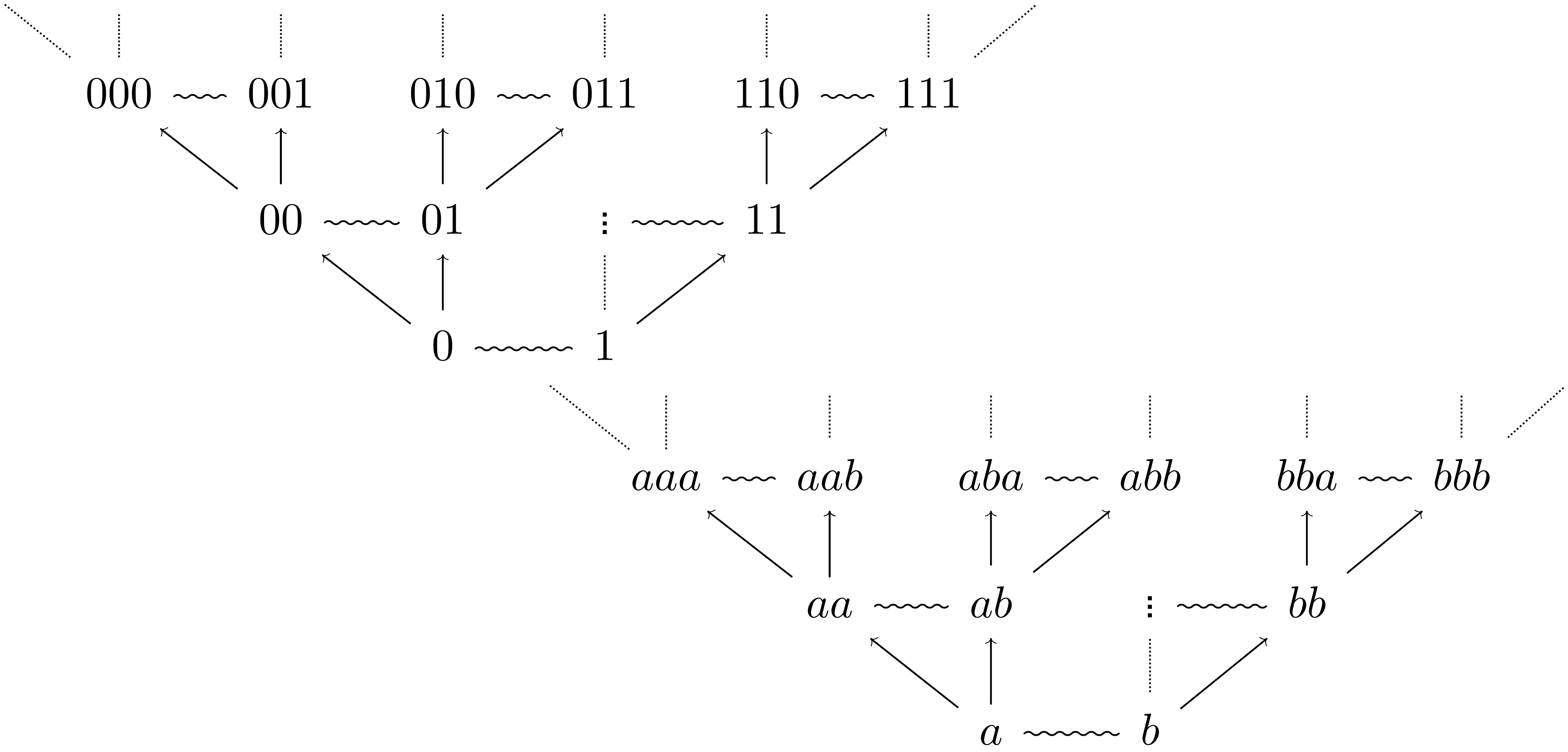
If E is countable, $(\mathcal{C}^\infty(E), \subseteq)$ is a Berry dI domain (and all such so obtained). Finite configurations: $\mathcal{C}(E)$.

Example: Streams as event structures



~~~ 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 *preserve concurrency*, and *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

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

Product—an example

$$
\begin{tikzcd}
b & & & & (b,*) & (b,*) \arrow[r, no head, squiggly] & (b,c) \arrow[ll, no head, squiggly, bend right] \\
& \times & & = & & & \\
a \arrow[uu] & & c & & (a,*) \arrow[uu] \arrow[r, no head, squiggly] \arrow[uurr] & (a,c) \arrow[uu] \arrow[r, no head, squiggly] & (*,c) \arrow[uu, no head, squiggly]
\end{tikzcd}
$$

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:

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

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.
$\rightsquigarrow$ *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:

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

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

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

Categories of stable families and event structures

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

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

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.

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

$(*) \quad [e]_x \subseteq z \iff [e]_x = [e]_z$, 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 $[e]_x = [e]_z$.

"$y \subseteq rhs$": $[e]_x \in y \Rightarrow [e]_x \subseteq \bigcup y \Rightarrow [e]_x = [e]_{\bigcup y} \in rhs$, 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. $f = G(g) \circ \eta_A$.

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

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

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 the right adjoint of 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

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

Product of event structures

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

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

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

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

(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.

...... (Adjunctions compose.)