

Models for Concurrency and Games

Glynn Winskel

The two ingredients

Models: Petri nets, event structures, Mazurkiewicz trace languages, transition systems, ...

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

This lecture should give an idea of

- **partial-order models**, a form of model becoming important in a range of areas from security, systems, model checking, systems biology, to proof theory
- why I believe such models will become central in **semantics** of computation and can combine the two approaches, **operational** and **denotational** semantics through the medium of games

Games informally

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

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

Two important operations on games: **parallel composition** of games $G \parallel H$; **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 , $G \rightarrow H$, is a strategy in $G^\perp \parallel H$; strategies compose with identities given by copy-cat. A strategy in G corresponds to a strategy from the empty game \emptyset to G . So

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

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

Games in a model for concurrency

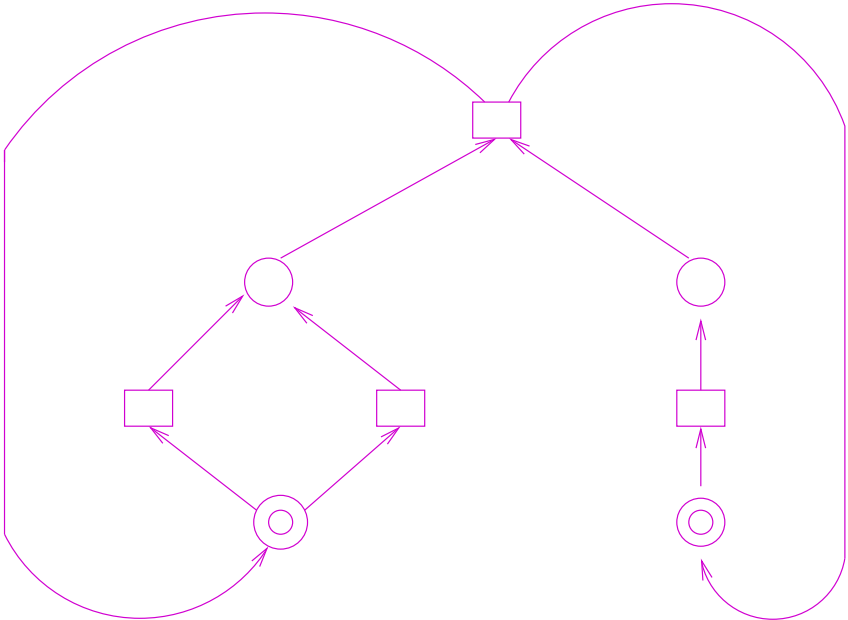
Lead to

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

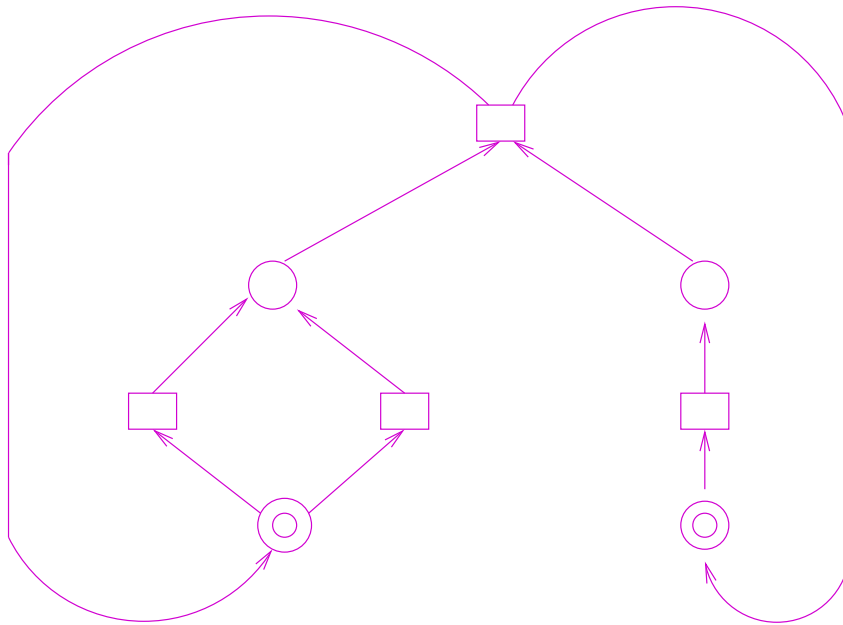
Causal/partial-order models

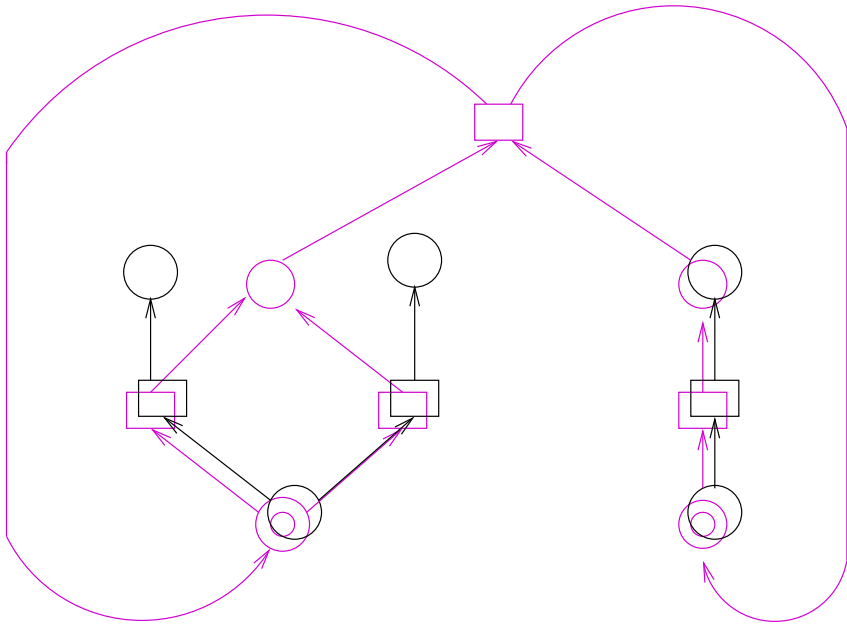
their range and applications ...

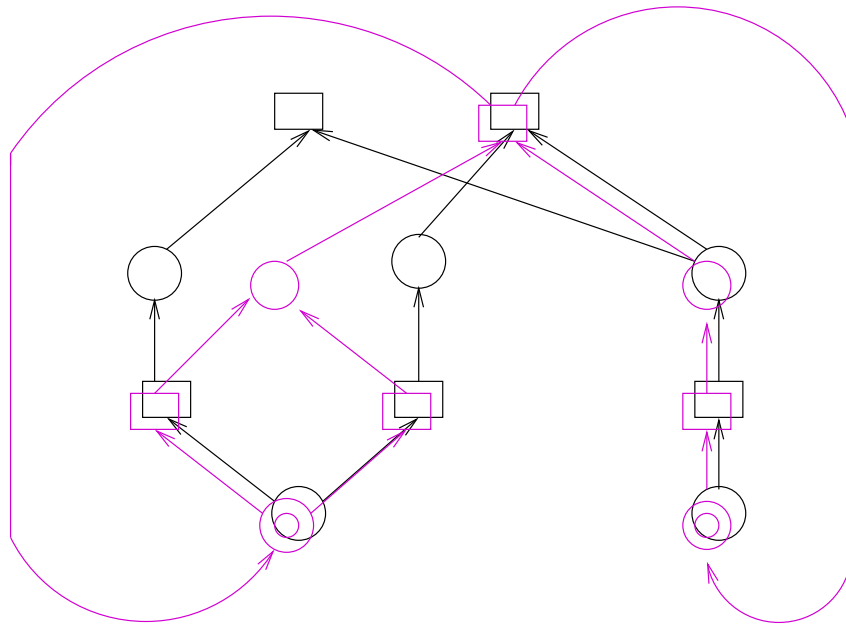
A (safe) Petri net

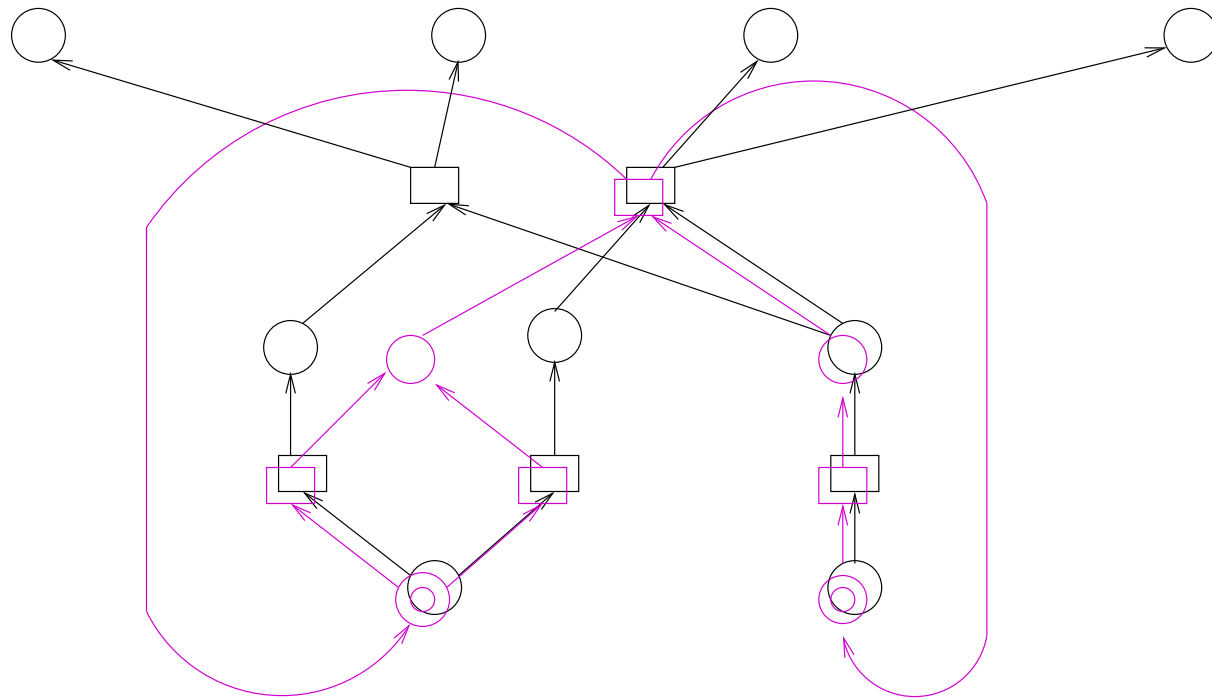


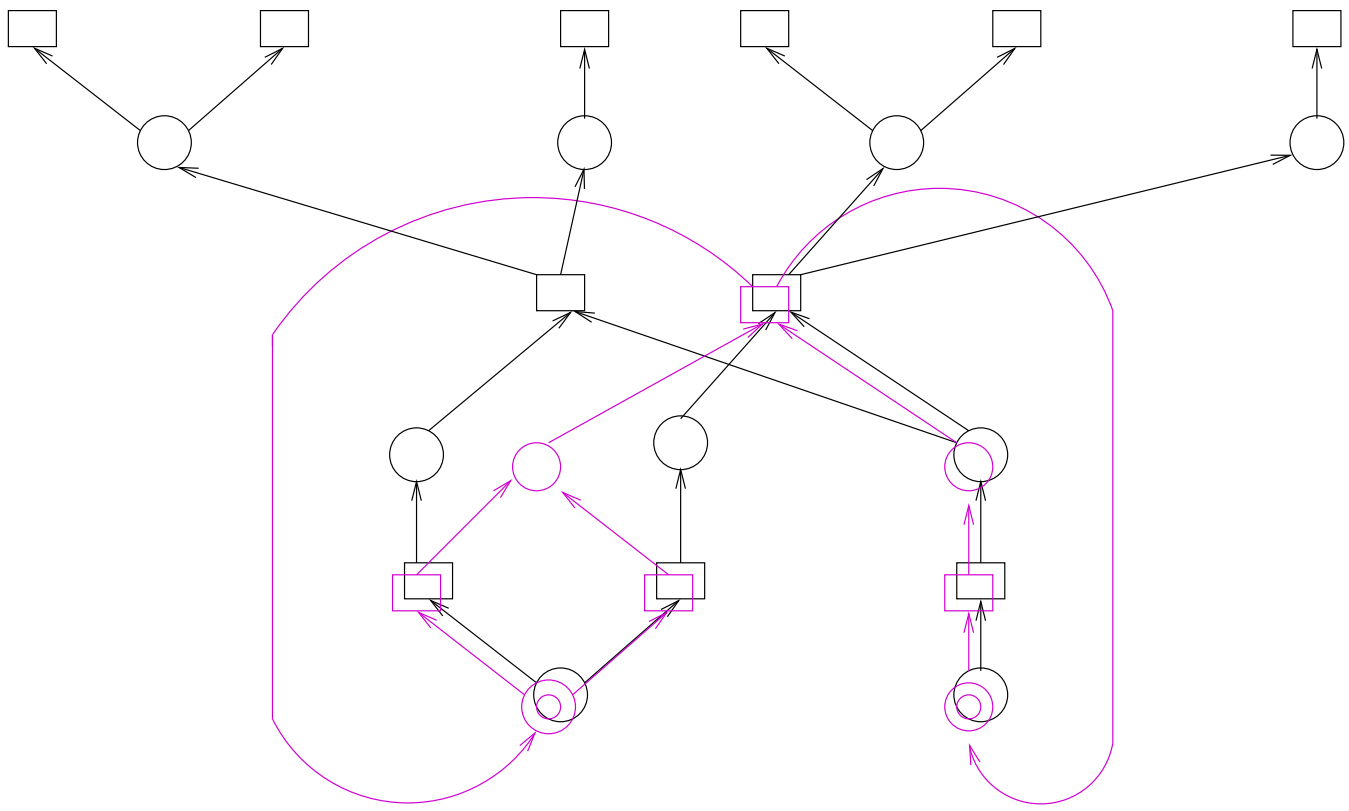
Unfolding a (safe) Petri net:

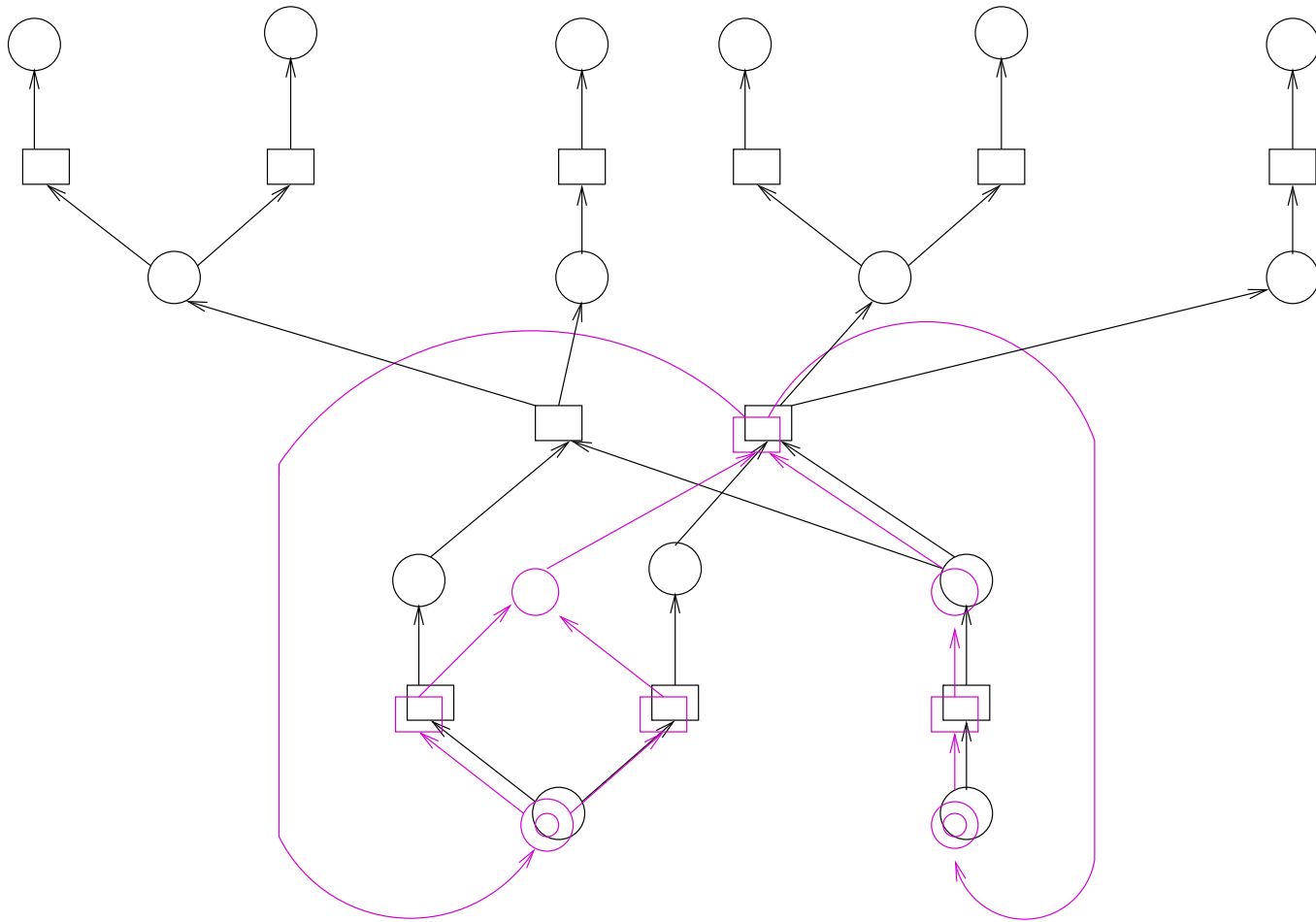


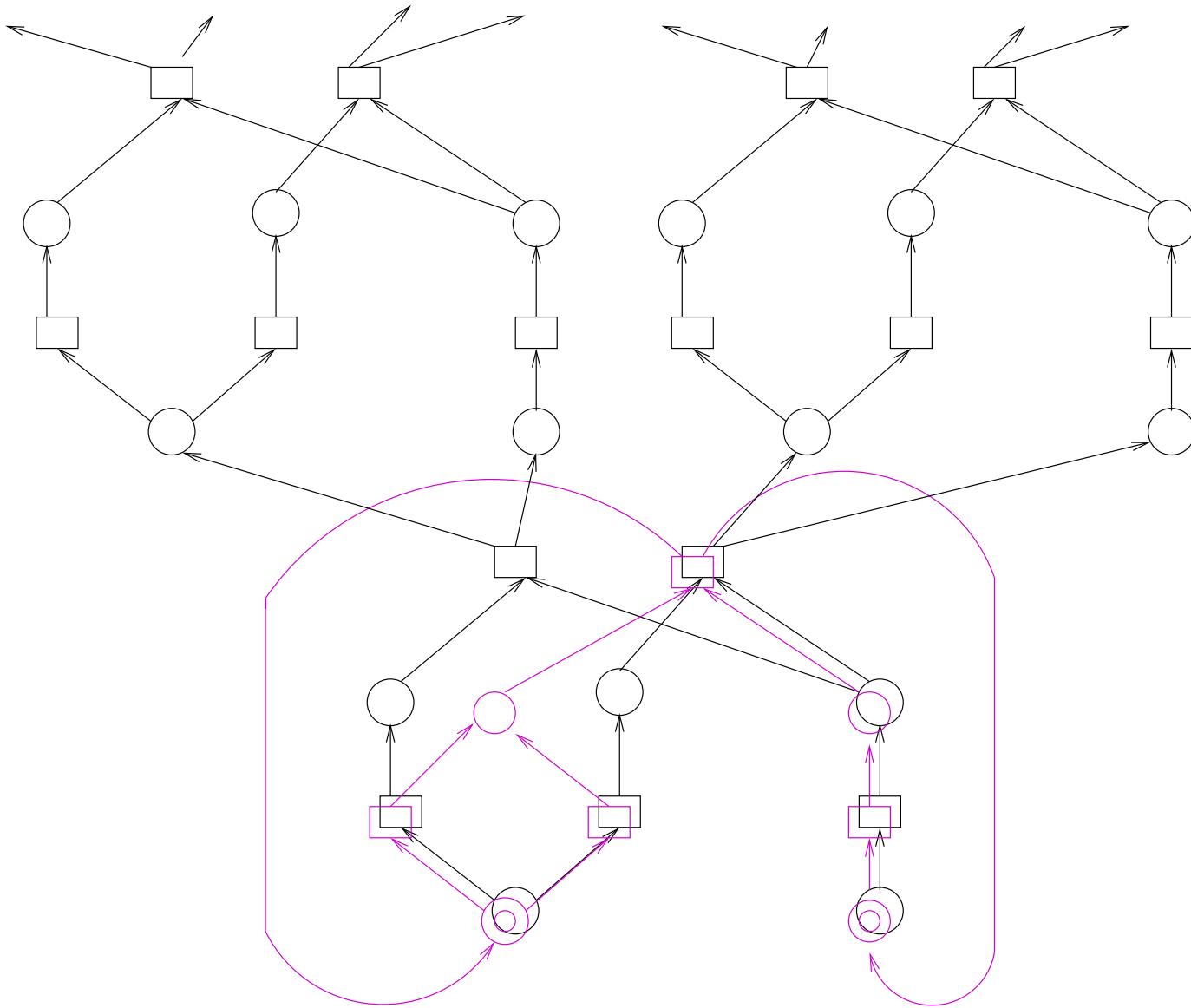


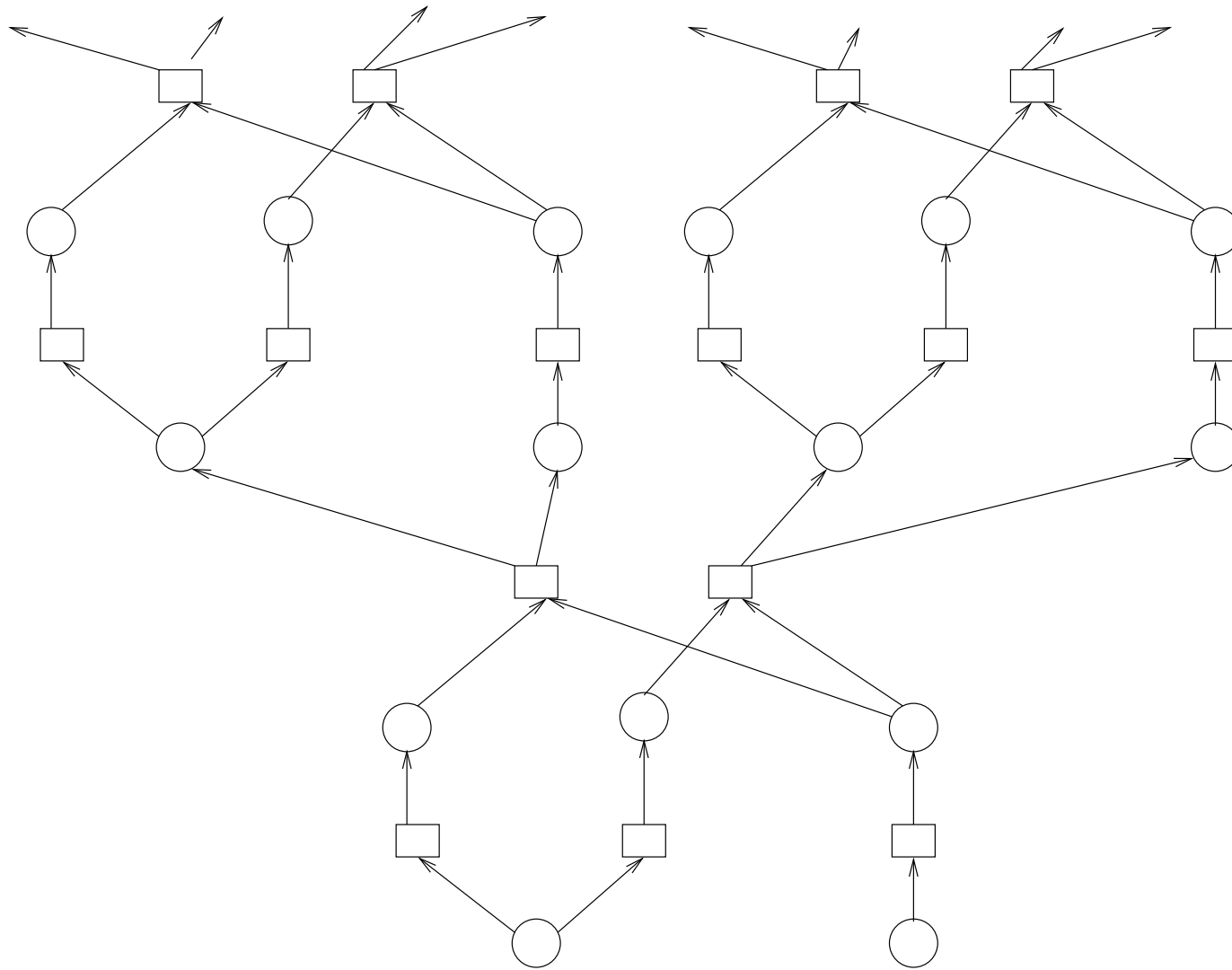


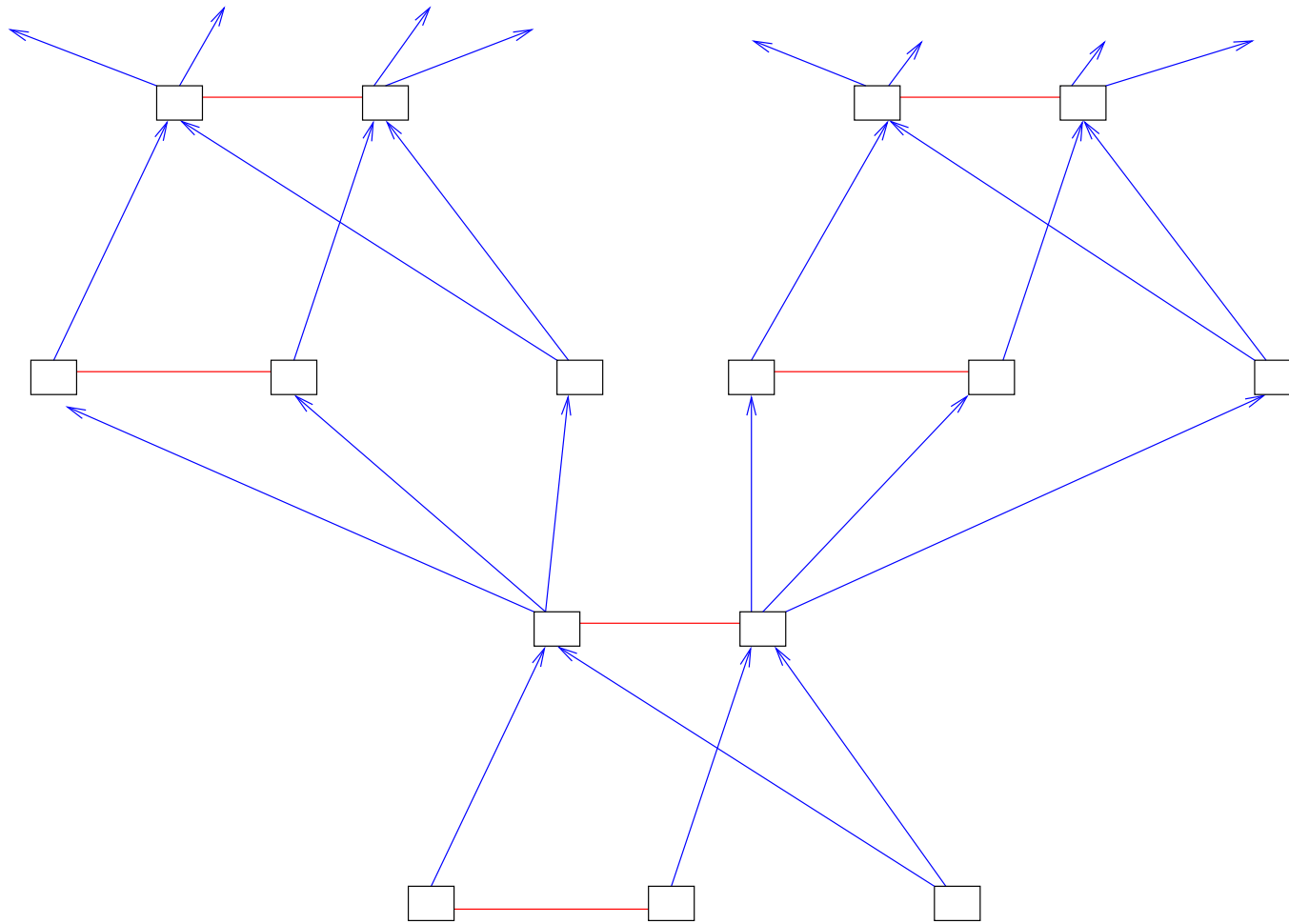












An event structure

Applications of partial-order models

Security protocols, as strand spaces [Guttman et al];

Systems biology, analysis of chemical pathways [Danos-Feret-Fontana-Krivine];

Hardware, in the design of asynchronous circuits [Yakovlev];

Types and proof, domain theory [Berry, Curien-Faggian, Girard];

Nondeterministic dataflow [Jonsson];

Network diagnostics [Benveniste et al];

Logic of programs, in concurrent separation logic;

Partial order model checking [McMillan];

Distributed computation, classically [Lamport] and recently in analysis of trust [Nielsen-Krukow-Sassone].

Domain theory and denotational semantics

Its history and limitations ...

What is a computational process?

Pre 1930's: An algorithm (*informal*)

Post 1930's: An effective partial function $f : \mathbb{N} \rightarrow \mathbb{N}$ (*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?): $t ::= x \mid \lambda x.t \mid (t t')$

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 $f : D \rightarrow E$ 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

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

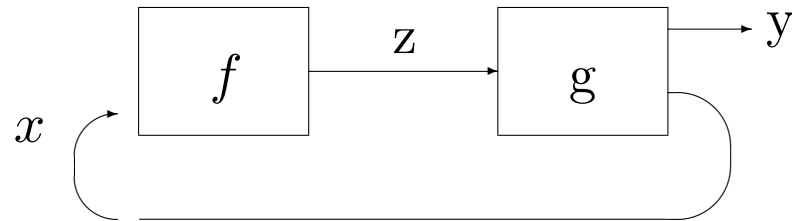
has a least upper bound $\bigsqcup_{n \in \omega} d_n$.

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

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

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

Deterministic dataflow—Kahn networks



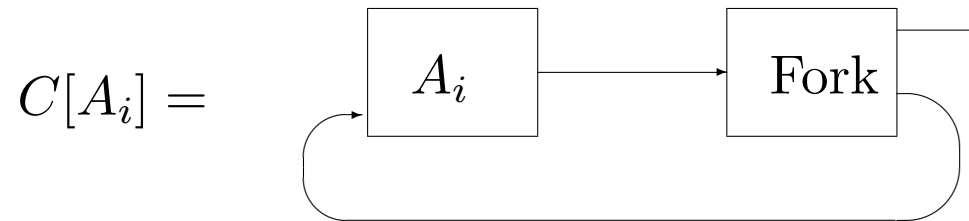
A process built from basic processes connected by channels at which they input and output.

Simple semantics: Associate channels with streams x, y, z .

Provided f and g are continuous functions on streams there is a least fixed point

$$(x, y, z) = (g(z)_2, g(z)_1, f(x)) .$$

Nondeterministic dataflow—the Brock-Ackerman anomaly



Both nondeterministic processes

$$A_1 = O + OIO \quad \text{and} \quad A_2 = O + IOO$$

have the same I/O relation, comprising

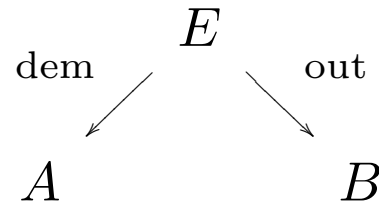
$$(\varepsilon, O), (I, O), (I, OO) .$$

But

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

A solution

A process with input A and output B :



where A , B and E are event structures,

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

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

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

Game semantics

Traditional game semantics 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 ...

Logic

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

Propositions as types, proofs as programs

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

Propositions as games, proofs as strategies

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

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

Ch 2. EVENT STRUCTURES

Representations of domains

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

(*'Topological'*) [Scott]: *Propositions* about finite properties;
more information corresponds to more propositions being true.
Functions are ordered pointwise.

Can represent domains via logical theories. ('Logic of domains')

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

Event structures

An *event structure* comprises (E, \leq, 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

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

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

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

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

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

Configurations of an event structure

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

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

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

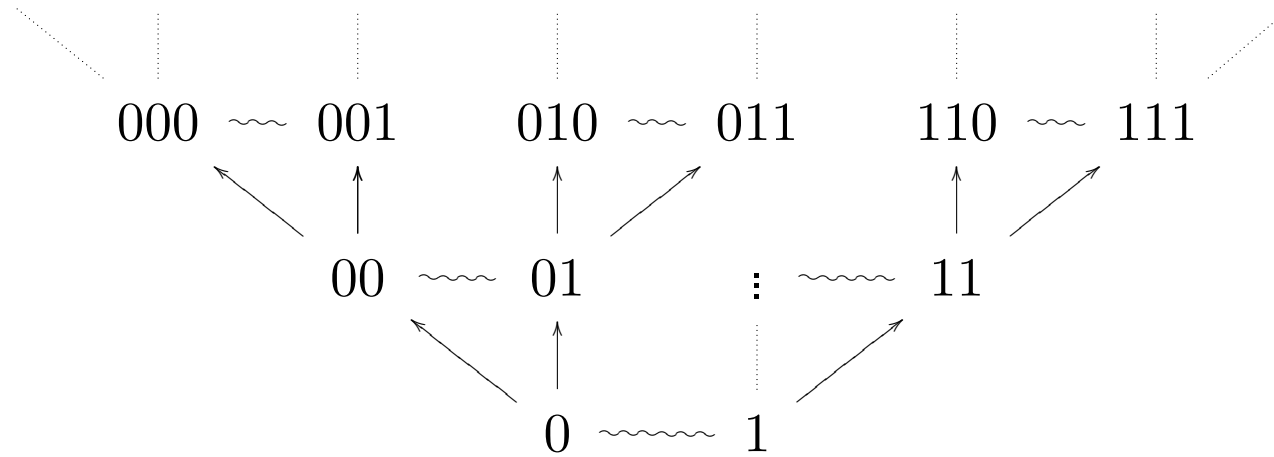
For an event e the set $[e] =_{\text{def}} \{e' \in E \mid e' \leq e\}$ is a configuration describing the whole causal history of the event e .

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

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

Finite configurations: $\mathcal{C}(E)$.

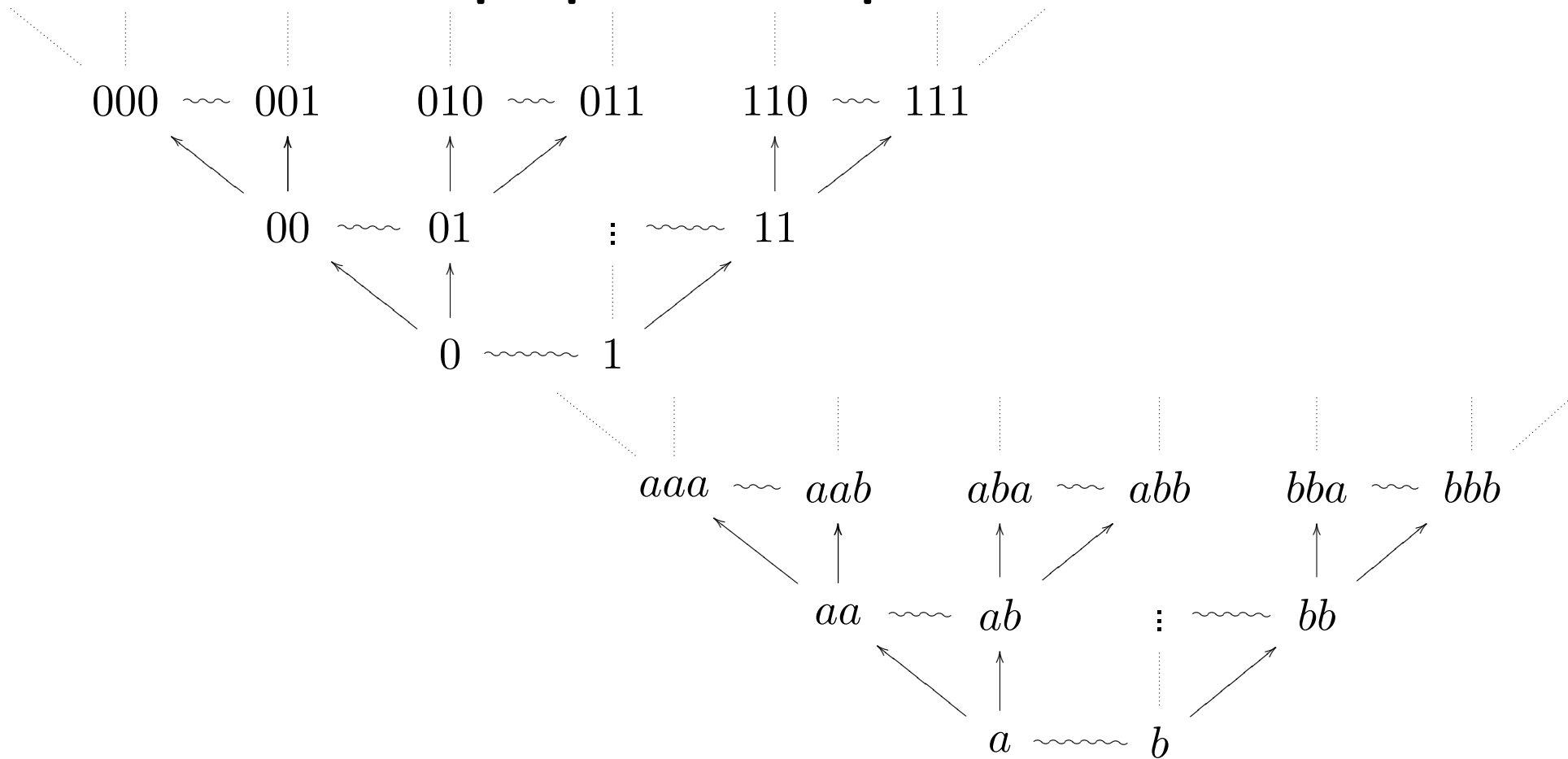
Example: Streams as event structures



~~~~ conflict (inconsistency)

→ causal dependency  $\leq$

# Simple parallel composition



## Maps of event structures

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

In this context, a *simulation map* of event structures  $f : E \rightarrow E'$  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  $e_1 = e_2$ .      ('event linearity')

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

## Process constructions on event structures

**“Partial synchronous” product:**  $A \times B$  with projections  $\Pi_1$  and  $\Pi_2$ ,  
*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.*)

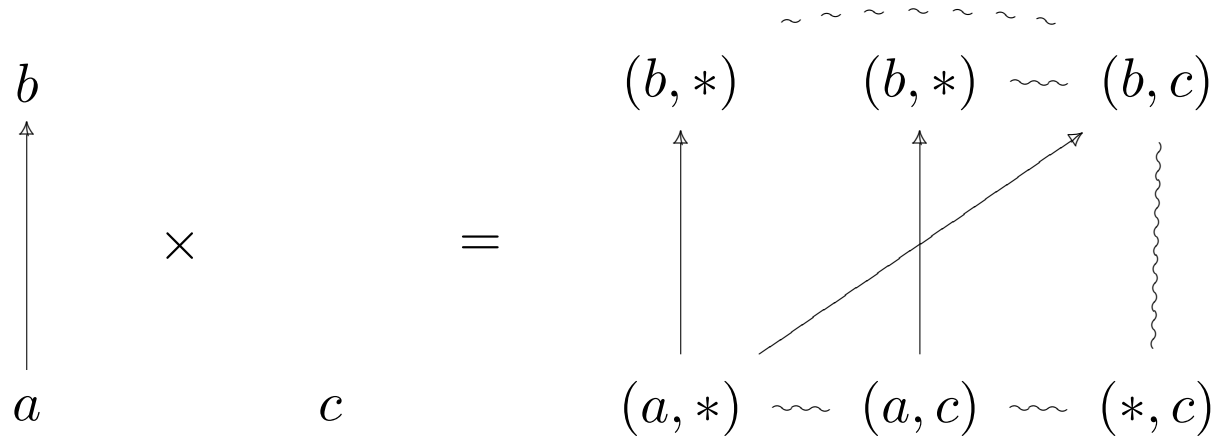
**Restriction:**  $E \upharpoonright R$ , the restriction of an event structure  $E$  to a subset of events  $R$ , has events  $E' = \{e \in E \mid [e] \subseteq R\}$  with causal dependency and consistency restricted from  $E$ .

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

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

[*Event structures as types and processes? Spans* ]

# Product—an example



## **Ch 3. STABLE FAMILIES**

## Stable families—the secret weapon

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

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

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

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

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

( $Z \uparrow$  means  $\exists x \in \mathcal{F} \forall z \in Z. z \subseteq x$ , and expresses the compatibility of  $Z$ .)

We call elements of  $\bigcup \mathcal{F}$  *events* of  $\mathcal{F}$ .

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

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

When  $e \in x$  define the prime configuration

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

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

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

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

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

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

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

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

## Categories of stable families and event structures

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

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

$\text{Pr}$  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  $E \rightarrow \text{Pr}(\mathcal{C}(E))$  takes an event  $e$  to the prime configuration  $[e] =_{\text{def}} \{e' \in E \mid e' \leq e\}$ . The counit  $\text{max} : \mathcal{C}(\text{Pr}(\mathcal{F})) \rightarrow \mathcal{F}$  takes prime configuration  $[e]_x$  to  $e$ .

## Product of stable families

Let  $\mathcal{A}$  and  $\mathcal{B}$  be stable families with events  $A$  and  $B$ , respectively. Their product, the stable family  $\mathcal{A} \times \mathcal{B}$ , has events comprising pairs in

$A \times_* B =_{\text{def}} \{(a, *) \mid a \in A\} \cup \{(a, b) \mid a \in A \ \& \ b \in B\} \cup \{(*, b) \mid b \in B\}$ ,  
the product of sets with partial functions, with (partial) projections  $\pi_1$  and  $\pi_2$ —treating  $*$  as ‘undefined’—with configurations

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

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

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

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

$(e \in y \iff e' \notin y)$ .

## Product of event structures

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

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

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

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

# Ch 4. CONCURRENT GAMES

# Concurrent games

## Basics

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

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

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

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

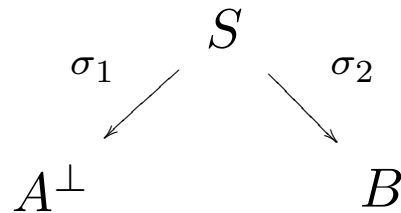
A (nondeterministic) concurrent *pre-strategy* in game  $A$  is a total map  $\sigma : S \rightarrow A$  of event structures with polarity.

## Pre-strategies

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

$$\sigma : S \rightarrow A^\perp \parallel B.$$

It determines a span of event structures with polarity



where  $\sigma_1, \sigma_2$  are *partial* maps of event structures with polarity; one and only one of  $\sigma_1, \sigma_2$  is defined on each event of  $S$ .

## Concurrent copy-cat

Identities on games  $A$  are given by copy-cat strategies  $\gamma_A : \mathbb{C}_A \rightarrow A^\perp \parallel A$  —strategies for player based on copying the latest moves made by opponent.

$\mathbb{C}_A$  has the same events, consistency and polarity as  $A^\perp \parallel A$  but with causal dependency  $\leq_{\mathbb{C}_A}$  given as the transitive closure of the relation

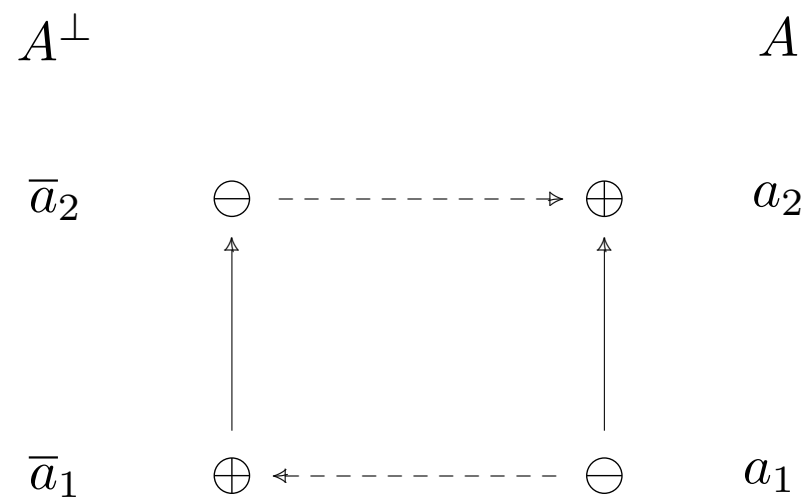
$$\leq_{A^\perp \parallel A} \cup \{(\bar{c}, c) \mid c \in A^\perp \parallel A \ \& \ pol_{A^\perp \parallel A}(c) = +\}$$

where  $\bar{c} \leftrightarrow c$  is the natural correspondence between  $A^\perp$  and  $A$ . The map  $\gamma_A$  is the identity on the common underlying set of events. Then,

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

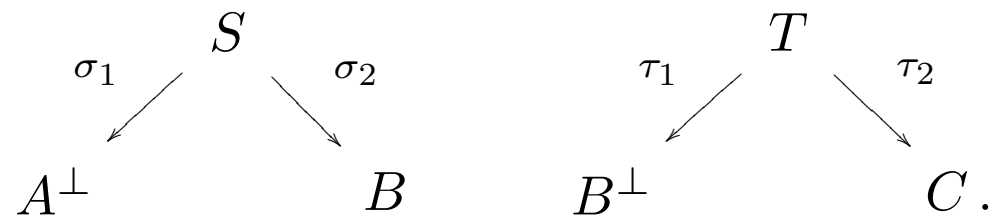
# Copy-cat—an example

$\mathbb{C}_A$

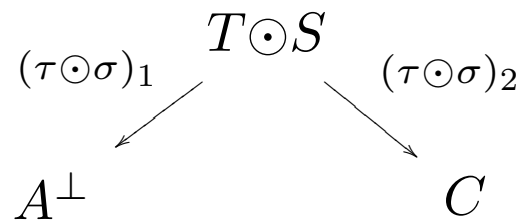


## Composing pre-strategies

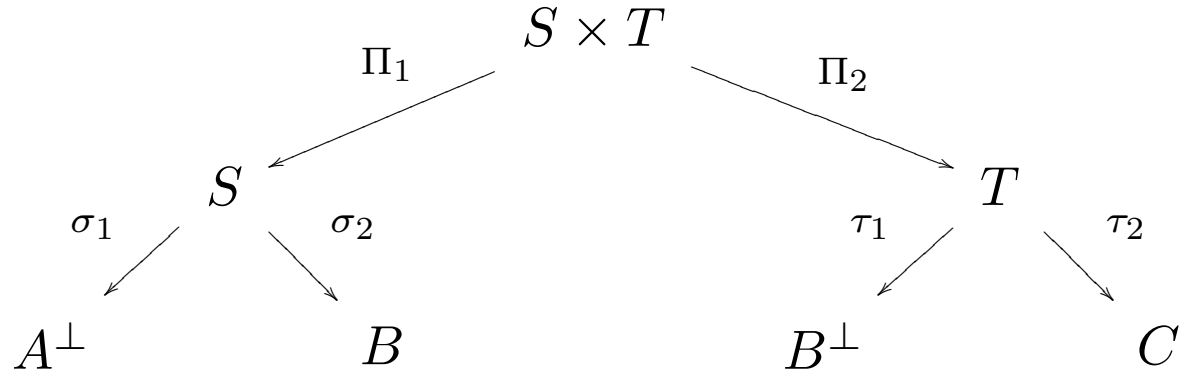
Two pre-strategies  $\sigma : A \rightarrow B$  and  $\tau : B \rightarrow C$  as spans:



Their composition



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



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

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

## Theorem characterizing concurrent strategies

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

$$\begin{array}{ccc} x & \dashv\vdash & x' \\ \downarrow & & \downarrow \\ \sigma(x) & \dashv\vdash & y \end{array}$$

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

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

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

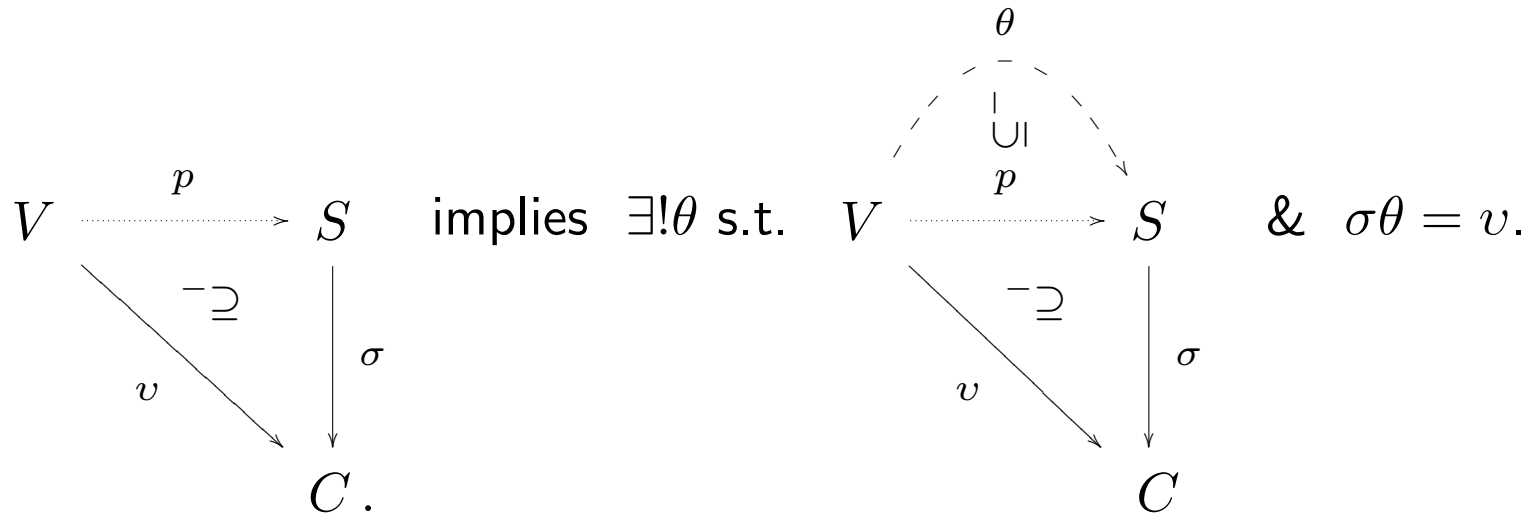
[ $\rightarrow$  stands for immediate causal dependency]

**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  $\sigma : A \rightarrow B$ .

## Idea of the proof

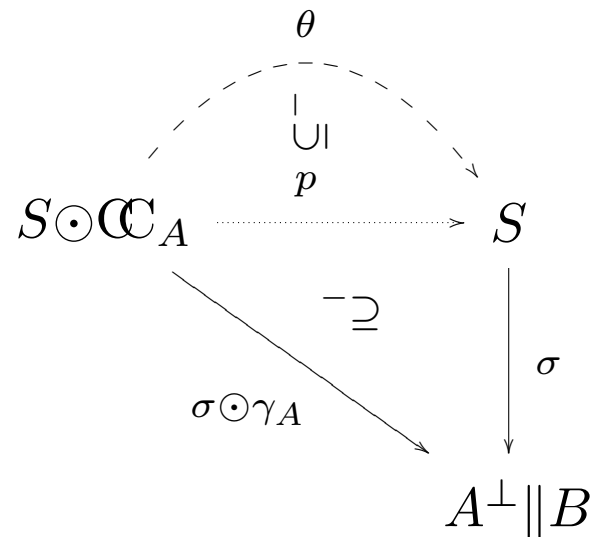
**Necessity** Copy-cats  $\gamma_A$  are receptive and innocent. If  $\sigma \cong \gamma_B \odot \sigma \odot \gamma_A$  then  $\sigma : A \rightarrow B$  inherits receptivity and innocence from that of copy-cat.

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



## Idea of the proof (cont)

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



# The bicategory of concurrent games

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

$\rightsquigarrow$  A bicategory, **Games**, whose

*objects* are event structures with polarity—the games,

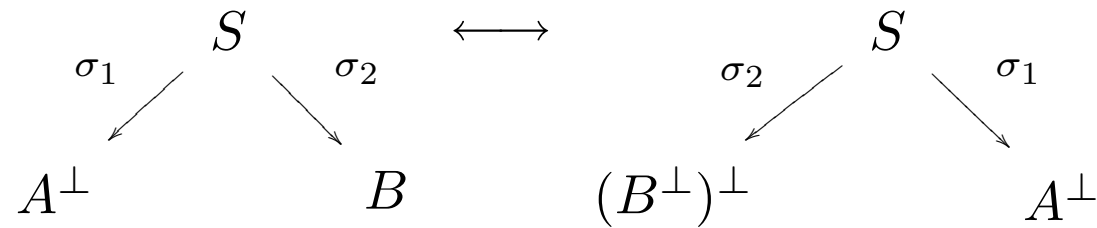
*arrows* are strategies  $\sigma : A \rightarrow B$

*2-cells* are maps of spans.

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

# Duality

A (pre-)strategy  $\sigma : A \rightarrow B$  corresponds to a dual (pre-)strategy  $\sigma^\perp : B^\perp \rightarrow A^\perp$ :



## Strategies—alternative description 1

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

(i)  $\sigma x \xrightarrow{a} \subset$  &  $\text{pol}_A(a) = - \Rightarrow \exists! s \in S. x \xrightarrow{s} \subset$  &  $\sigma(s) = a$ , for all  $x \in \mathcal{C}(S)$ ,  $a \in A$ .

(ii)(+) If  $x \xrightarrow{e} \subset x_1 \xrightarrow{e'} \subset$  &  $\text{pol}_S(e) = +$  in  $\mathcal{C}(S)$  and  $\sigma x \xrightarrow{\sigma(e')} \subset$  in  $\mathcal{C}(A)$ , then  $x \xrightarrow{e'} \subset$  in  $\mathcal{C}(S)$ .

(ii)(-) If  $x \xrightarrow{e} \subset x_1 \xrightarrow{e'} \subset$  &  $\text{pol}_S(e') = -$  in  $\mathcal{C}(S)$  and  $\sigma x \xrightarrow{\sigma(e')} \subset$  in  $\mathcal{C}(A)$ , then  $x \xrightarrow{e'} \subset$  in  $\mathcal{C}(S)$ .

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

## Strategies—alternative description 2

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

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

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.*

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

## Strategies—alternative description 3, via just +-moves

A strategy  $\sigma : S \rightarrow A$  determines  $S \xrightarrow{q} S^+$  where  $q$  is projection and

$$\begin{array}{ccc} S & \xrightarrow{q} & S^+ \\ \sigma \downarrow & \dashv\cong & \swarrow d \\ & & A \end{array}$$

$d : \mathcal{C}(S) \rightarrow \mathcal{C}(A)$  s.t.  $d(x) = \sigma[x]$ . Universal property showing  $d$  determines  $\sigma$ :

$$\begin{array}{ccc} U & \xrightarrow{g} & S^+ \\ f \downarrow & \dashv\cong & \swarrow d \\ & & A \end{array} \Rightarrow \exists! \phi \text{ s.t. } \begin{array}{ccccc} & & g & & \\ & & \curvearrowright & & \\ U & \xrightarrow{\phi} & S & \xrightarrow{q} & S^+ \\ f \downarrow & & \sigma \downarrow & \dashv\cong & \swarrow d \\ & & & & A \end{array} \quad \& \sigma\phi = f \quad \& q\phi = g.$$

# Ch 5. DETERMINISTIC STRATEGIES

## Deterministic strategies

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

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

where  $\text{Neg}[X] =_{\text{def}} \{s' \in S \mid \exists s \in X. \text{pol}_S(s') = - \ \& \ s' \leq s\}$ .

Say a strategy  $\sigma : S \rightarrow A$  is deterministic if  $S$  is deterministic.

**Proposition** An event structure with polarity  $S$  is deterministic iff  $x \xrightarrow{s} \subset \& x \xrightarrow{s'} \subset \& \text{pol}_S(s) = +$  implies  $x \cup \{s, s'\} \in \mathcal{C}(S)$ , for all  $x \in \mathcal{C}(S)$ .

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

## Nondeterministic copy-cats

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

$$\begin{array}{c} \ominus \rightarrow \oplus \\ A^\perp \quad \ominus \rightarrow \oplus \quad A \\ \oplus \leftarrow \ominus \end{array}$$

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

$$\begin{array}{c} A^\perp \quad \ominus \rightarrow \oplus \quad A \\ \oplus \leftarrow \ominus \end{array}$$

**Lemma** Let  $A$  be an event structure with polarity. The copy-cat strategy  $\gamma_A$  is deterministic iff  $A$  satisfies

$$\begin{aligned} \forall x \in \mathcal{C}(A). \quad x \xrightarrow{a} \subset \ \& \ x \xrightarrow{a'} \subset \ \& \ pol_A(a) = + \ \& \ pol_A(a') = - \\ \Rightarrow x \cup \{a, a'\} \in \mathcal{C}(A). \quad (\ddagger) \end{aligned}$$

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

**Lemma** A deterministic strategy  $\sigma : S \rightarrow A$  is injective on configurations (equivalently,  $\sigma : S \rightsquigarrow A$ ).

$\rightsquigarrow$  sub-bicategory **DGames**, 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  $\sigma : S \rightarrow A$ , iff

**reachability:**  $\emptyset \in F$  and if  $x \in F$ ,  $\emptyset \xrightarrow{a_1} x_1 \xrightarrow{a_2} \dots \xrightarrow{a_k} x_k = x$  within  $F$ ;

**determinacy:** If  $x \xrightarrow{a}$  and  $x \xrightarrow{a'}$  in  $F$  with  $pol_A(a) = +$ , then  $x \cup \{a, a'\} \in F$ ;

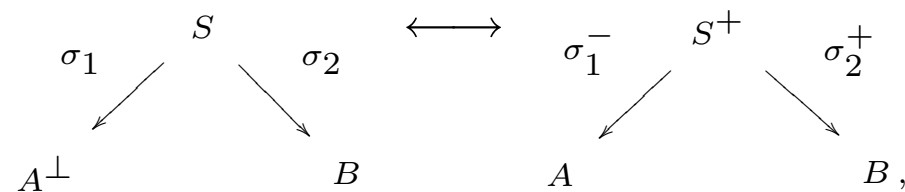
**receptivity:** If  $x \in F$  and  $x \xrightarrow{a}$  in  $\mathcal{C}(A)$  and  $pol_A(a) = -$ , then  $x \cup \{a\} \in F$ ;

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

**1-stable:** If  $x_1 \xrightarrow{a} x$  and  $x_2 \xrightarrow{b} x$  in  $F$ , then  $x_1 \cap x_2 \in F$ .

## **Ch 6. Games people play**

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



where  $S^+$  is the projection of  $S$  to its +ve events;  $\sigma_2^+$  is the restriction of  $\sigma_2$  to  $S^+$  is rigid;  $\sigma_2^-$  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.

When deterministic (and event structures are countable) we obtain a sub-bicategory equivalent to Berry's **dl-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  $\sigma : S \rightarrow A$  determines a closure operator  $\varphi$  on  $\mathcal{C}^\infty(S)$ : for  $x \in \mathcal{C}^\infty(S)$ ,

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

The closure operator  $\varphi$  on  $\mathcal{C}^\infty(S)$  induces a *partial* closure operator  $\varphi_p$  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 **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 5. 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  $L =_{\text{def}} \mathcal{C}^\infty(A) \setminus W$ .

## Winning strategies

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

A strategy in  $G$  is a strategy in  $A$ .

A strategy  $\sigma : S \rightarrow A$  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  $x \xrightarrow{s} \subset$  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

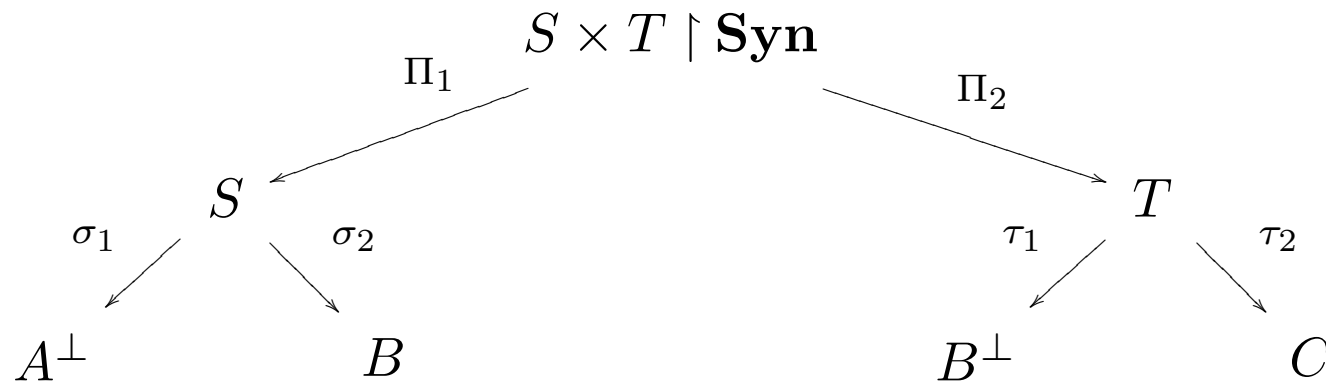
*Informally, a strategy is winning for Player if any play against a counter-strategy of Opponent results in a win for Player.*

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

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

Note  $\sigma : \emptyset \rightarrow A$  and  $\tau : A \rightarrow \emptyset \dots$

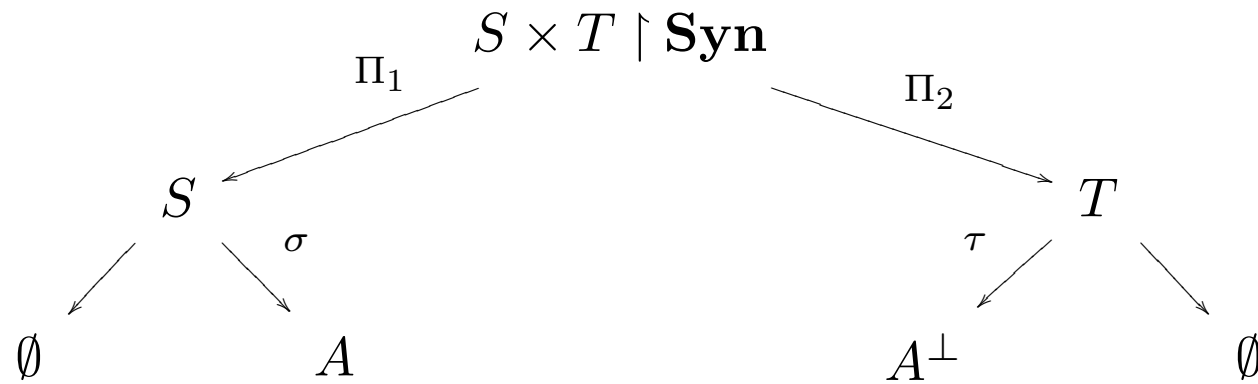
## Composition of pre-strategies without hiding



where

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

## Special case



where

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

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

## Characterization of winning strategies

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

Its proof uses a key lemma:

**Lemma 61 (restated)** Let  $\sigma : S \rightarrow A^\perp \parallel B$  and  $\tau : B^\perp \parallel C$  be receptive pre-strategies. Then,

$z \in \mathcal{C}^\infty(S \times T \upharpoonright \mathbf{Syn})$  is +-maximal iff

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

## Operations on games with winning conditions

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

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

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

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

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

## Derived operations

**Tensor** Defining  $G \otimes H =_{\text{def}} (G^\perp \parallel H^\perp)^\perp$  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,

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

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

**Function space** With  $G \multimap H =_{\text{def}} G^\perp \parallel H$  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  $G \multimap H$ , so  $G \multimap H = (A^\perp \parallel B, W_{G \multimap H})$ . For  $x \in \mathcal{C}^\infty(A^\perp \parallel B)$ ,

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

## The bicategory of winning strategies

**Lemma** Let  $\sigma$  be a winning strategy in  $G^\perp \parallel H$  and  $\tau$  be a winning strategy in  $H^\perp \parallel K$ . Their composition  $\tau \odot \sigma$  is a winning strategy in  $G^\perp \parallel K$ .

But copy-cat need not be winning: Let  $A$  consist of two events, one +ve event  $\oplus$  and one -ve event  $\ominus$ , inconsistent with each other. Take as winning conditions the set  $\{\{\oplus\}\}$ . The event structure  $\mathbb{C}_A$ :

$$\begin{array}{c} A^\perp \quad \ominus \rightarrow \oplus \quad A \\ \oplus \leftarrow \ominus \end{array}$$

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

A robust sufficient condition for copy-cat to be winning: copy-cat is deterministic. The notes give a necessary and sufficient condition.

$\rightsquigarrow$  bicategory of games with winning strategies.

## Two applications

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

**Determinacy of concurrent games:** A necessary condition on a game  $A$  for  $(A, W)$  to be determined for all winning conditions: that copy-cat  $\gamma_A$  is deterministic. Is this sufficient to ensure determinacy w.r.t. all Borel winning conditions?

There must be many more!