

Event Structures and Games

Glynn Winskel

Strathclyde101, 26 November 2020

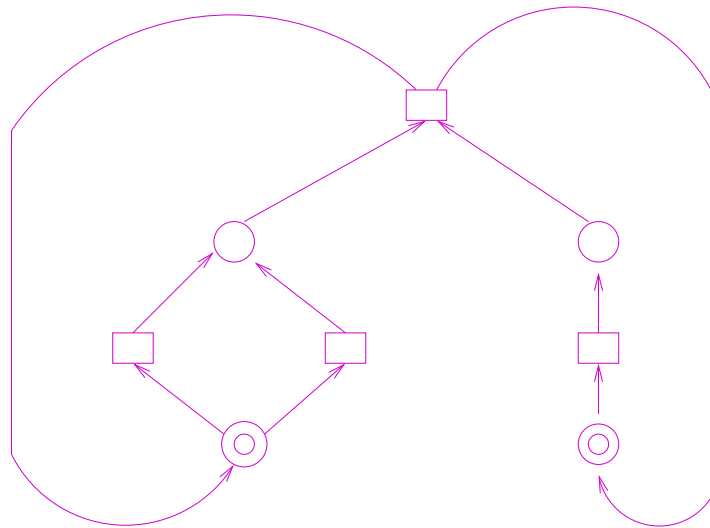
Event structures, a model based on causal dependency of events - concurrent analogue of trees. Locality \rightsquigarrow causal dependency.

\rightsquigarrow Broad applications, in security protocols, systems biology, weak memory, partial-order model checking, distributed computation, logic, semantics, ...

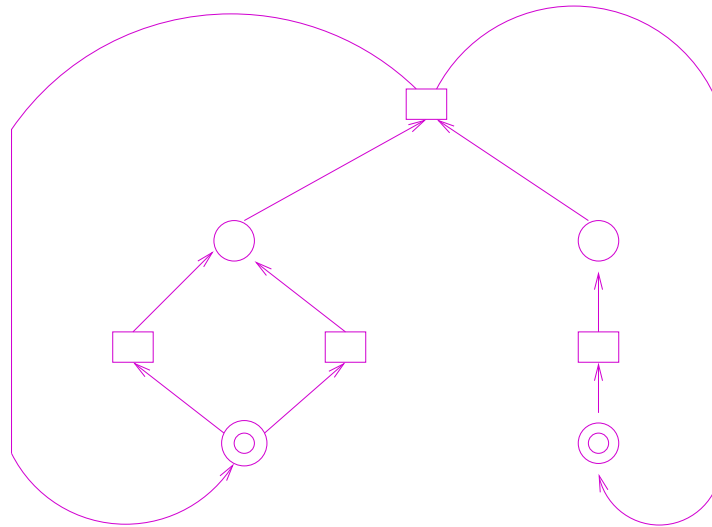
Distributed games, with behaviour based on event structures, rather than trees. Aim: to generalise domain theory, tackle its anomalies and limitations w.r.t. concurrency and quantitative aspects; repair the divides between denotational vs. operational, semantic vs. algorithmic.

Ways to compose winning and optimal strategies \rightsquigarrow *structural game theory*.

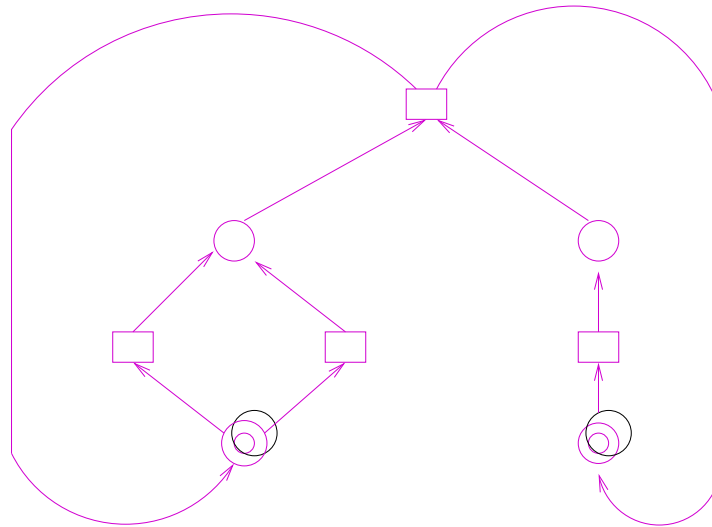
A (basic) Petri net



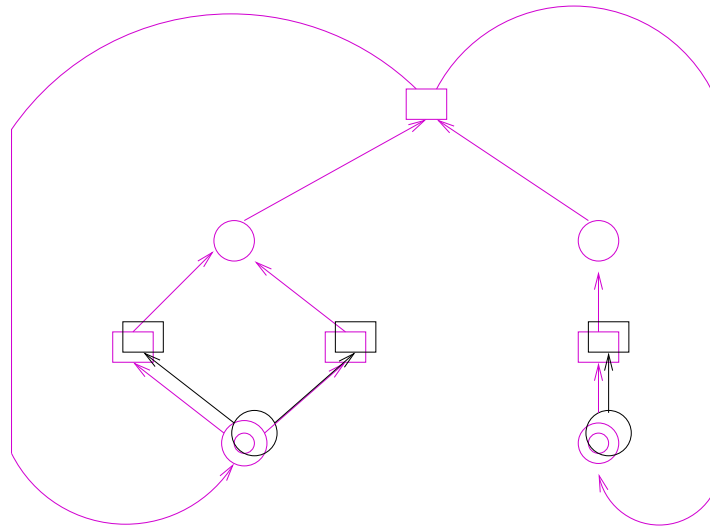
Unfolding a Petri net



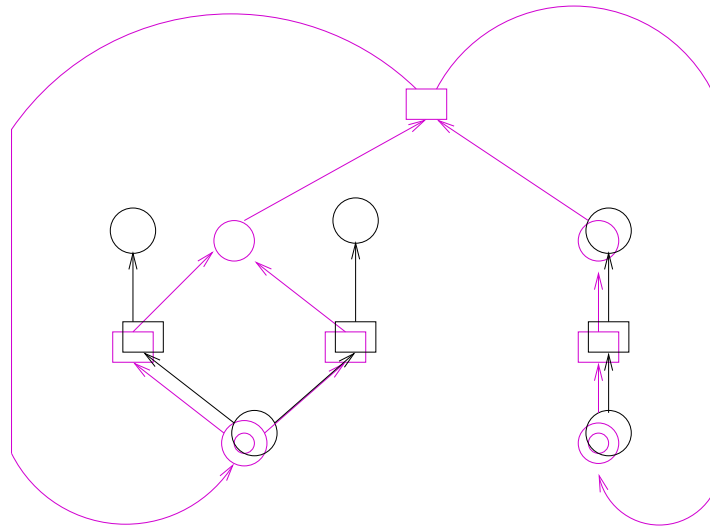
Unfolding a Petri net



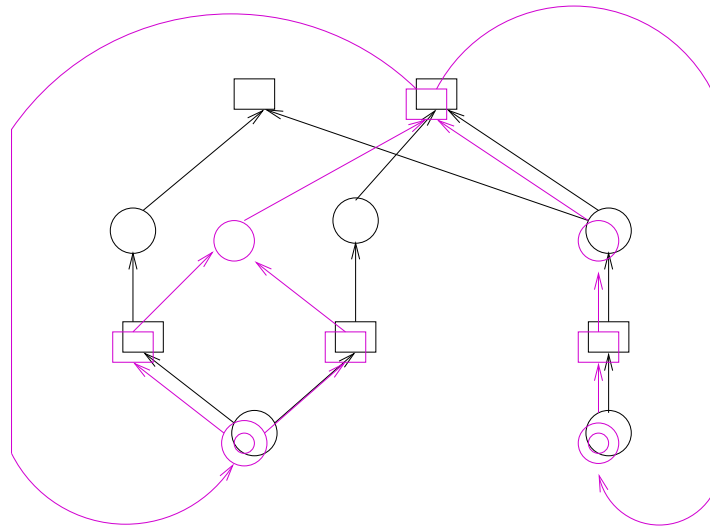
Unfolding a Petri net



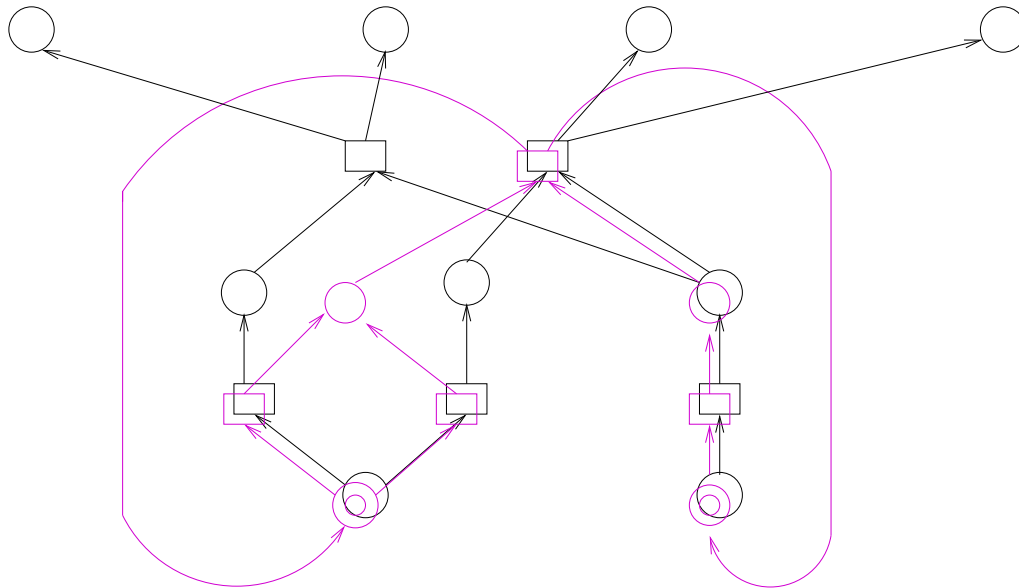
Unfolding a Petri net



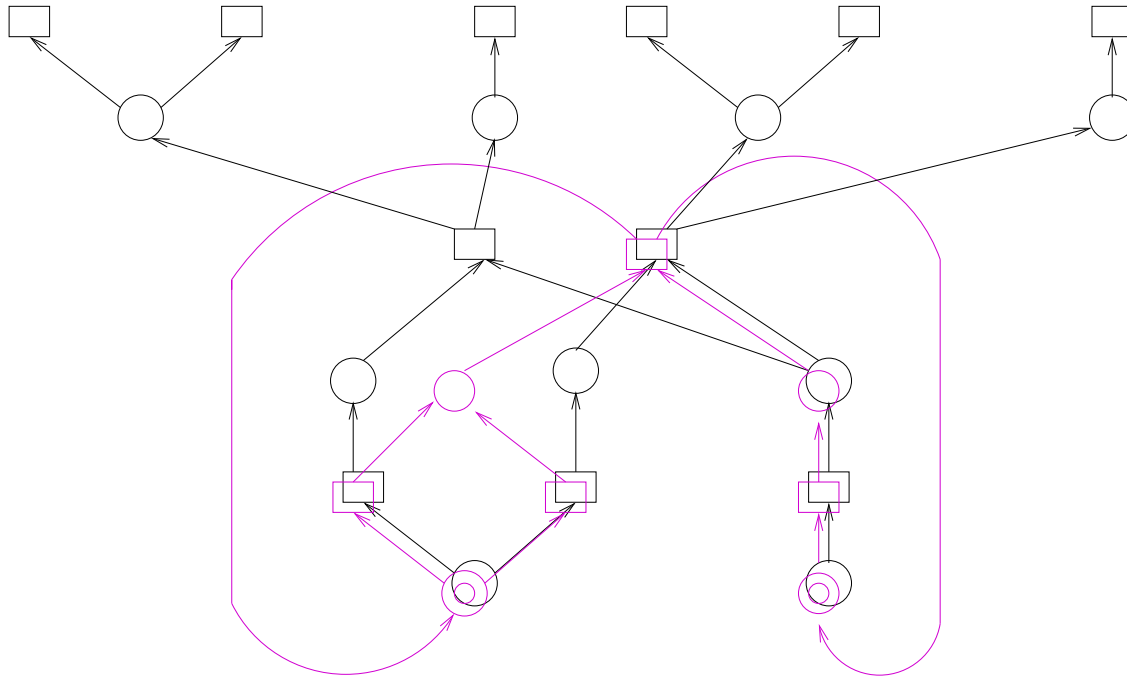
Unfolding a Petri net



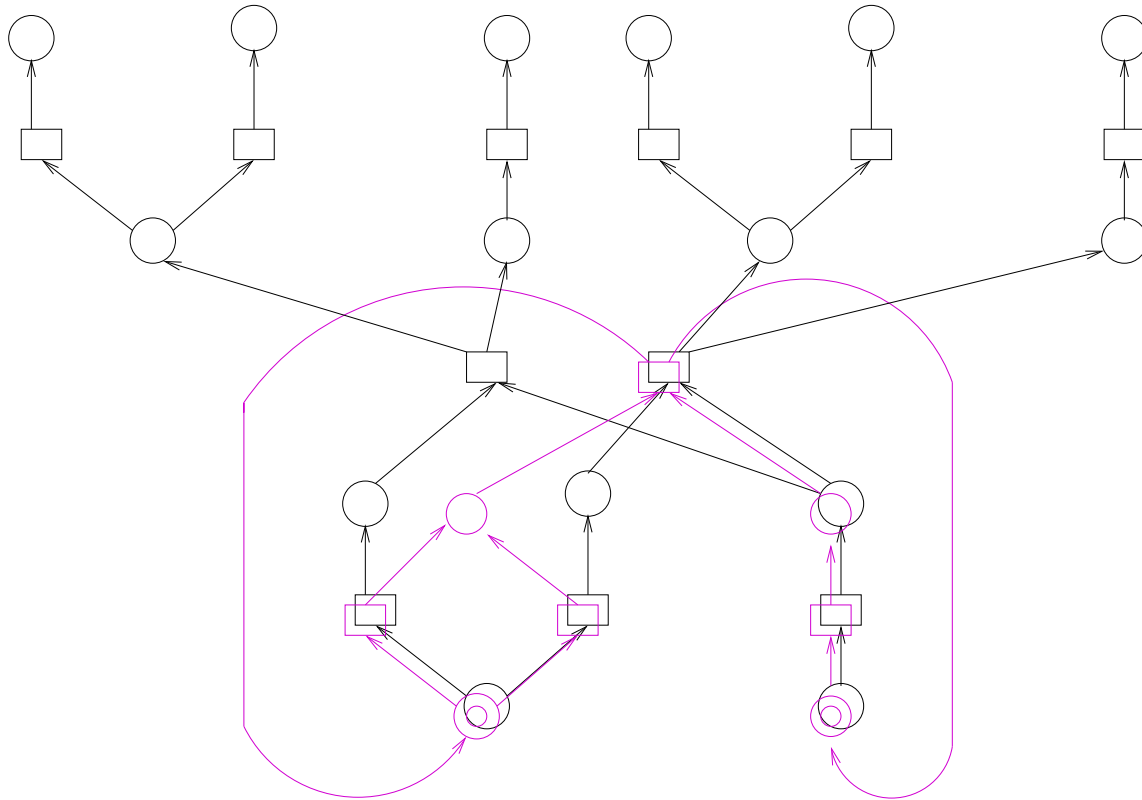
Unfolding a Petri net



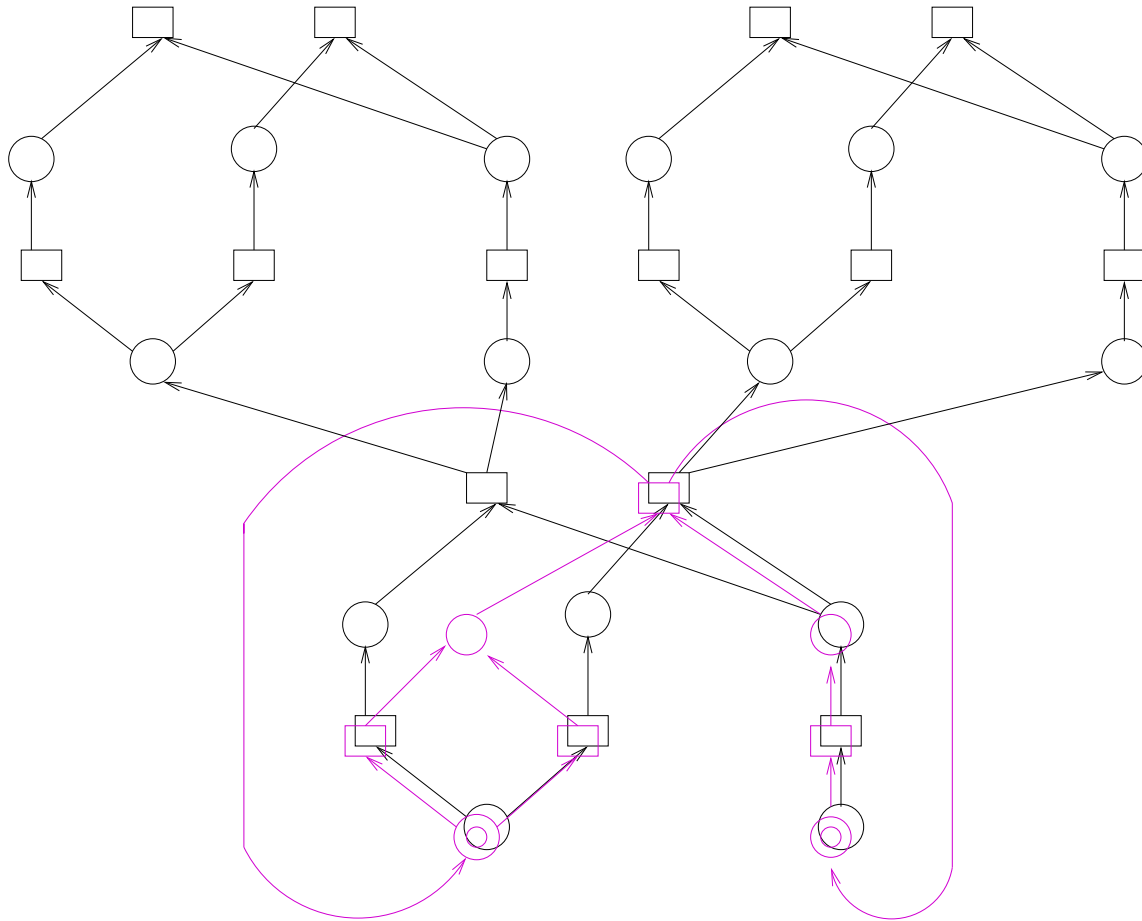
Unfolding a Petri net

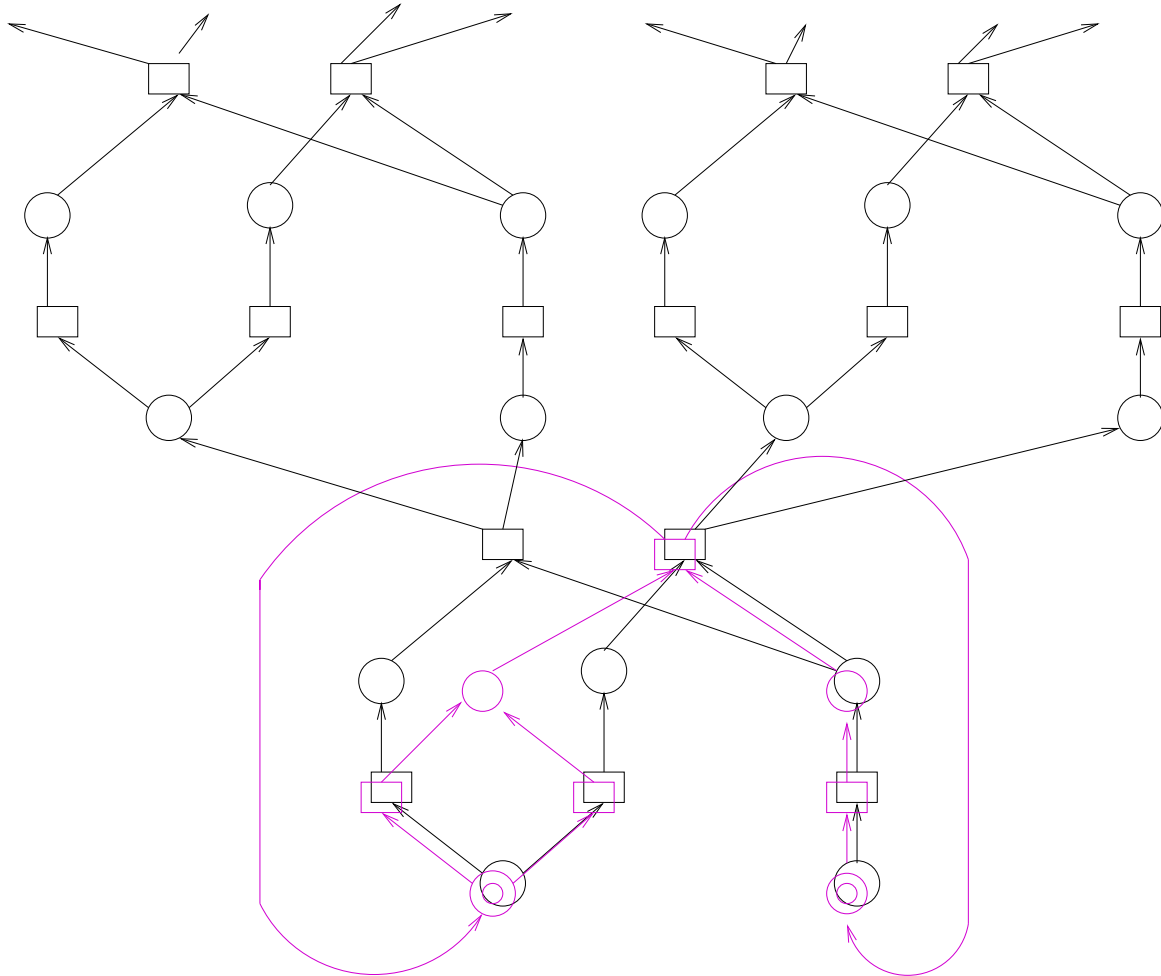


Unfolding a Petri net

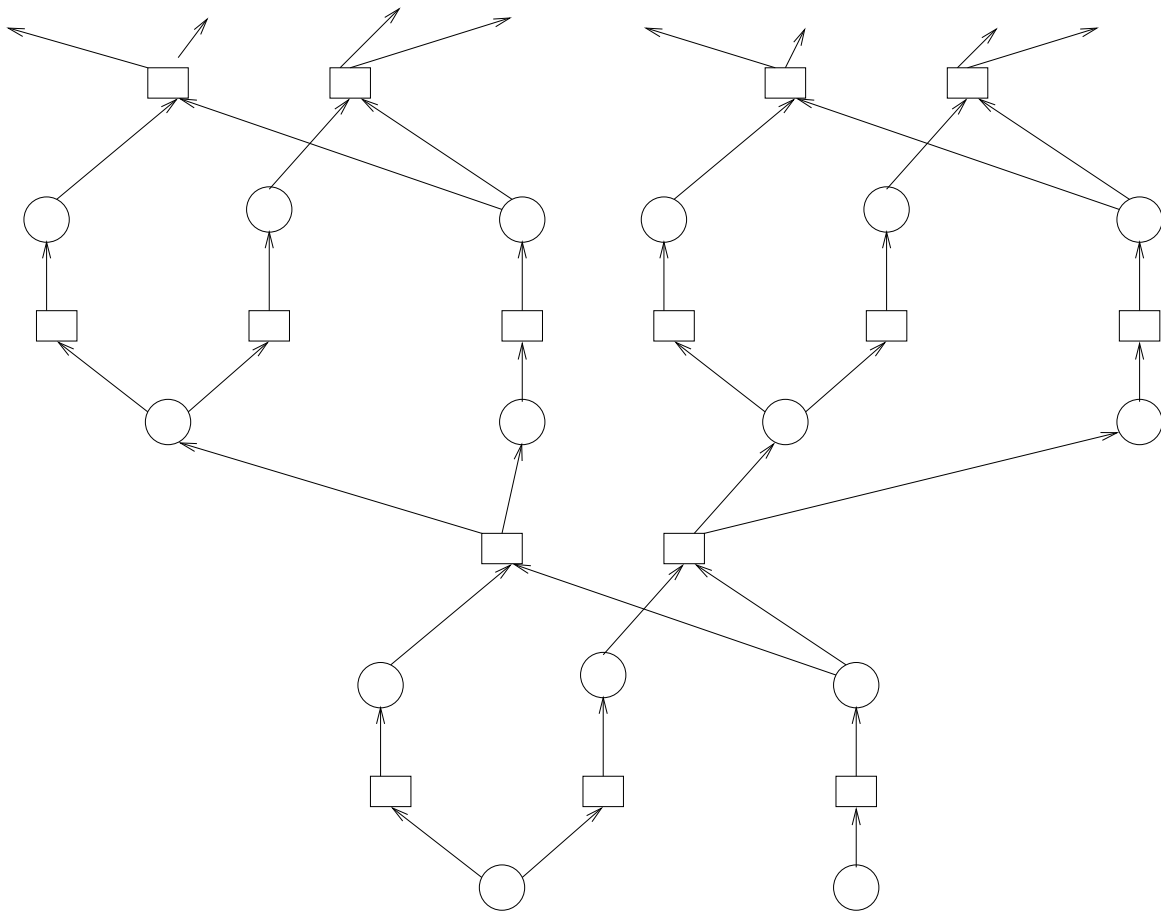


Unfolding a Petri net

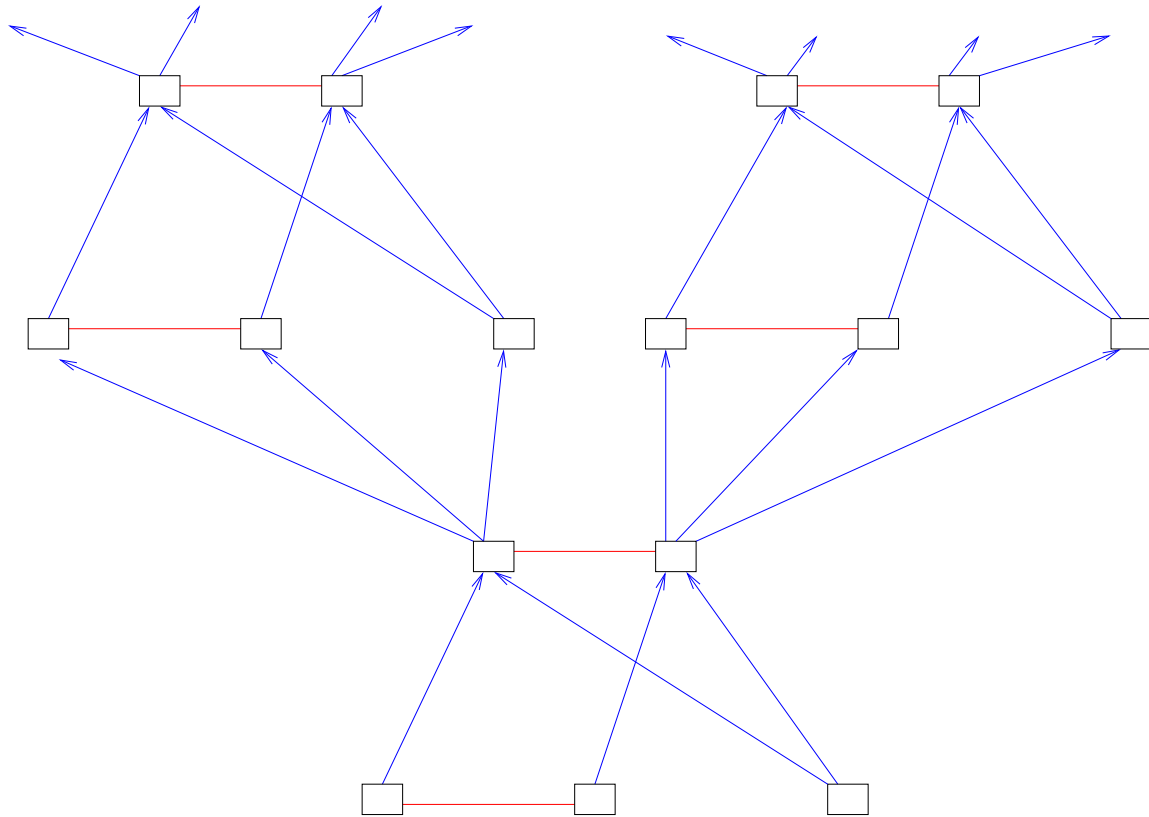




An occurrence net



An event structure



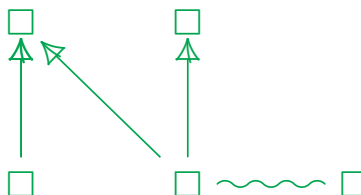
Event structures - the formal definition of the simplest kind

Definition

An **event structure** comprises $(E, \leq, \#)$, consisting of a set of **events** E

- partially ordered by \leq , the **causal dependency relation**, and
- a binary irreflexive symmetric relation, the **conflict relation**, which satisfy $\{e' \mid e' \leq e\}$ is finite and $e \# e' \leq e'' \implies e \# e''$.

Two events are **concurrent** when neither in conflict nor causally related.



Definition

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

Consistent: $\forall e, e' \in x. \neg(e \# e')$ and

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

Maps of event structures

Definition

A **map** of event structures $f : E \rightarrow E'$
is a partial function on events $f : E \rightarrow E'$ 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)*

Maps of event structures

Definition

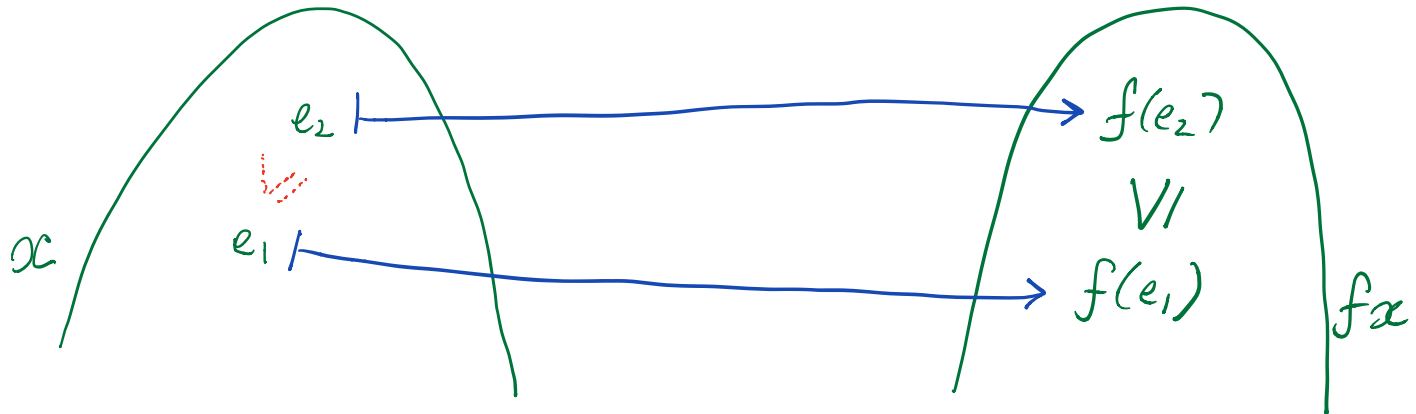
A **map** of event structures $f : E \rightarrow E'$
is a partial function on events $f : E \rightarrow E'$ such that

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)*

Maps preserve concurrency, and locally reflect causal dependency:

$$\forall x \in \mathcal{C}(E), e_1, e_2 \in x. f(e_1) \leq f(e_2) \implies e_1 \leq e_2.$$



Maps of event structures

Definition

A **map** of event structures $f : E \rightarrow E'$
is a partial function on events $f : E \rightarrow E'$ 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)*



Semantics of synchronising processes [Hoare, Milner] can be expressed in terms of **universal constructions** on event structures;
in games, pullbacks and partial-total factorisation play a central role.

Relations between models via **adjunctions**.

E.g. the event-structure unfolding of a basic Petri net is a right adjoint.

Coreflection of event structures in stable families v useful for constructions.

Strong bisimulation via **open maps**, defined diagrammatically.

\rightsquigarrow Preheaf models for concurrency ...

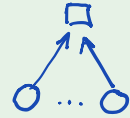
Symmetry as “self bisimulation” helps compensate for the overly-concrete nature of models for concurrency. *E.g.*, unfolding of Petri nets with multiplicities defined universally only **up to symmetry**.

Remark: Petri nets as containers (Thanks Fredrik!)

A basic Petri net with events E and conditions B can be seen as a pair of containers, one associating

events with their preconditions: $(E \triangleright Pre)$ where $Pre : E \rightarrow \mathcal{P}(B)$

shapes
↘
positions



events with their postconditions: $(E \triangleright Post)$ where $Post : E \rightarrow \mathcal{P}(B)$



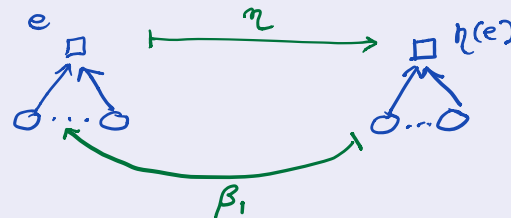
(Its initial marking identified with the postconds of a distinguished initial event)

A (total) map $(\eta, \beta) : N \rightarrow N'$ of basic Petri nets can be reformulated as a pair of container maps :

$$(\eta, \beta_1) : (E \triangleright Pre) \rightarrow (E' \triangleright Pre')$$

$$(\eta, \beta_2) : (E \triangleright Post) \rightarrow (E' \triangleright Post')$$

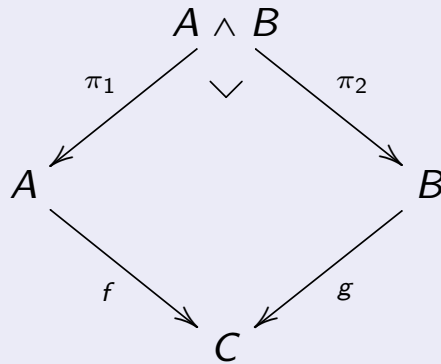
(with $\eta : E \rightarrow E'$ preserving initial events, ...)



Quantum Petri nets are being used to formalise quantum strategies.

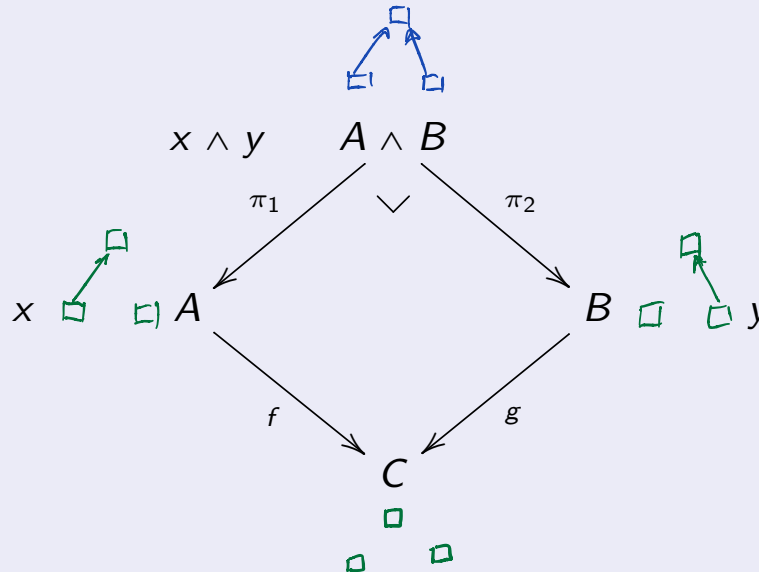
Pullbacks - for composing processes with a common interface

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



Pullbacks - for composing processes with a common interface

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



Finite configurations of $A \wedge B$ correspond to the composite bijections

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

between configurations $x \in \mathcal{C}(A)$ and $y \in \mathcal{C}(B)$ s.t. $fx = gy$ which are **secured bijections**, i.e. for which the transitive relation generated on $x \wedge y$ by

$$(a, b) \leq (a', b') \text{ if } a \leq_A a' \text{ or } b \leq_B b'$$

is a partial order.

Defined part of a map - for hiding

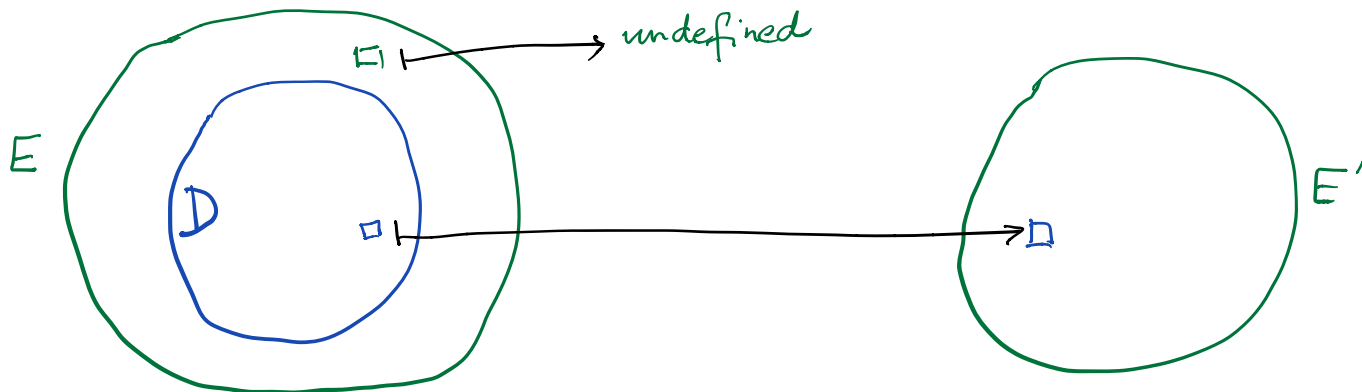
A partial map

$$f : E \rightarrow E'$$

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

$$E \xrightarrow{p} D \xrightarrow{t} E'$$

where $t : D \rightarrow E'$ is the **defined part** of f .



Games, a paradigm for interaction [Conway, Joyal]

The dichotomy *Player* vs. *Opponent* has many readings:
Team of Players vs. *Team of Opponents*; *Allies* vs. *Enemies*;
Prover vs. *Disprover*; *Process* vs. *Environment*

Operations on (2-party) games:

Dual game G^\perp - interchange the role of *Player* and *Opponent*;
Counter-strategy = strategy for *Opponent* = strategy for *Player* in dual game.
Parallel composition of games $G \parallel H$.

A strategy (for *Player*) *from* a game G *to* a game H is a strategy in $G^\perp \parallel H$.
 A strategy (for *Player*) *from* a game H *to* a game K is a strategy in $H^\perp \parallel K$.
Compose by letting them play against each other in the common game H .

The *Copycat* strategy in $G^\perp \parallel G$, so from G to G ...

Concurrent games

Games and strategies are represented by *event structures with polarity*, an event structure where events carry a polarity \boxplus / \boxminus (*Player*/*Opponent*).

Maps are those of event structures which preserve polarity.

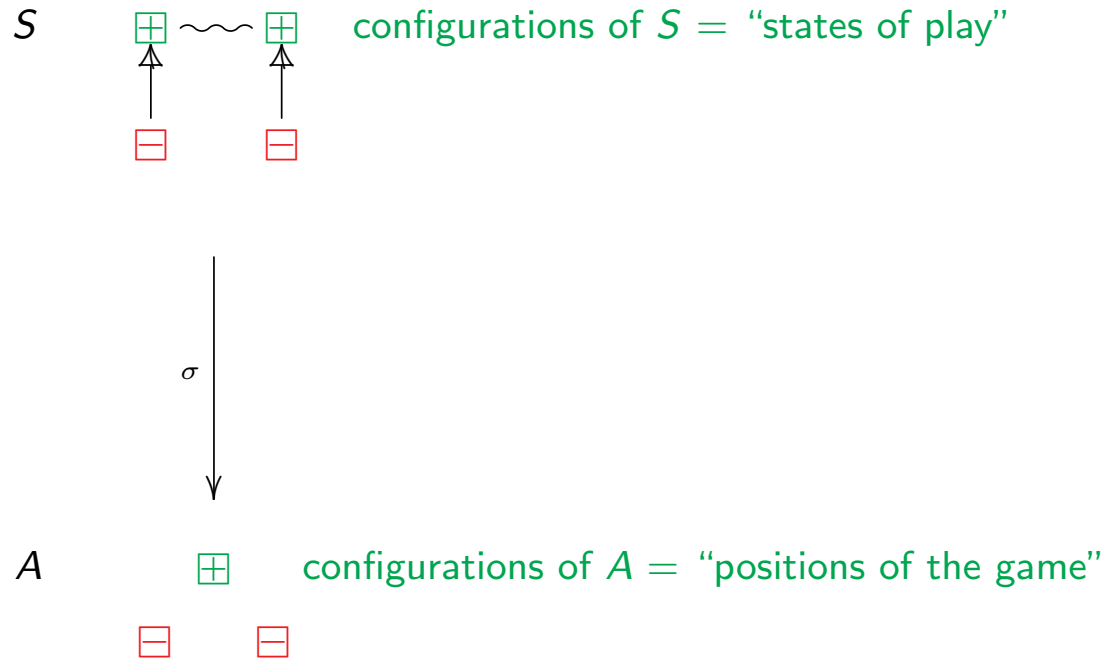
Dual, B^\perp , of an event structure with polarity B is a copy of the event structure B with a reversal of polarities; this switches the roles of *Player* and *Opponent*.

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

A strategy *from* a game A *to* a game B is a strategy in $A^\perp \parallel B$, written

$$\sigma : A \multimap B$$

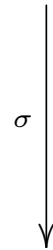
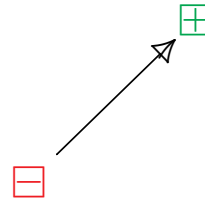
A strategy in a game A is a special total map $\sigma : S \rightarrow A$, e.g.



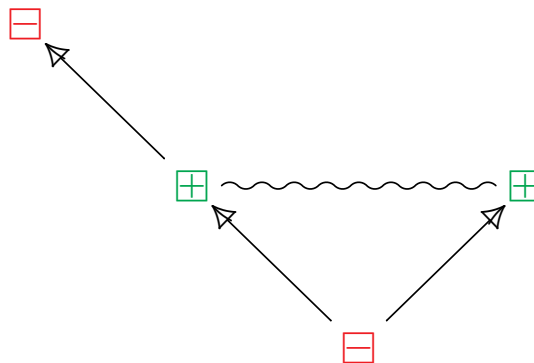
The strategy: answer either move of Opponent by the Player move.

When games are trees

S

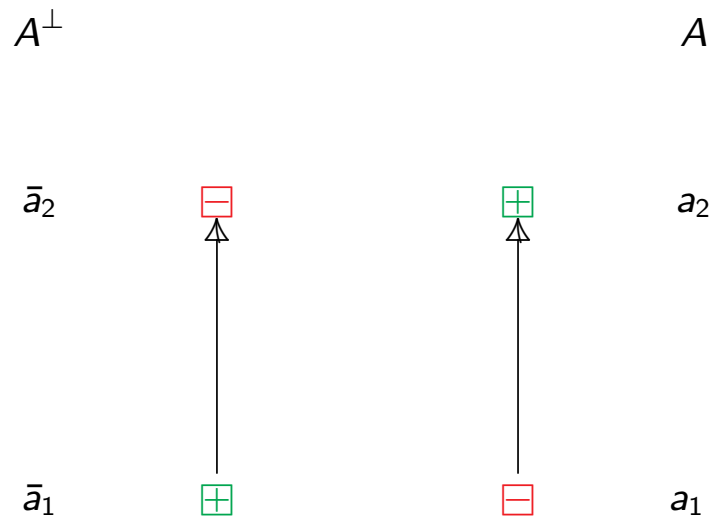


A

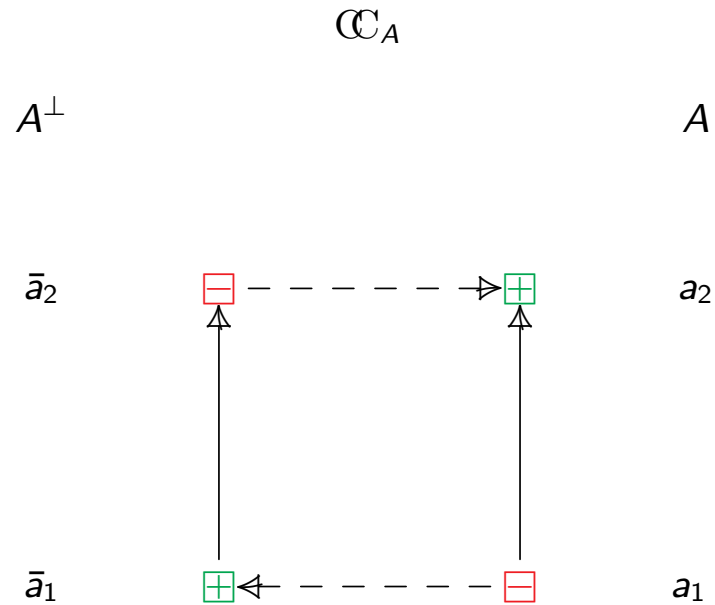


The strategy: force Opponent to get stuck.

Copycat strategy from A to A



Copycat strategy from A to A



Composition of strategies $\sigma : A \multimap B$ and $\tau : B \multimap C$

To compose

$$\begin{array}{ccc} S & & T \\ \sigma \downarrow & & \downarrow \tau \\ A^\perp \parallel B & & B^\perp \parallel C \end{array}$$

synchronise complementary moves over common game B via pullback:

$$\begin{array}{ccccc} & & T \circledast S & & \\ & \swarrow & \downarrow & \searrow & \\ S \parallel C & & \tau \circledast \sigma & & A \parallel T \\ & \swarrow \sigma \parallel C & & \swarrow A \parallel \tau & \\ & A \parallel B \parallel C & & & \end{array}$$

Composition of strategies $\sigma : A \multimap B$ and $\tau : B \multimap C$

To compose

$$\begin{array}{ccc}
 S & & T \\
 \sigma \downarrow & & \downarrow \tau \\
 A^\perp \parallel B & & B^\perp \parallel C
 \end{array}$$

synchronise complementary moves over common game B via pullback; then hide synchronisations via partial-total factorisation:

$$\begin{array}{ccc}
 \text{before hiding} & T \circledast S & \cdots \cdots \cdots > T \odot S & \text{after hiding} \\
 \tau \circledast \sigma \downarrow & & & \downarrow \tau \odot \sigma \\
 A^\perp \parallel B \parallel C & \cdots \cdots \cdots > & A^\perp \parallel C
 \end{array}$$

Conditions on a strategy are those needed to make copycat identity w.r.t. composition.

For copycat to be identity w.r.t. composition

a **strategy** in a game A has to be $\sigma : S \rightarrow A$, a total map of event structures with polarity, which is

(i) whenever $\sigma x \subseteq^- y$ in $\mathcal{C}(A)$ there is a unique $x' \in \mathcal{C}(S)$ s.t.

$x \subseteq x'$ & $\sigma x' = y$, i.e.

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

A strategy should be receptive to Opponent moves allowed by the game.

(ii) whenever $y \subseteq^+ \sigma x$ in $\mathcal{C}(A)$ there is a (necessarily unique) $x' \in \mathcal{C}(S)$ s.t.

$x' \subseteq x$ & $\sigma x' = y$, i.e.

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

A strategy should only adjoin immediate causal dependencies $\boxed{-} \rightarrow \boxed{+}$.

For copycat to be identity w.r.t. composition

a **strategy** in a game A has to be $\sigma : S \rightarrow A$, a total map of event structures with polarity, which is

(i) whenever $\sigma x \subseteq^- y$ in $\mathcal{C}(A)$ there is a unique $x' \in \mathcal{C}(S)$ s.t.

$x \subseteq x'$ & $\sigma x' = y$, i.e.

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

A strategy should be receptive to Opponent moves allowed by the game.

(ii) whenever $y \subseteq^+ \sigma x$ in $\mathcal{C}(A)$ there is a (necessarily unique) $x' \in \mathcal{C}(S)$ s.t.

$x' \subseteq x$ & $\sigma x' = y$, i.e.

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

A strategy should only adjoin immediate causal dependencies $\boxminus \rightarrow \boxplus$.

\rightsquigarrow compact-closed bicategory of concurrent games and strategies.

Strategies as profunctors

Defining 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 partial order and a factorization system:

$$\begin{array}{c} x \\ \text{---} \sqsubseteq \text{---} \\ \text{---} \sqsubseteq^+ \text{---} \\ \text{---} \sqcup \text{---} \\ \exists! z. \quad y \supseteq^- z. \end{array}$$

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

Theorem Strategies σ correspond to discrete fibrations, i.e.,

$$\begin{array}{ccc} \exists! x'. & x' & \cdots \sqsubseteq_S \cdots x \\ & \downarrow \sigma'' & \downarrow \sigma'' \\ & y & \sqsubseteq_A \sigma x, \end{array}$$

which preserve \supseteq^- , \subseteq^+ and \emptyset . So strategies $\sigma : A \multimap B$ correspond to (certain) profunctors $\sigma'' : (\mathcal{C}(A), \sqsubseteq_A) \multimap (\mathcal{C}(B), \sqsubseteq_B)$.

\rightsquigarrow Lax functors from strategies to profunctors, and to Scott domains ...

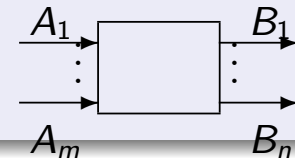
A language for concurrent strategies

Types: Games A, B, C, \dots with operations $A^\perp, A \parallel B$, sums $\sum_{i \in I} A_i$, recursively-defined types, \dots

A term

$$x_1 : A_1, \dots, x_m : A_m \vdash t \dashv y_1 : B_1, \dots, y_n : B_n,$$

denotes a strategy from $A_1 \parallel \dots \parallel A_m$ to $B_1 \parallel \dots \parallel B_n$.



Idea: t denotes a strategy $S \rightarrow \vec{A}^\perp \parallel \vec{B}$.

The term t describes witnesses, finite configurations of S , to a relation between finite configurations \vec{x} of \vec{A} and \vec{y} of \vec{B} . Cf. profunctors.

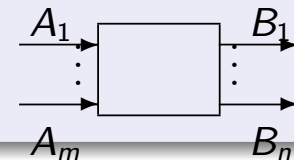
A language for concurrent strategies

Types: Games A, B, C, \dots with operations A^\perp , $A \parallel B$, sums $\sum_{i \in I} A_i$, recursively-defined types, \dots

A term

$$x_1 : A_1, \dots, x_m : A_m \vdash t \dashv y_1 : B_1, \dots, y_n : B_n,$$

denotes a strategy from $A_1 \parallel \dots \parallel A_m$ to $B_1 \parallel \dots \parallel B_n$.



Copypat $x : A \vdash y \sqsubseteq_A x \dashv y : A$ and other terms “wiring in” causality.

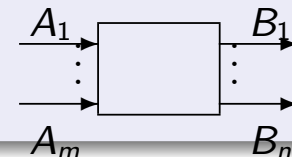
A language for concurrent strategies

Types: Games A, B, C, \dots with operations $A^\perp, A \parallel B$, sums $\sum_{i \in I} A_i$, recursively-defined types, \dots

A term

$$x_1 : A_1, \dots, x_m : A_m \vdash t \dashv y_1 : B_1, \dots, y_n : B_n,$$

denotes a strategy from $A_1 \parallel \dots \parallel A_m$ to $B_1 \parallel \dots \parallel B_n$.



Copycat $x : A \vdash y \sqsubseteq_A x \dashv y : A$ and other terms “wiring in” causality.

Composition
$$\frac{\Gamma \vdash t \dashv \Delta \quad \Delta \vdash u \dashv H}{\Gamma \vdash \exists \Delta. [t \parallel u] \dashv H}$$

Duality
$$\frac{A, \Gamma \vdash t \dashv \Delta}{\Gamma \vdash t \dashv A^\perp, \Delta}$$

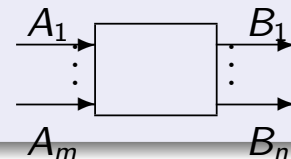
A language for concurrent strategies

Types: Games A, B, C, \dots with operations $A^\perp, A \parallel B$, sums $\sum_{i \in I} A_i$, recursively-defined types, \dots

A term

$$x_1 : A_1, \dots, x_m : A_m \vdash t \dashv y_1 : B_1, \dots, y_n : B_n,$$

denotes a strategy from $A_1 \parallel \dots \parallel A_m$ to $B_1 \parallel \dots \parallel B_n$.

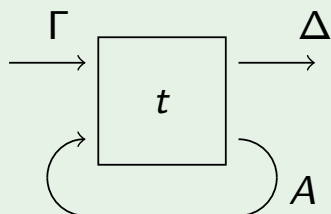


Copycat $x : A \vdash y \sqsubseteq_A x \dashv y : A$ and other terms “wiring in” causality.

Composition $\frac{\Gamma \vdash t \dashv \Delta \quad \Delta \vdash u \dashv H}{\Gamma \vdash \exists \Delta. [t \parallel u] \dashv H}$

Duality $\frac{A, \Gamma \vdash t \dashv \Delta}{\Gamma \vdash t \dashv A^\perp, \Delta}$

Feedback



$$\Gamma \vdash \exists x : A, y : A^\perp. [x \sqsubseteq_A y \parallel t] \dashv \Delta$$

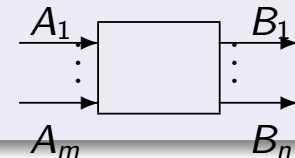
A language for concurrent strategies

Types: Games A, B, C, \dots with operations $A^\perp, A \parallel B$, sums $\sum_{i \in I} A_i$, recursively-defined types, \dots

A term

$$x_1 : A_1, \dots, x_m : A_m \vdash t \dashv y_1 : B_1, \dots, y_n : B_n,$$

denotes a strategy from $A_1 \parallel \dots \parallel A_m$ to $B_1 \parallel \dots \parallel B_n$.



Copycat $x : A \vdash y \sqsubseteq_A x \dashv y : A$ and other terms “wiring in” causality.

Composition
$$\frac{\Gamma \vdash t \dashv \Delta \quad \Delta \vdash u \dashv H}{\Gamma \vdash \exists \Delta. [t \parallel u] \dashv H}$$

Duality
$$\frac{A, \Gamma \vdash t \dashv \Delta}{\Gamma \vdash t \dashv A^\perp, \Delta}$$

Sum $\sum_{i \in I} t_i$

Conjunction $t_1 \wedge t_2$

Special cases - recovering functions 1

A concurrent strategy is **deterministic** when conflicting behaviour of Player implies conflicting behaviour of Opponent.

Stable spans and stable functions The sub-bicategory where the events of games are purely +ve is that of **stable spans** used in nd dataflow; feedback given by trace.

$$\begin{array}{c} S \\ \downarrow \sigma \\ A^\perp \parallel B \end{array}$$

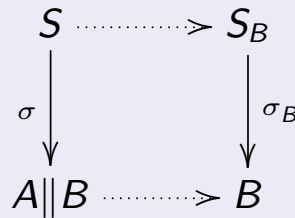
Its deterministic sub-bicategory **Stable** is equivalent to **stable functions between Berry domains** (coherent w.r.t. countable event structures with binary conflict); Girard's **coherence spaces** when causal dependency trivial.

Open games?

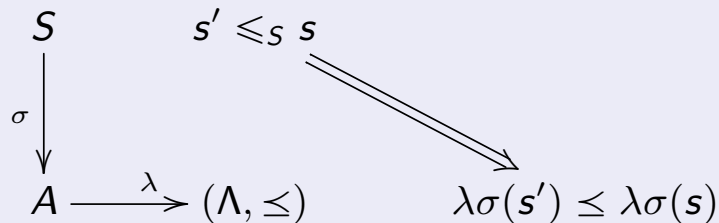
Special cases - recovering functions 2

Two tools for recovering functions in **parts** of a game:

1. Projecting a strategy to a parallel component of a game yields a strategy:



2. *Imperfect information* via an “access order” (Λ, \leq) on moves of the game; causal dependency of the game and additional causal dependencies of the strategy must respect it:



Open games via a dialectica category (thanks Jules!), e.g.

The dialectical category with maps

$(f, g) : \begin{pmatrix} X \\ R \end{pmatrix} \rightarrow \begin{pmatrix} Y \\ S \end{pmatrix}$ where $f : X \rightarrow Y$ and $g : X \times S \rightarrow R$ in **Stable**
embeds fully and faithfully in the sub-bicategory of strategies comprising deterministic strategies in games

$$(X^+ \parallel R^-) \multimap (Y^+ \parallel S^-) = (X^+ \parallel R^-)^\perp \parallel (Y^+ \parallel S^-) \text{ with access order}$$

$$X^- < Y^+$$

$$\wedge$$

$$R^+ > S^-$$

Their deterministic counterstrategies correspond to configurations of X paired with $h : Y \rightarrow S$ in **Stable**:

$$X^+ < Y^-$$

$$\wedge$$

$$R^- > S^+$$

Now have all the ingredients for open games w.r.t. **Stable** (and **Stable spans**).