i 0

Welcome!

EPIT32 — April 26-30, 2004 — CIRM

Introduction to models for concurrency

Rémi Morin

May 19, 2004

Overview

① Semantics of 1-safe Petri nets

$\leadsto$ Javier Esparza's lecture

② Theory of Mazurkiewicz traces

$\leadsto$ Benoît Caillaud's lecture

③ Introduction to message sequence charts

$\leadsto$ Anca Muscholl's lecture

④ Some generalizations to dynamic traces

Keywords

① Relationships between models

② Labelled partial orders (pomsets)

③ Refinements of sets of pomsets

You'll find these slides at the EPIT32 web site

EPIT32 — April 26-30, 2004 — CIRM

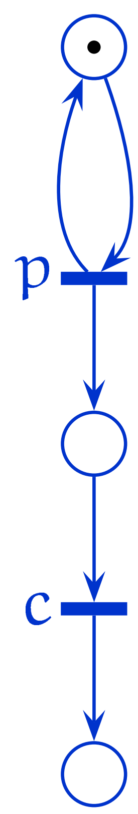
Semantics of 1-safe Petri nets

Rémi Morin

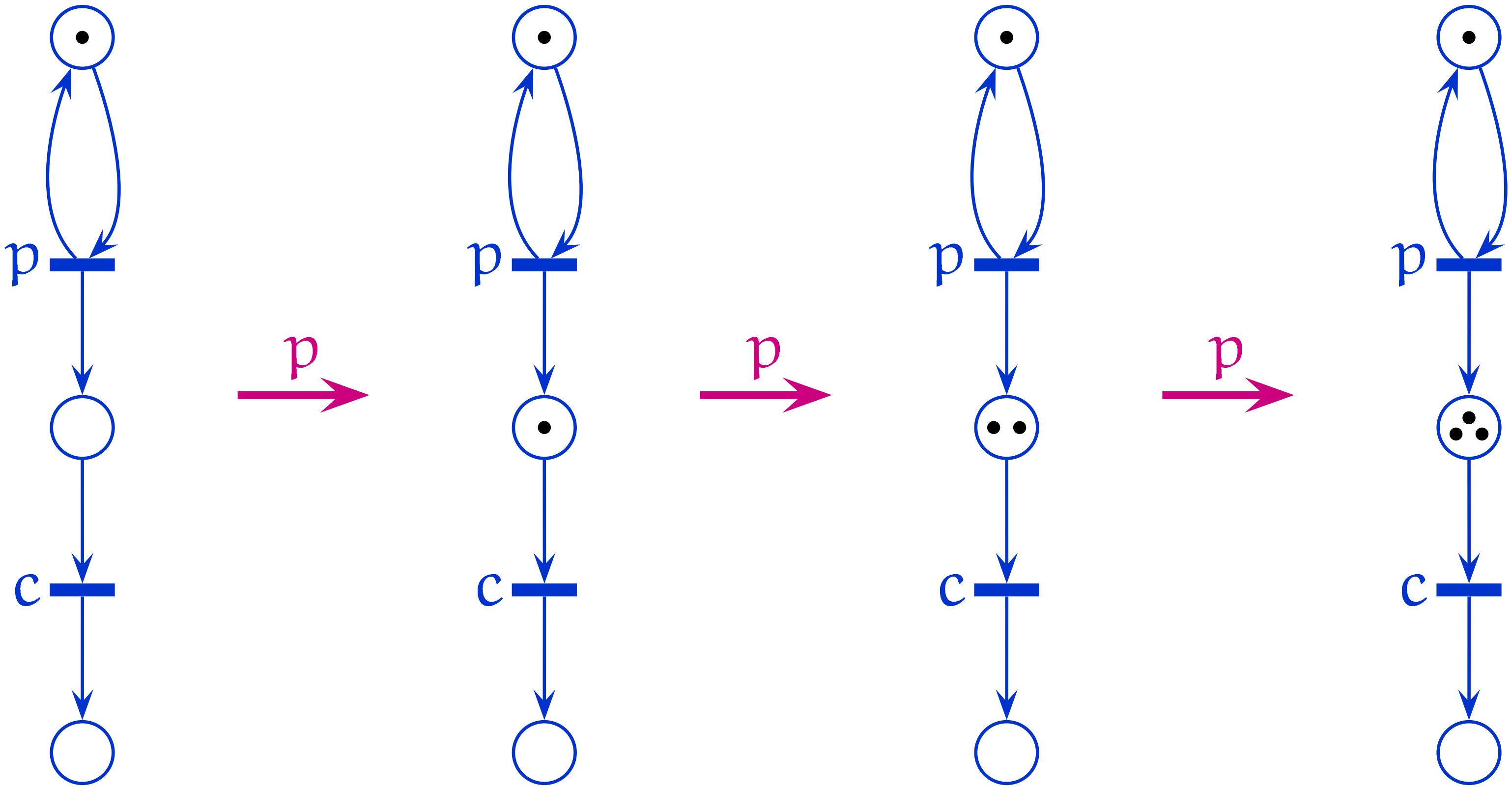
May 19, 2004

Some examples

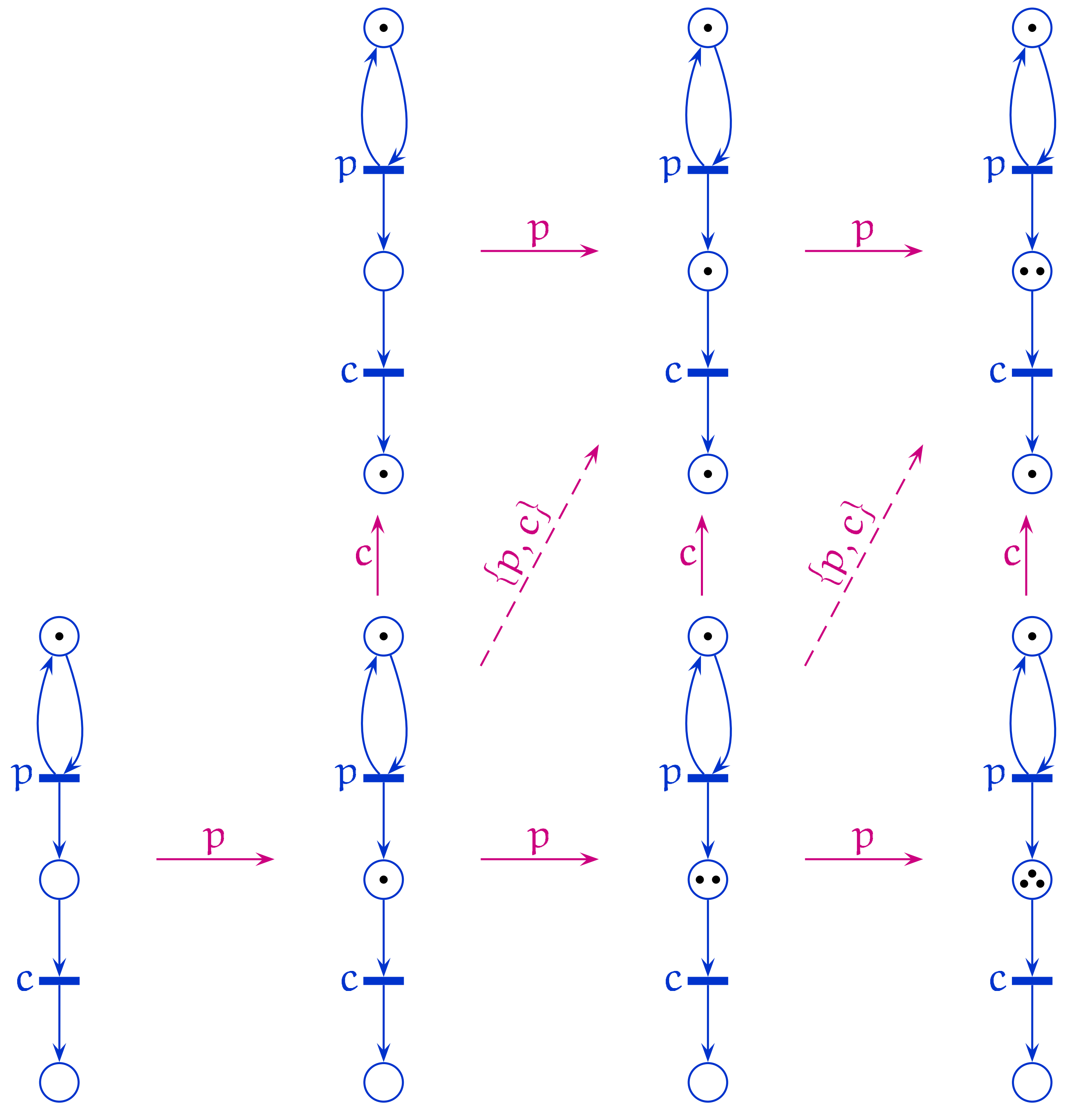
Our running example: The Producer-Consumer system



Token game of the Producer-Consumer system



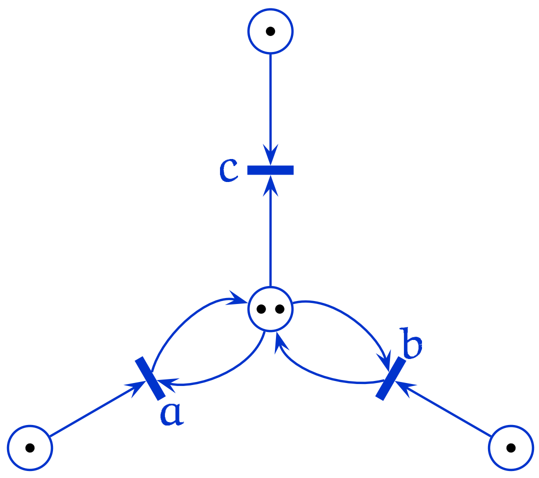
Step transitions of the Producer-Consumer system



$(p, \{p, c\}) \in I$

$(\varepsilon, \{p, c\}) \not\in I$

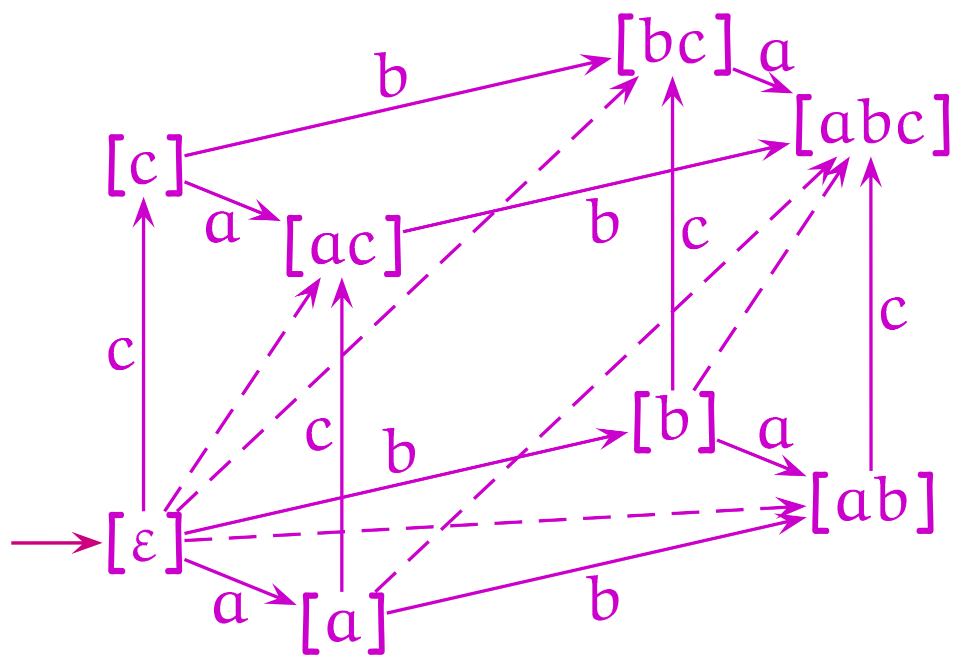
Dynamic and higher-dimensional independence relations



$(\varepsilon, \{a, b\}) \in I$

$(c, \{a, b\}) \notin I$ **True concurrency**

$(\varepsilon, \{a, b, c\}) \notin I$



Petri nets and their token game

A Petri net is a quadruple $\mathcal{N} = (S, T, W, m_{in})$ where

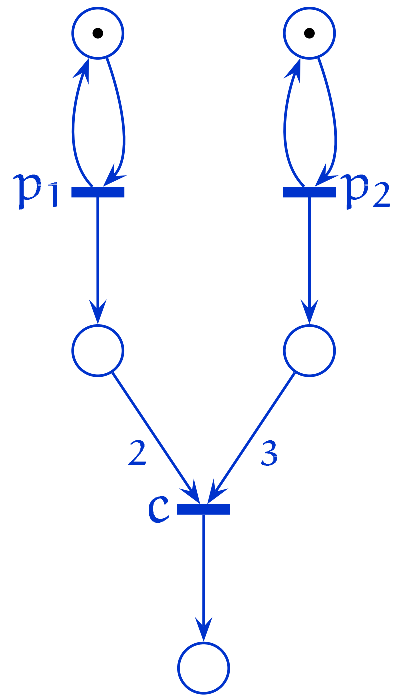
- S is a set of places and T is a set of transitions such that $S \cap T = \emptyset$;
- W is a map from $(S \times T) \cup (T \times S)$ to $\mathbb{N}$, called weight function;
- $m_{in} \in \mathbb{N}^S$ is an initial marking.

$\mathsf{Mar}_{\mathcal{N}} = \mathbb{N}^S$ is the set of all markings of $\mathcal{N}$.
A transition $t \in T$ is enabled at $m \in \mathsf{Mar}_{\mathcal{N}}$ if

$$m(s) \geqslant W(s,t) \text{ for all } s \in S.$$

In this case, we write $m\,[t\rangle\, m'$ where

$$m'(s) = m(s) + W(t,s) - W(s,t).$$



- ✔ *Some examples*
- ☛ *Reachability problems*

Reachable states

A marking $\mathfrak{m}$ is called reachable if there is a firing sequence

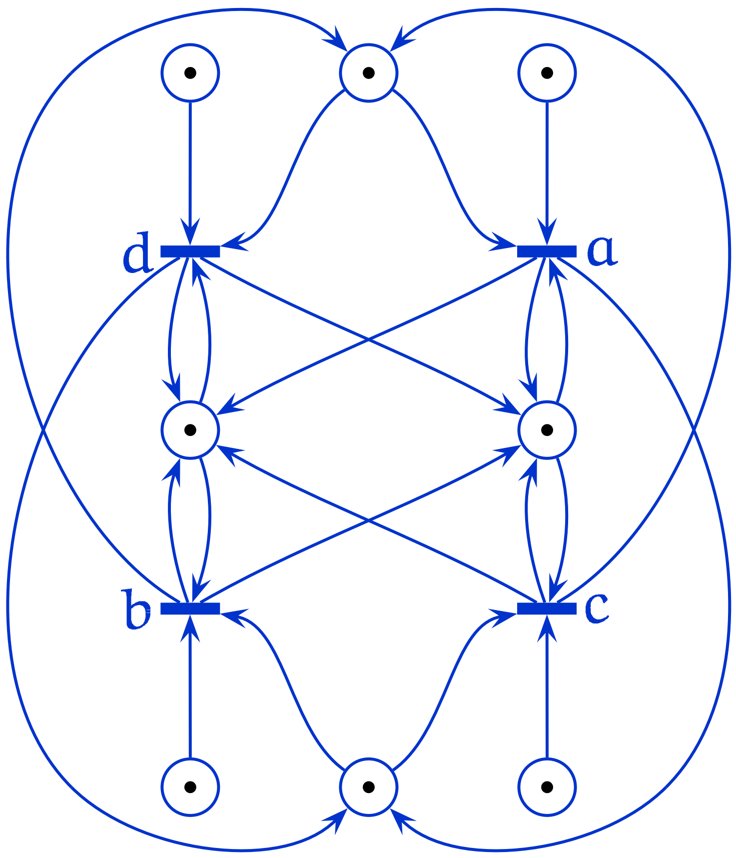
$$\mathfrak{m}_{in} = \mathfrak{m}_0 \, [t_1\rangle \, \mathfrak{m}_1 \, [t_2\rangle \, \ldots [t_n\rangle \, \mathfrak{m}_n = \mathfrak{m}$$

Reachability problem
INPUT: A (finite) Petri net $\mathcal{N}$ and a marking $\mathfrak{m}_{fin}$.
QUESTION: Is $\mathfrak{m}_{fin}$ reachable in $\mathcal{N}$?

It is known that **Reachability** is decidable [Mayr81] and requires exponential space [Lipton76]. To our knowledge, it is not known so far whether it is primitive recursive [Jancar00].

1-safe nets

A Petri net is 1-safe if each place contains at most 1 token in each reachable marking.

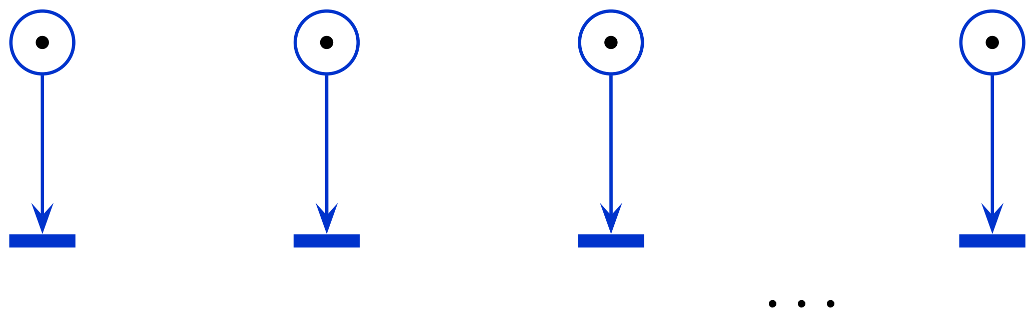


Is this net 1-safe? Is this net k-safe for some k?

You'll find probably in Volker Diekert's talk

- Karp & Miller's covering graph algorithm
- Elementary results on graph theory (Euler's theorem)
- Systems of linear equations (Presburger's arithmetics)

State explosion for 1-safe nets

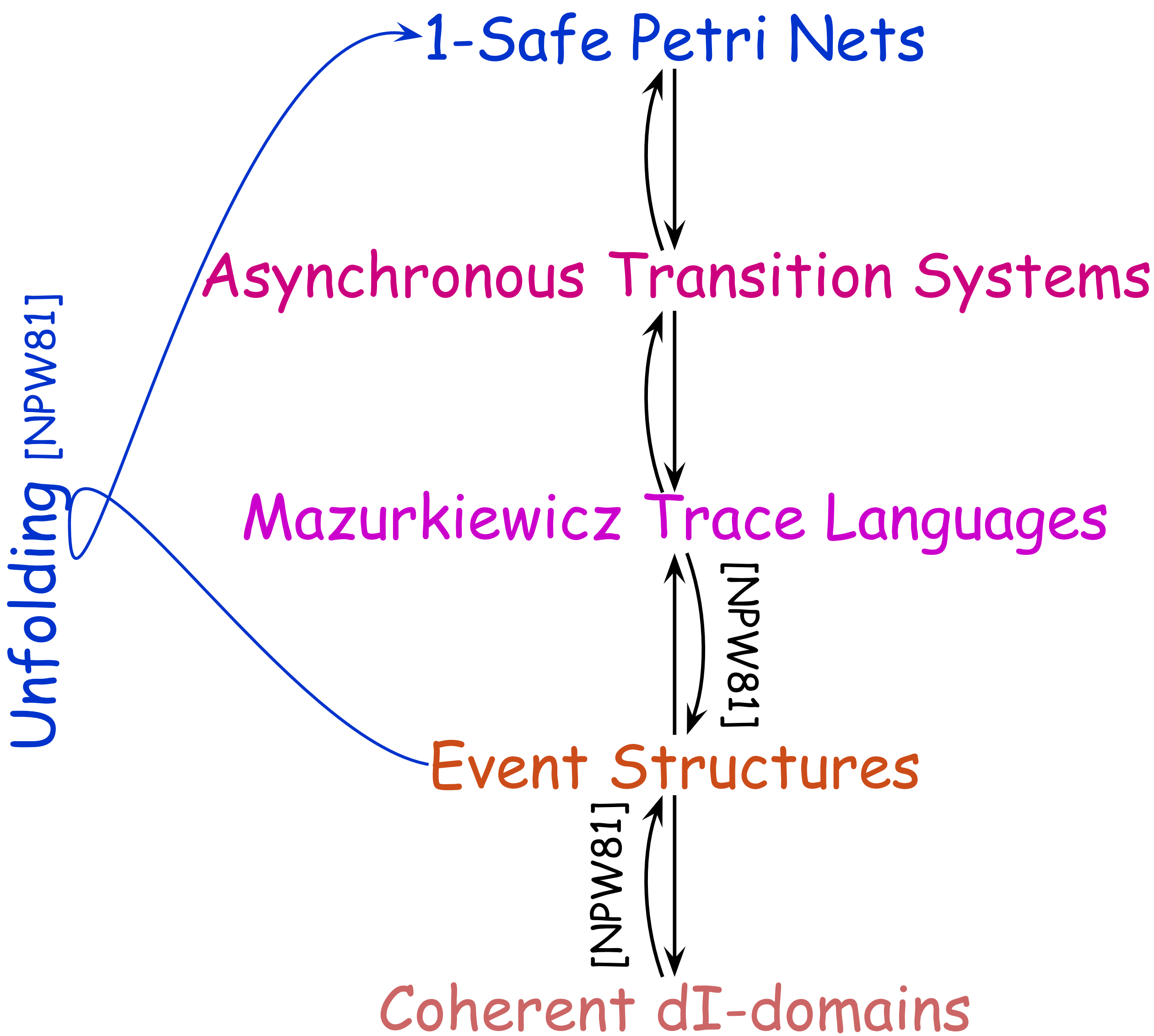


n places but 2^n reachables markings

You may not find in Javier Esparza's talk

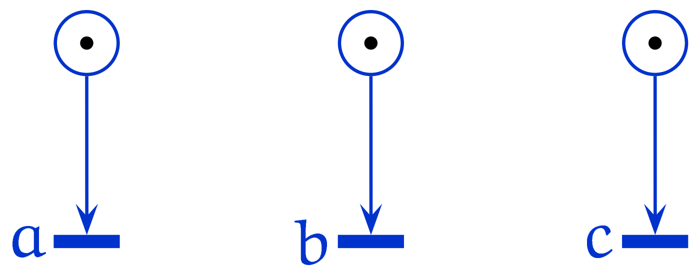
- Asynchronous transition systems [Bednarczyk87]
- Mazurkiewicz traces [Mazurkiewicz78, Diekert+Rozenberg96]
- Event structures [Nielsen+Plotkin+Winskel81]

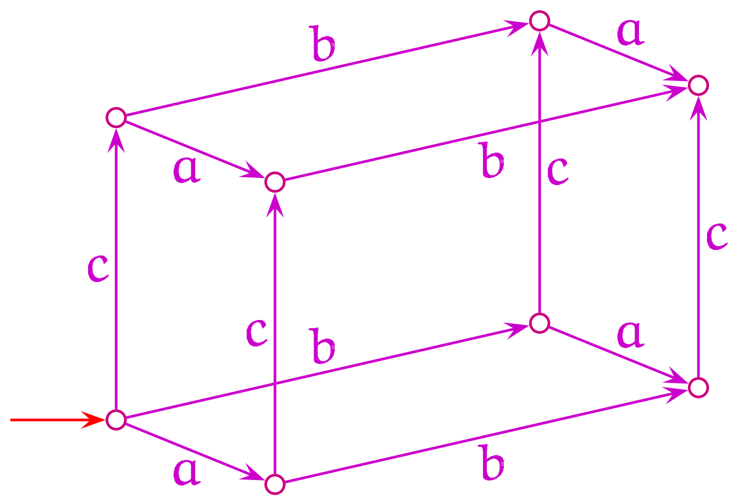
From 1-safe nets to coherent dI-domains



- ✔ *Some examples*
- ✔ *Reachability problems*
- ☛ *Trace semantics of 1-safe nets*

Marking graph of a 1-safe Petri net





Transitions a and b can fire concurrently: We put $a \| b$.
Similarly $a \| c$ and $b \| c$.
Formally, $t_1 \not\| t_2$ if there is some s with $W(s, t_1) \geqslant 1$ and $W(s, t_2) \geqslant 1$.

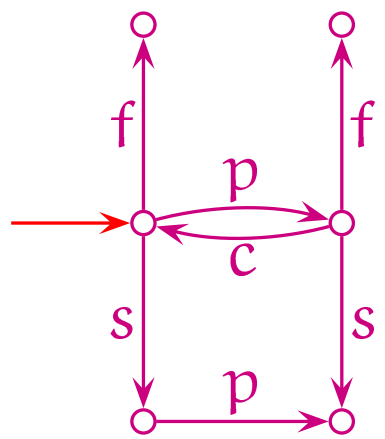
[Bednarczyk]'s asynchronous transition systems

An independence relation over Σ is a binary, symmetric and irreflexive relation $\| \subseteq \Sigma \times \Sigma$. An asynchronous transition system over $(\Sigma, \|)$ is a transition system $\mathcal{A} = (Q, \imath, \Sigma, \rightarrow)$ such that

D: $q \xrightarrow{a} q_1 \wedge q \xrightarrow{a} q_2 \Rightarrow q_1 = q_2$;

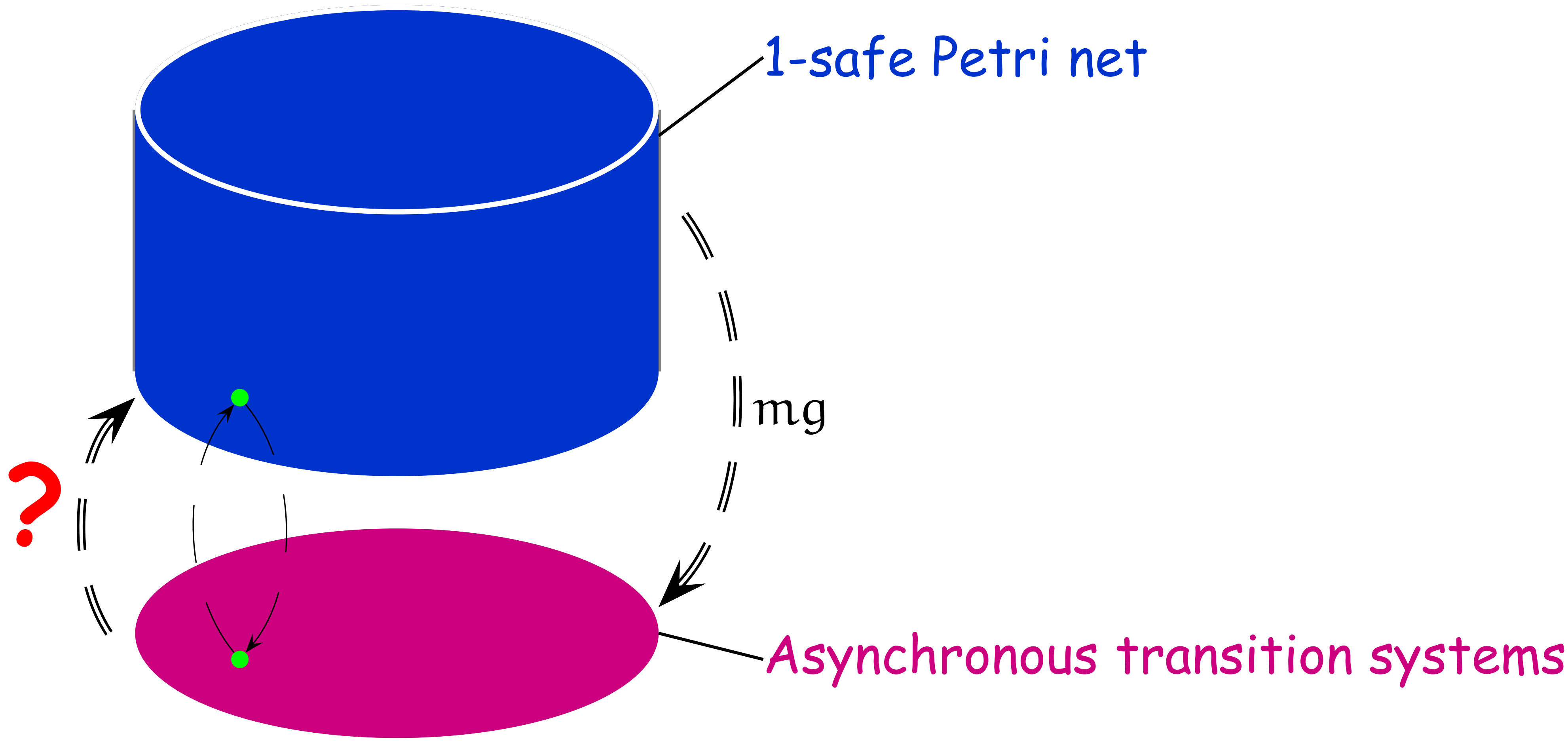
FD: $q \xrightarrow{a} q_1 \wedge q \xrightarrow{b} q_2 \wedge a \| b \Rightarrow \exists q_3 \in Q, q_2 \xrightarrow{a} q_3 \wedge q_1 \xrightarrow{b} q_3$;

ID: $q \xrightarrow{a} q_1 \wedge q_1 \xrightarrow{b} q_2 \wedge a \| b \Rightarrow \exists q_3 \in Q, q_1 \xrightarrow{b} q_3 \wedge q_3 \xrightarrow{a} q_2$.



$s \| p$

From Petri nets to asynchronous transition systems (and back)



[Ehrenfeucht+Rozenberg92,Reisig97,Badouel+Darondeau00]
Theories of regions characterize which ATS are marking graphs.

Mazurkiewicz traces

The trace equivalence associated to $(\Sigma, \|)$ is the least equivalence relation $\sim$ over $\Sigma^\star$ such that

$$\forall u, v \in \Sigma^\star, \forall a, b \in \Sigma : a \| b \Rightarrow u.ab.v \sim u.ba.v.$$

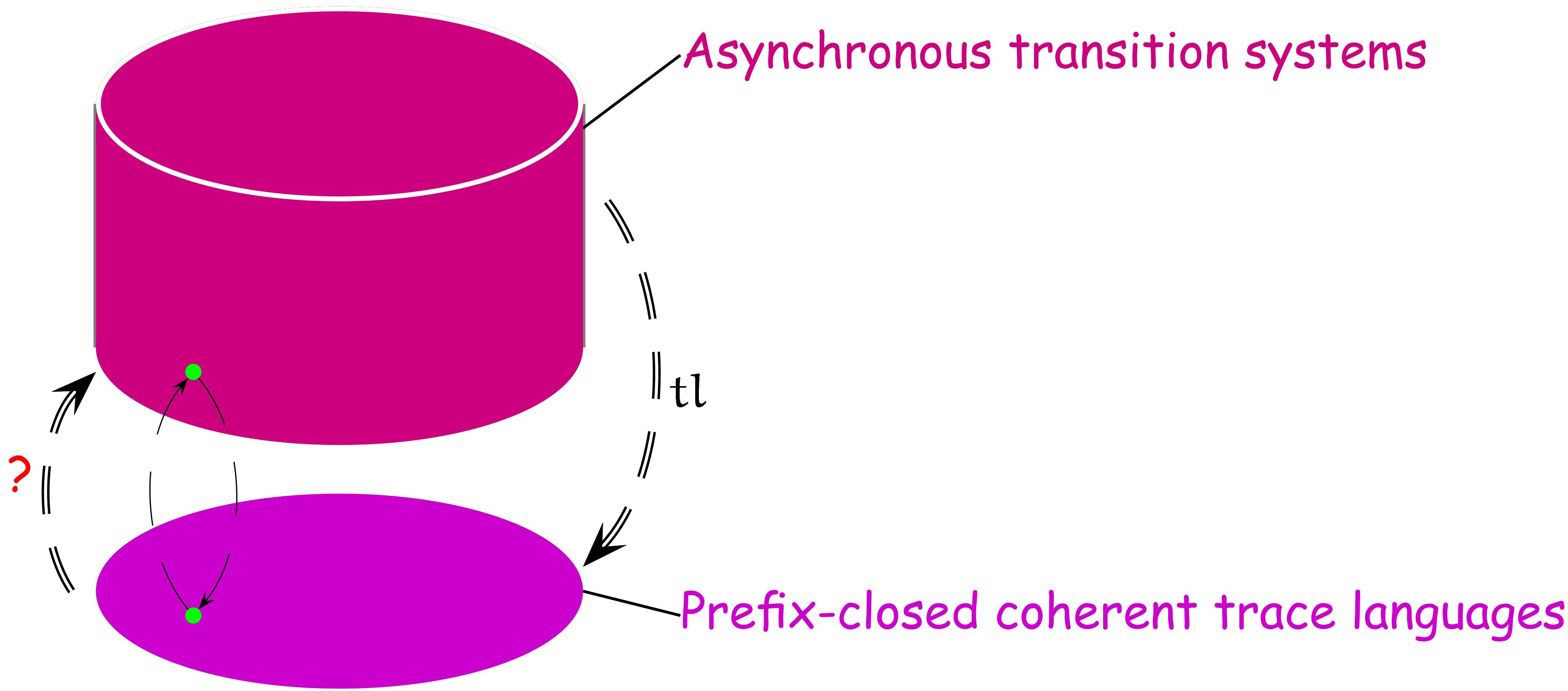
A trace $[u]$ is the equivalence class of a word $u \in \Sigma^\star$.

$\mathbb{M}(\Sigma, \|)$ denotes the set of all traces w.r.t. $(\Sigma, \|)$.

A trace language $\mathcal{L} \subseteq \mathbb{M}(\Sigma, \|)$ is identified to a subset $L \subseteq \Sigma^\star$ such that $u \sim v \wedge u \in L \Rightarrow v \in L$.
A trace language is

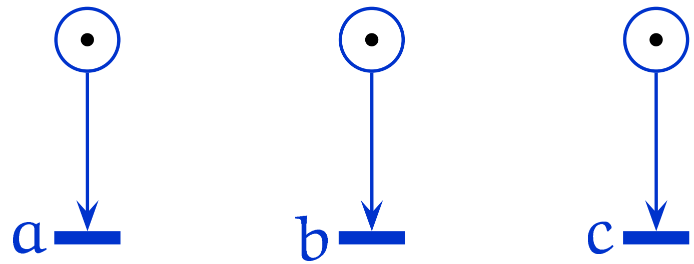
- prefix-closed if $u \leqslant v \wedge v \in L \Rightarrow u \in L$.
- coherent if $u.a \in L \wedge u.b \in L \wedge a \| b \Rightarrow u.ab \in L$.
- regular if it is accepted by a finite asynchronous transition system.

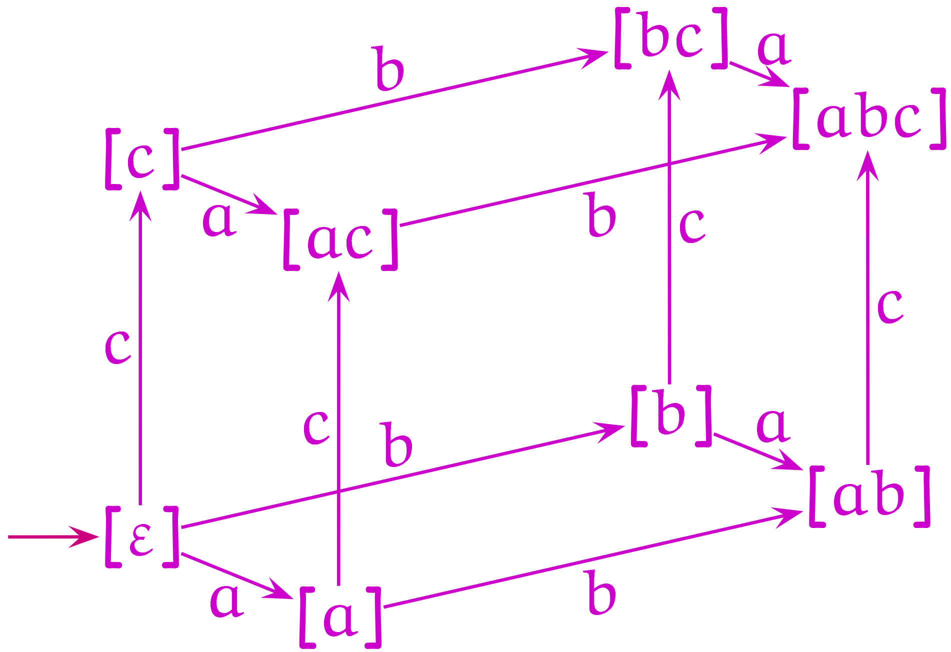
From asynchronous transition systems to trace languages (and back)



Holds with finite ATSs and regular TLs, too.

How to visit each reachable state in the marking graph?





Basic ideas to compute all reachable markings.

- Each trace maps to a reachable marking
- Build at least one representative trace for each reachable state
- Avoid considering equivalent sequential views for a trace

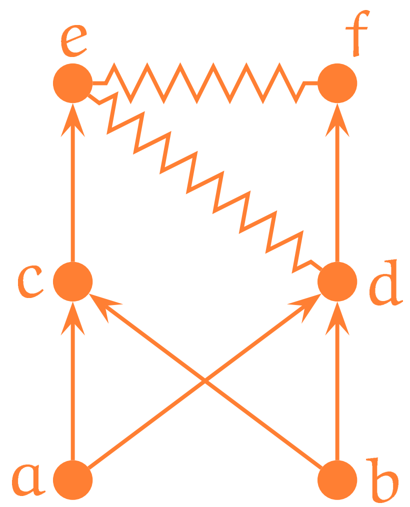
- ✔ *Some examples*
- ✔ *Reachability problems*
- ✔ *Trace semantics of 1-safe nets*
- ☛ *Event structure semantics*

[Winskel]'s prime event structures with binary conflict

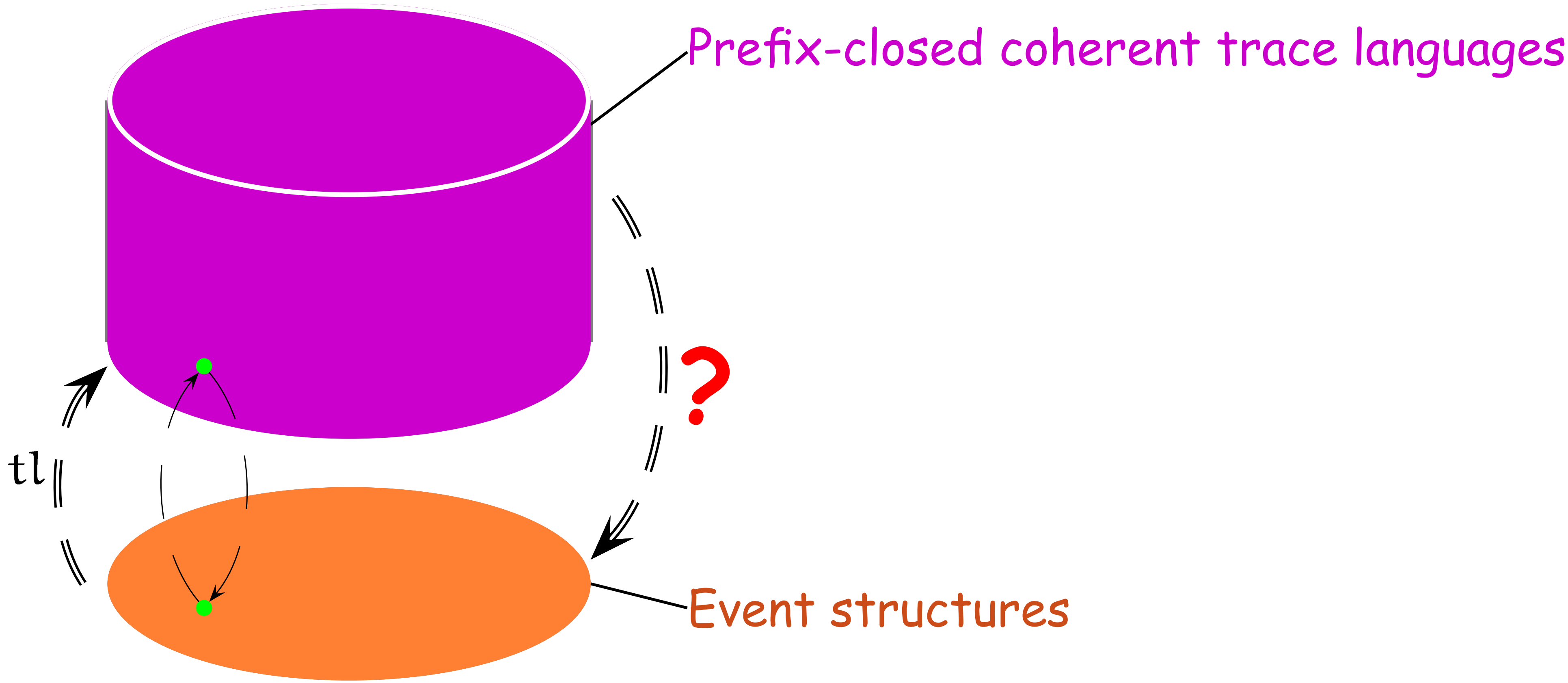
An event structure is a triple $\mathcal{E} = (E, \preccurlyeq, \sharp)$ where E is a set of events, $\preccurlyeq$ is a partial order over E and $\sharp$ is a symmetric and irreflexive relation over E such that

ES_1: for any event e, $\{f \mid f \preccurlyeq e\}$ is finite;

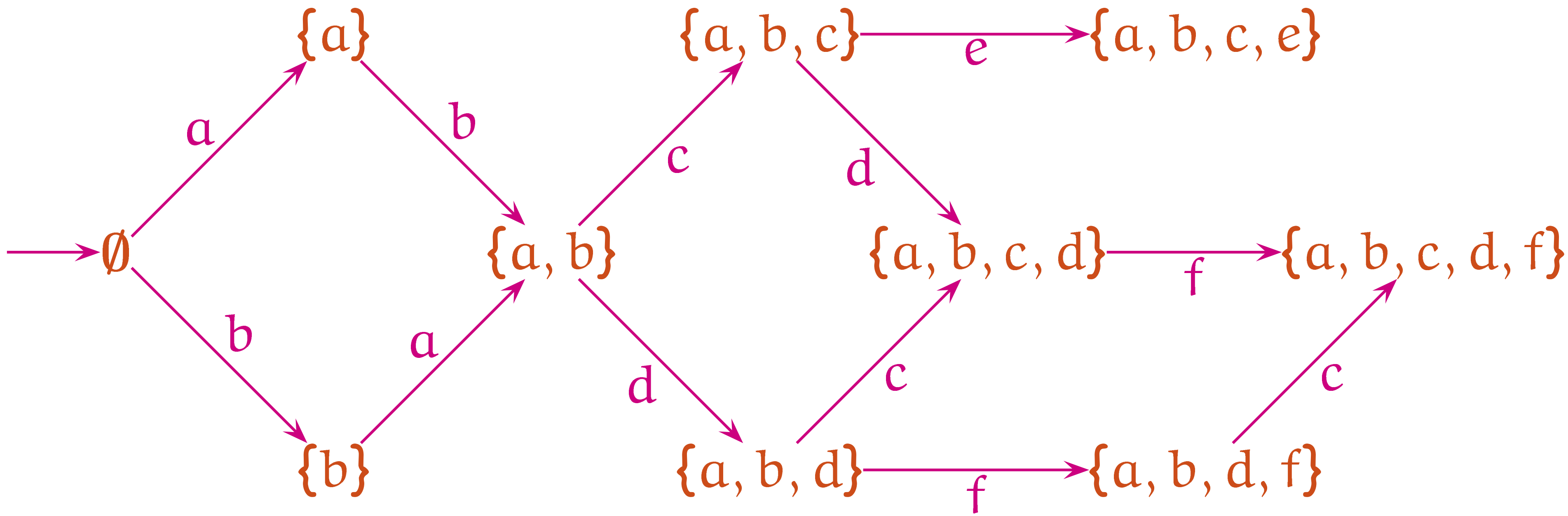
ES_2: for any events e, f and g: $(e \sharp f \wedge f \preccurlyeq g) \Rightarrow e \sharp g$.

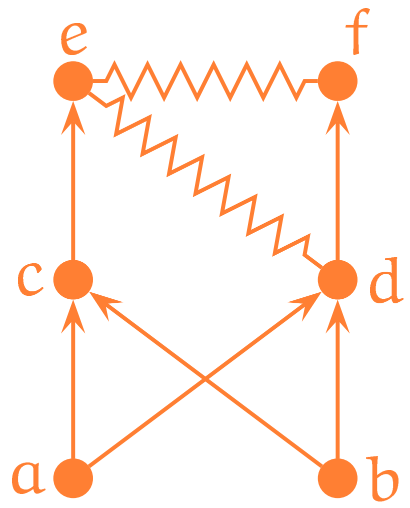


From trace languages to event structures (and back)

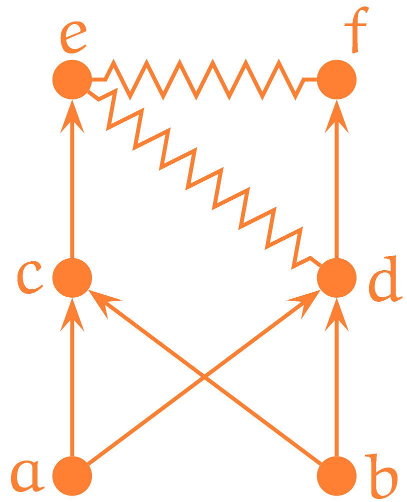


From event structures to trace languages: Example





We put e co f if neither $e \preccurlyeq f$, nor $f \preccurlyeq e$, nor $e \sharp f$.



A configuration of $\mathcal{E}$ is a set of events which is downward-closed and conflict-free.

A path is a linear extension of a configuration, i.e. it is a sequence $e_1 \ldots e_n$ of events such that

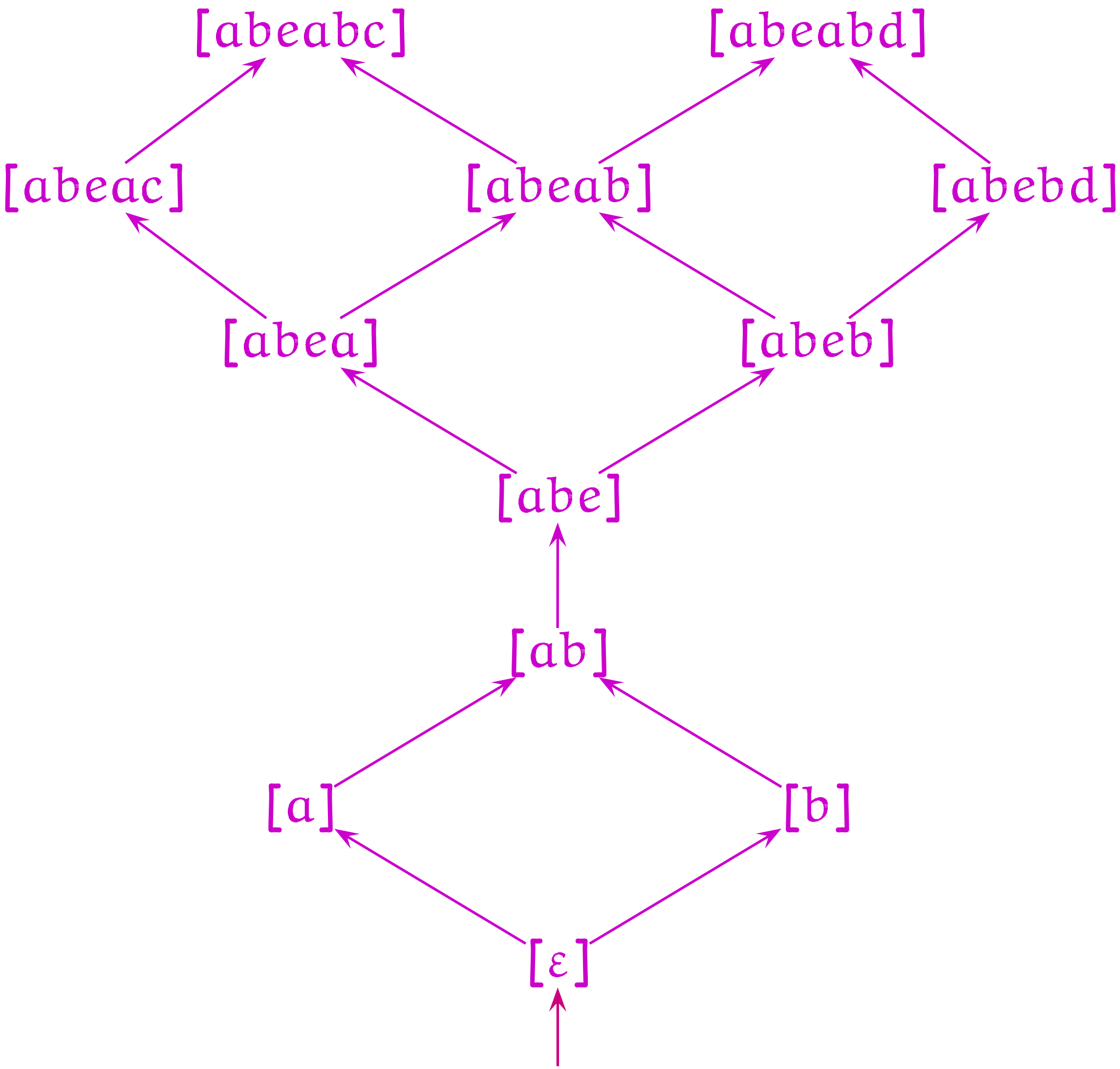
- $\forall 1 \leqslant i \leqslant n, f \preccurlyeq e_i \Rightarrow \exists j \leqslant i, f = e_j$.
- $\forall 1 \leqslant i < j \leqslant n, e_i \neq e_j \wedge \neg(e_i \sharp e_j)$.

Definition

The trace language of $\mathcal{E}$ is $\mathsf{Paths}(\mathcal{E})$ over (E, co).

From trace languages to event structures: An example (1/3)

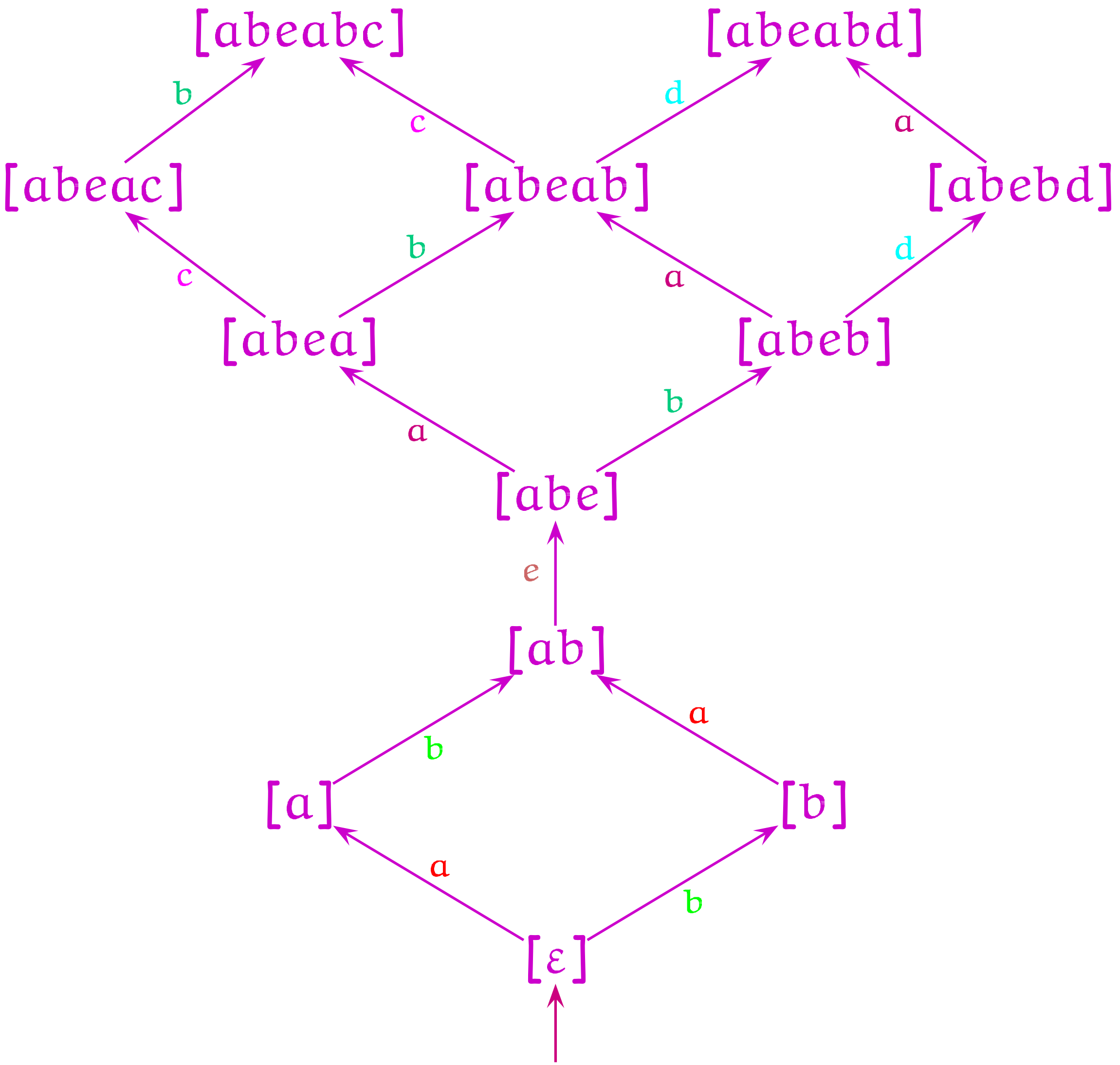
$a \| b$
$a \| d$
$b \| c$



How many events are there ?

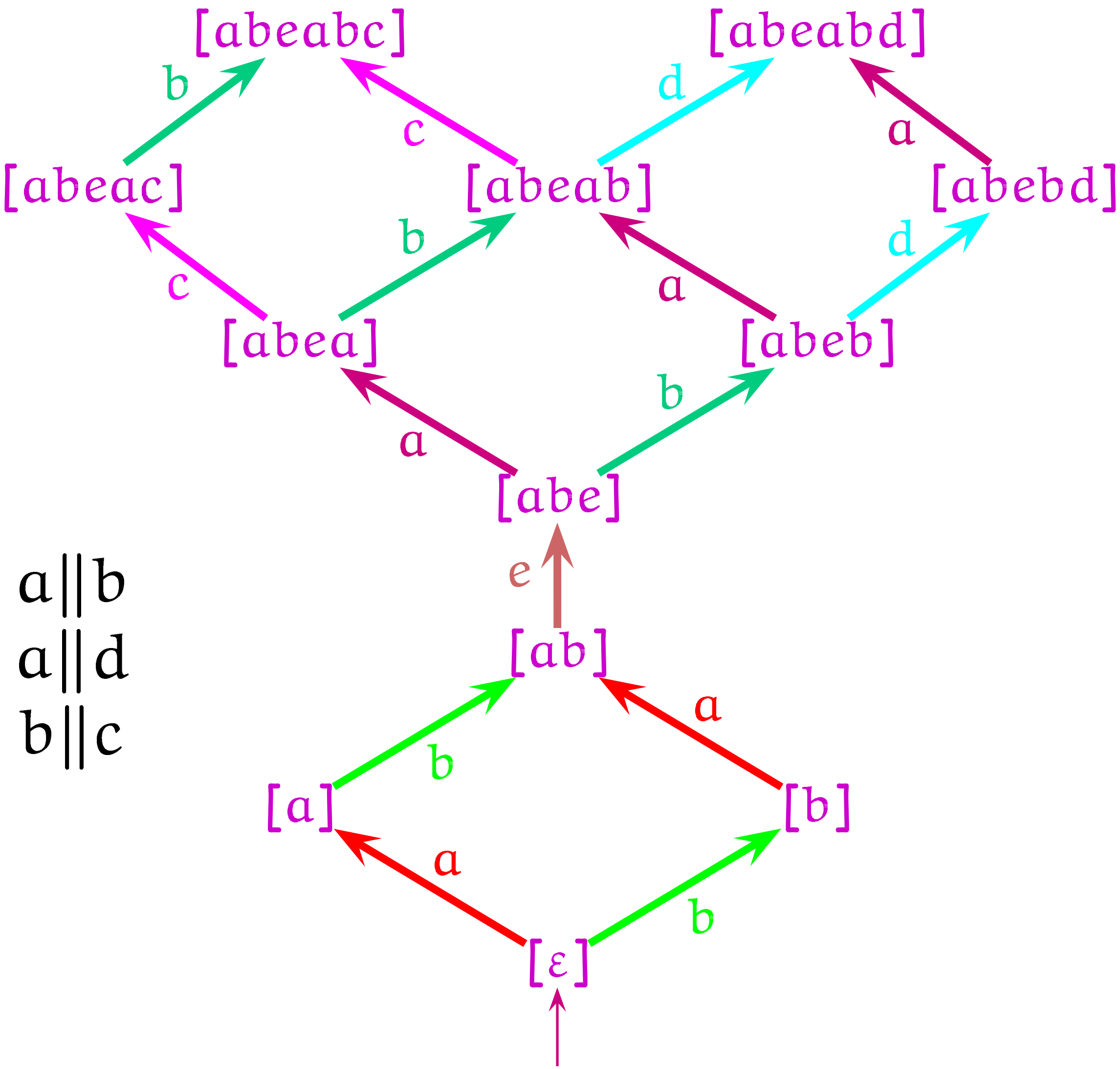
From trace languages to event structures: An example (1/3)

$a \| b$
$a \| d$
$b \| c$

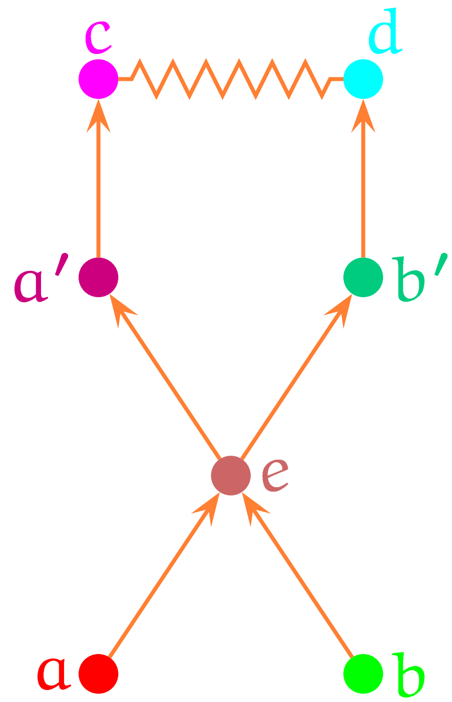


How many events are there ?

From trace languages to event structures: An example (2/3)

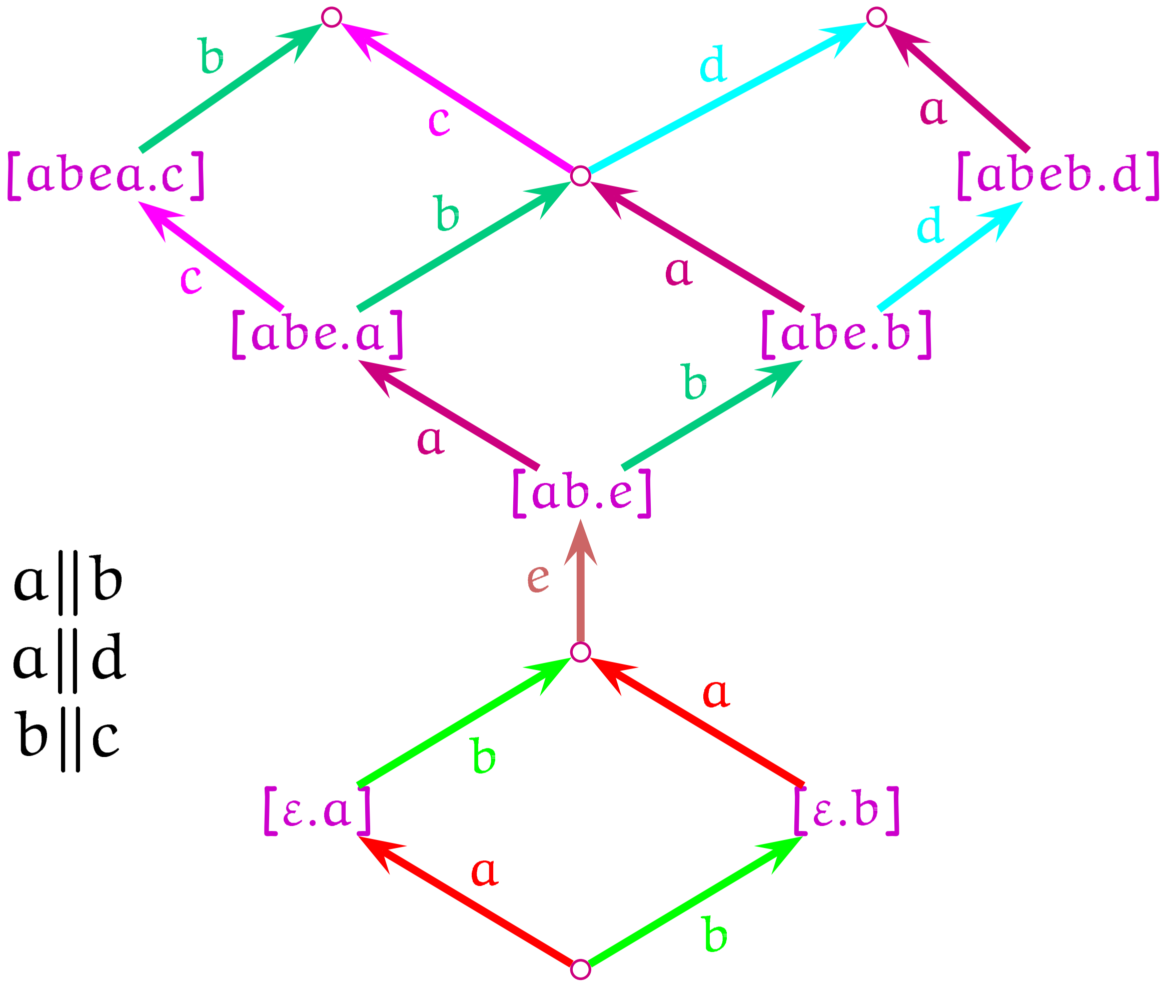


a||b
a||d
b||c

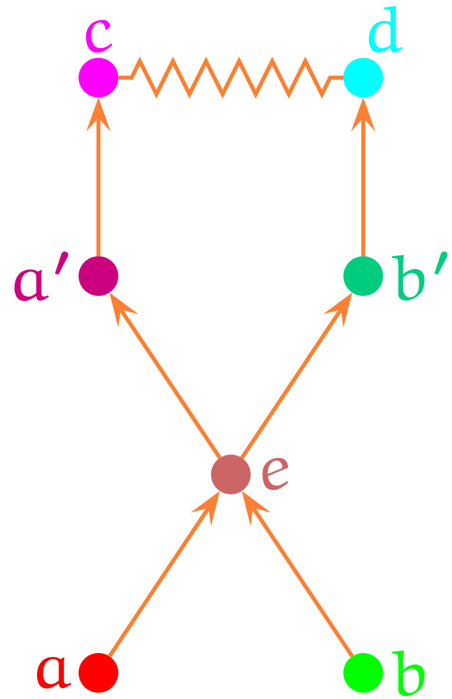


Events are equivalence classes of transitions !

An alternative equivalent approach



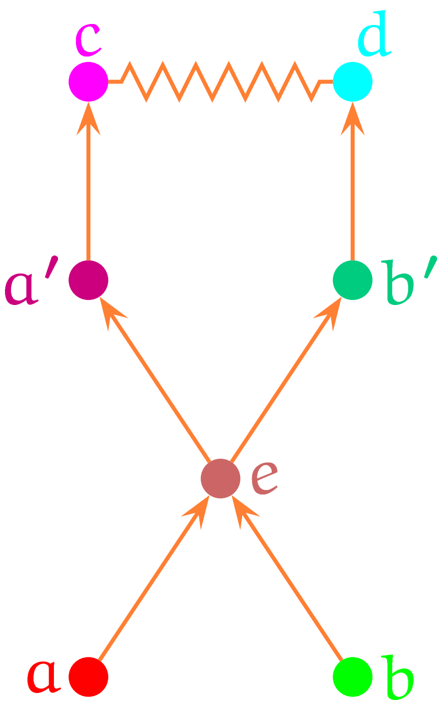
a||b
a||d
b||c

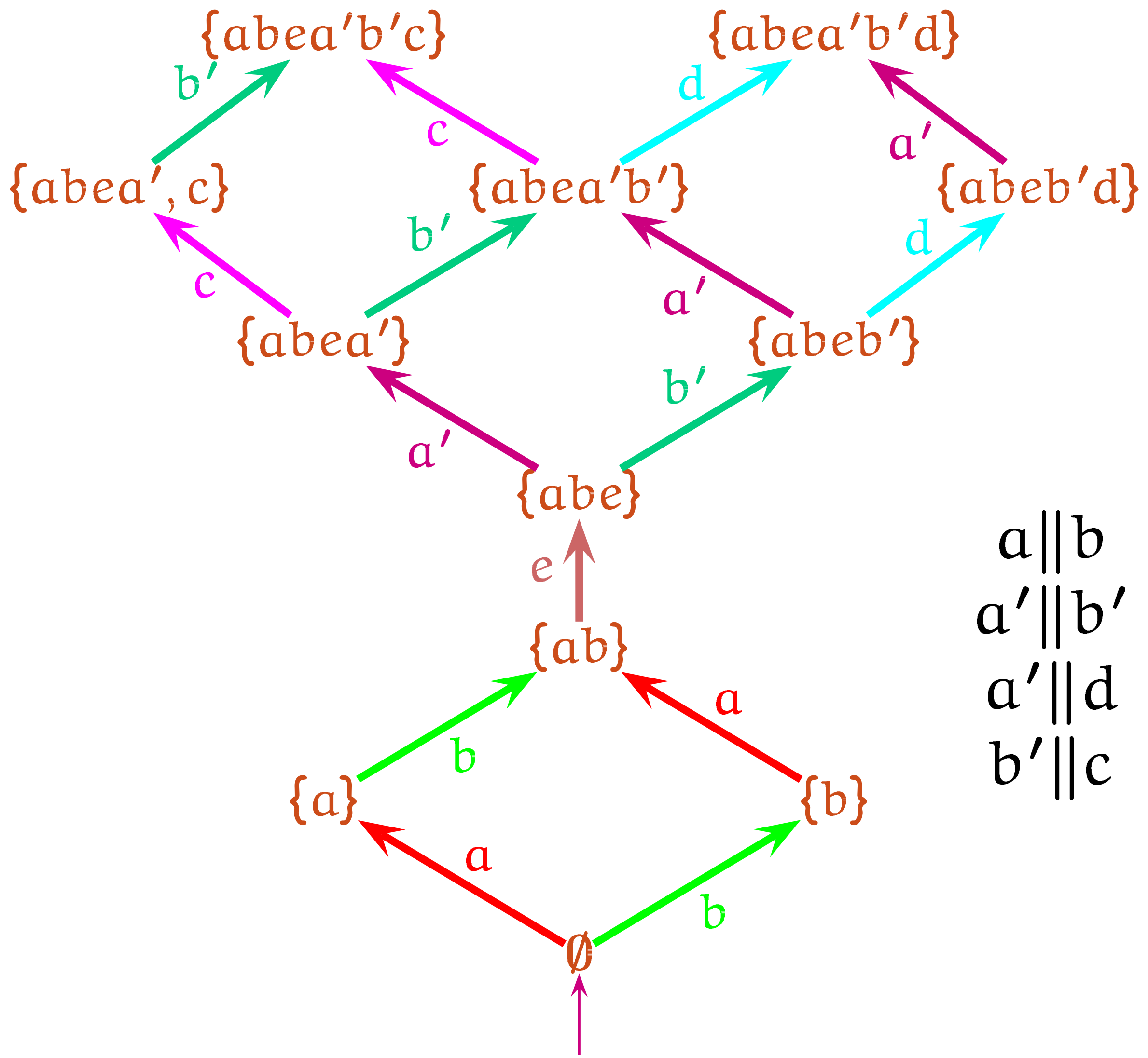


Events are prime traces, too!

From trace languages to event structures: An example (3/3)

Why is this event structure the right one ?





$a \| b$
$a' \| b'$
$a' \| d$
$b' \| c$

Let $\mathcal{L}$ be a prefix-closed coherent trace language over $(\Sigma, \|)$.
A **prime interval** is a pair $(u, a) \in \Sigma^\star \times \Sigma$ such that $u.a \in L$.

Definition

Projectivity $\asymp^p_{\mathcal{L}}$ is the least equivalence over $\mathsf{Pr}(\mathcal{L})$ such that

- $u.a.b \in L \wedge a\|b \Rightarrow (u, a) \asymp^p_{\mathcal{L}} (u.b, a)$ *[Independence]*
- $(u, a) \in \mathsf{Pr}(\mathcal{L}) \wedge (u', a) \in \mathsf{Pr}(\mathcal{L}) \wedge u \sim u' \Rightarrow (u, a) \asymp^p_{\mathcal{L}} (u', a)$ *[Confluence]*

Conflict and causality

For any word $u \in L$, the **set of events in** u is

$$\mathrm{Eve}(u) = \{\langle v, b\rangle_{\mathcal{L}} \mid v.b \leqslant u\}.$$

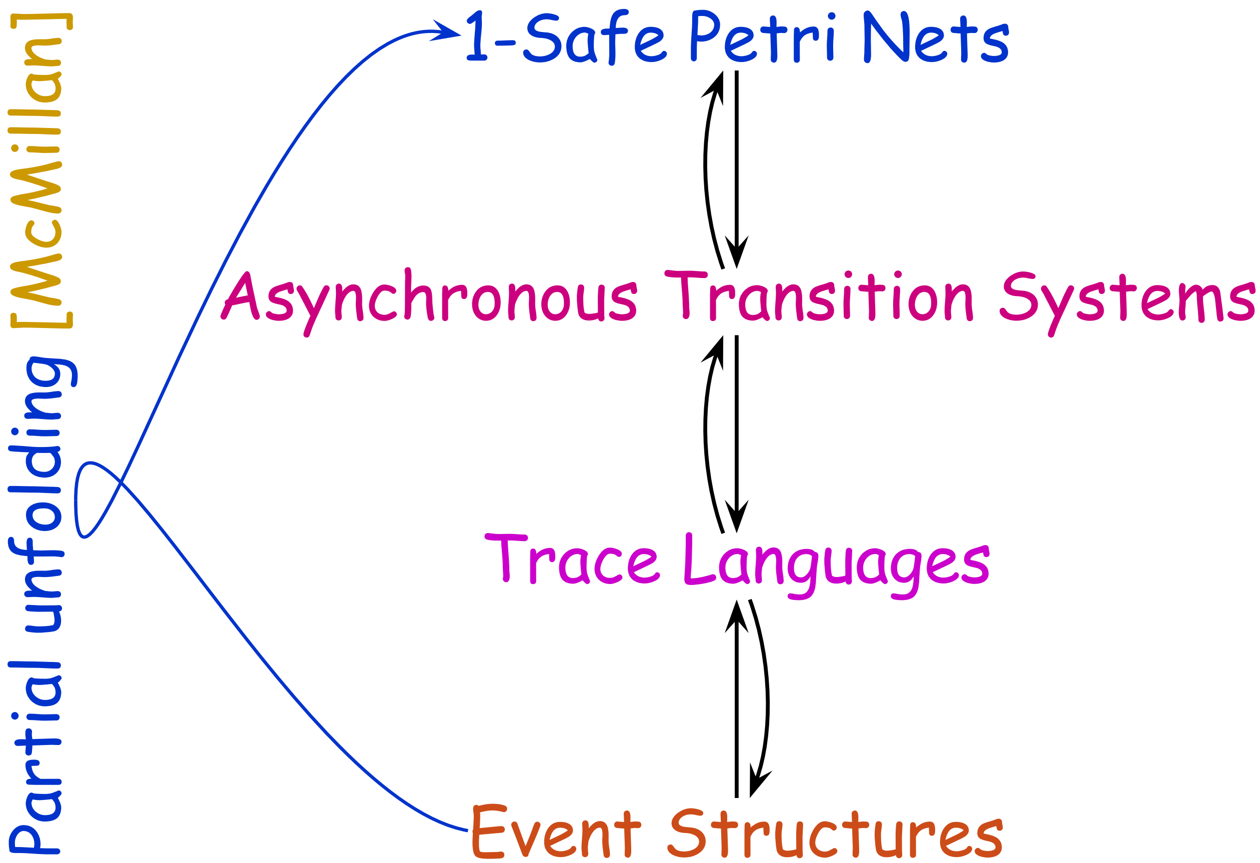
We put

$$\langle u, a\rangle \preccurlyeq \langle v, b\rangle \quad \text{if} \quad \forall w \in L, \langle v, b\rangle \in \mathrm{Eve}(w) \Rightarrow \langle u, a\rangle \in \mathrm{Eve}(w)$$

$$\langle u, a\rangle \sharp \langle v, b\rangle \quad \text{if} \quad \forall w \in L, \langle v, b\rangle \in \mathrm{Eve}(w) \Rightarrow \neg\langle u, a\rangle \in \mathrm{Eve}(w)$$

Proposition

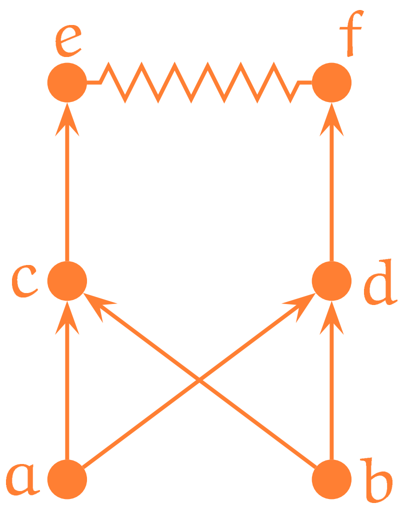
- $\mathcal{E} = (\mathrm{Pr}(\mathcal{L})/\asymp^p_{\mathcal{L}}, \preccurlyeq, \sharp)$ is an event structure.
- The configurations of $\mathcal{E}$ are $\{\mathrm{Eve}(w) \mid w \in L\}$.
- $u \sim u' \;\Leftrightarrow\; \mathrm{Eve}(u) = \mathrm{Eve}(u')$.
- $\phi : (\mathcal{L}, \leqslant) \to (C_f, \subseteq)$ is an isomorphism.



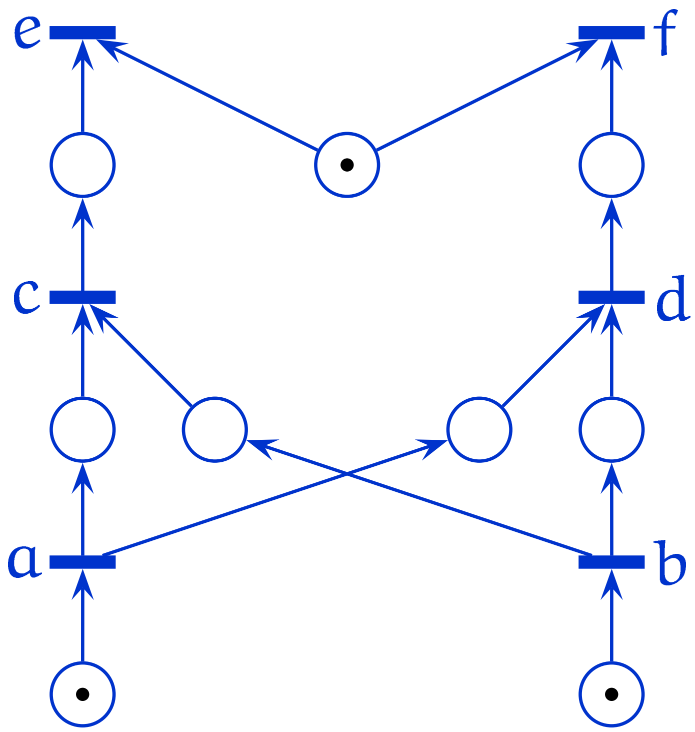
Basic ideas.

- Each configuration corresponds to some trace
- Each configuration maps to a reachable state
- Build one representative configuration for each reachable state

Back and forth between 1-safe Petri nets and event structures

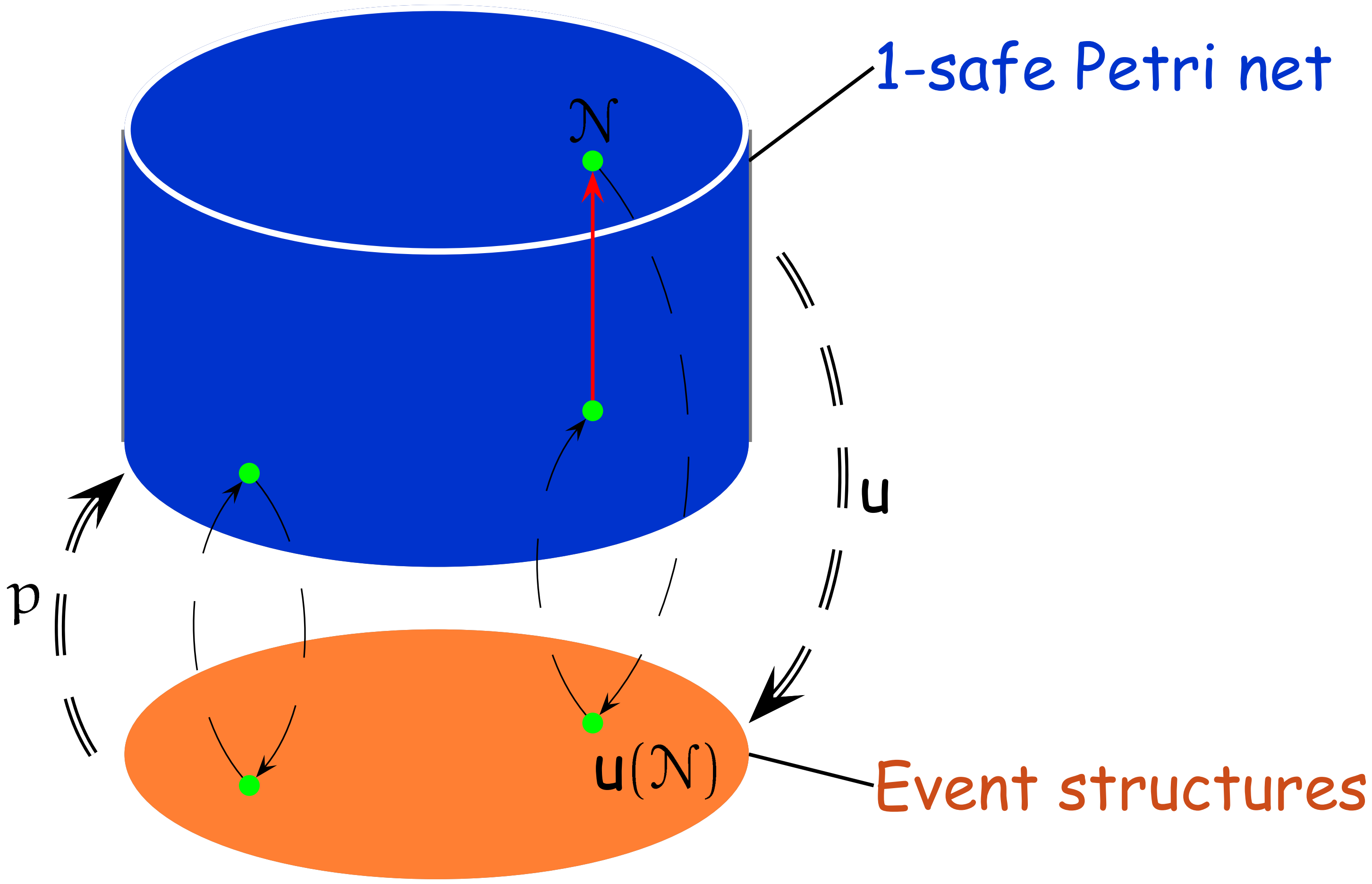


Event structure



Corresponding occurrence net

From 1-safe net to event structures (and back)



Theorem [Nielsen+Winskel]

Event structures are a coreflective sub-category of 1-safe nets.

EPIT32 — April 26-30, 2004 — CIRM

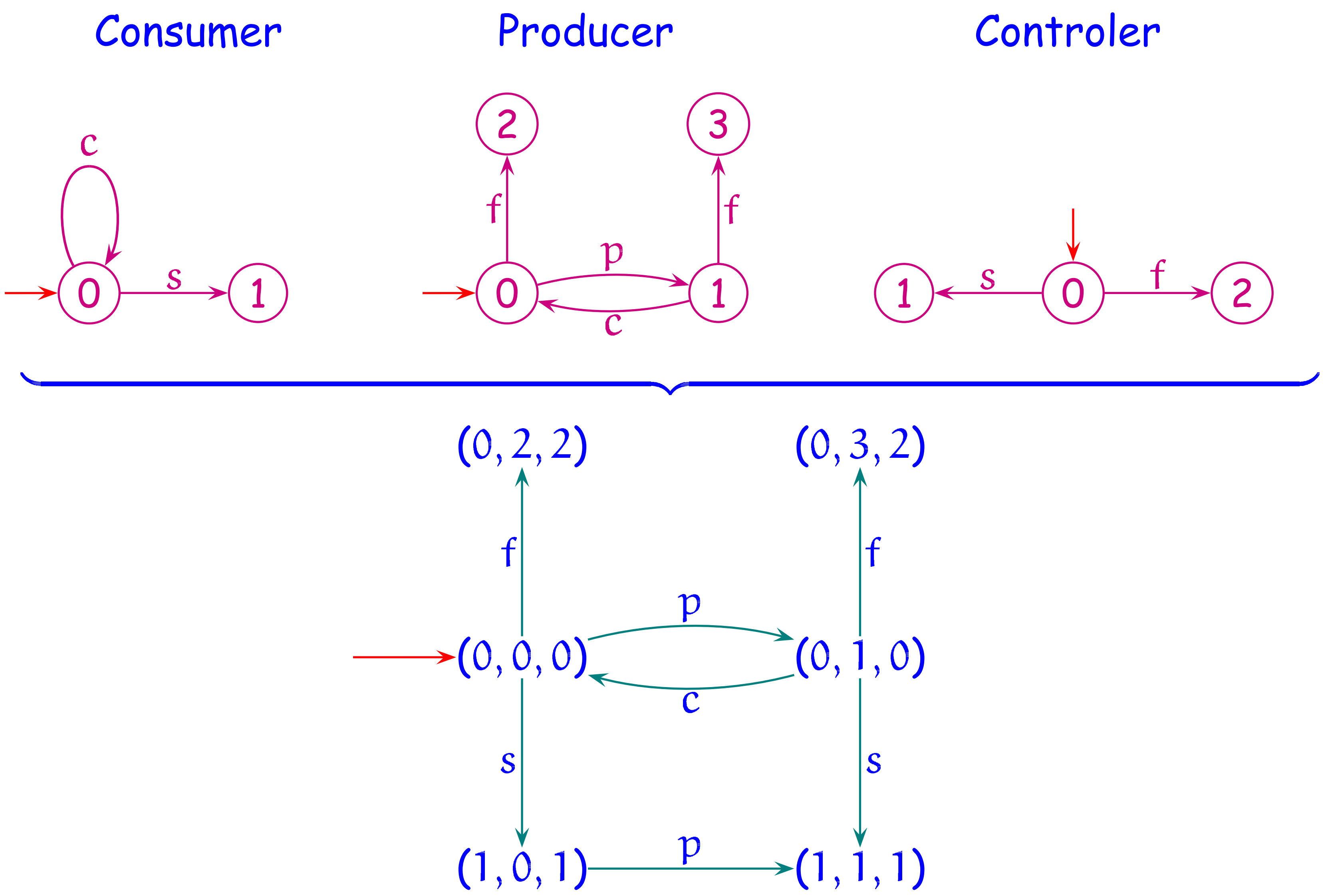
Theory of Mazurkiewicz traces

Rémi Morin

May 19, 2004

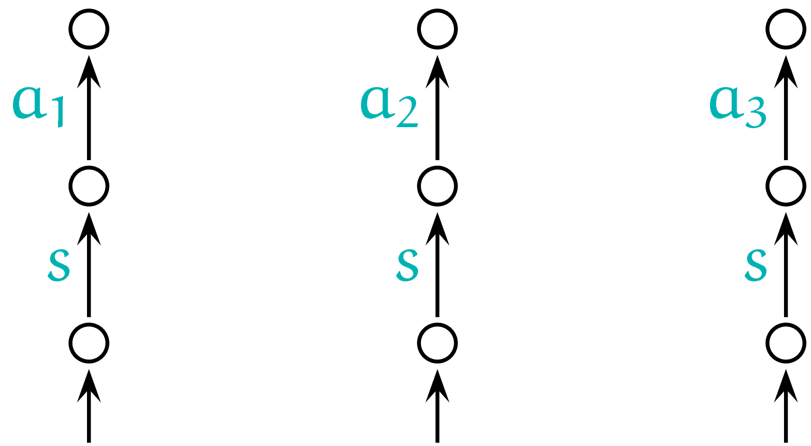
Synchronized product (a la Arnold-Nivat)

Synchronized product of automata: 1-safe Producer/Consumer



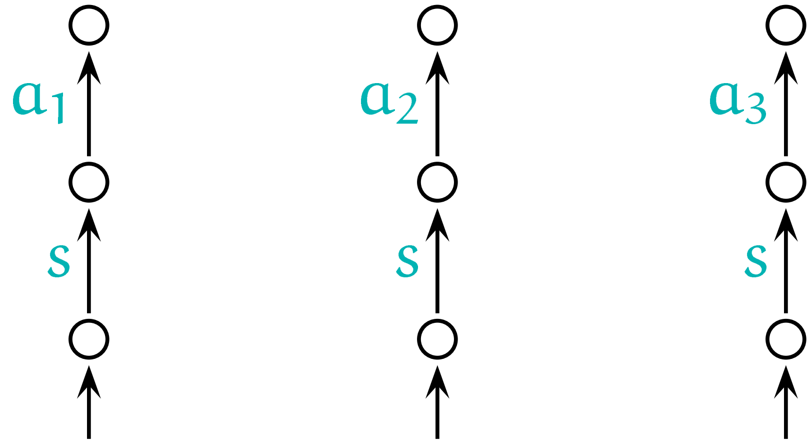
There are only 6 reachable global states

Synchronized product

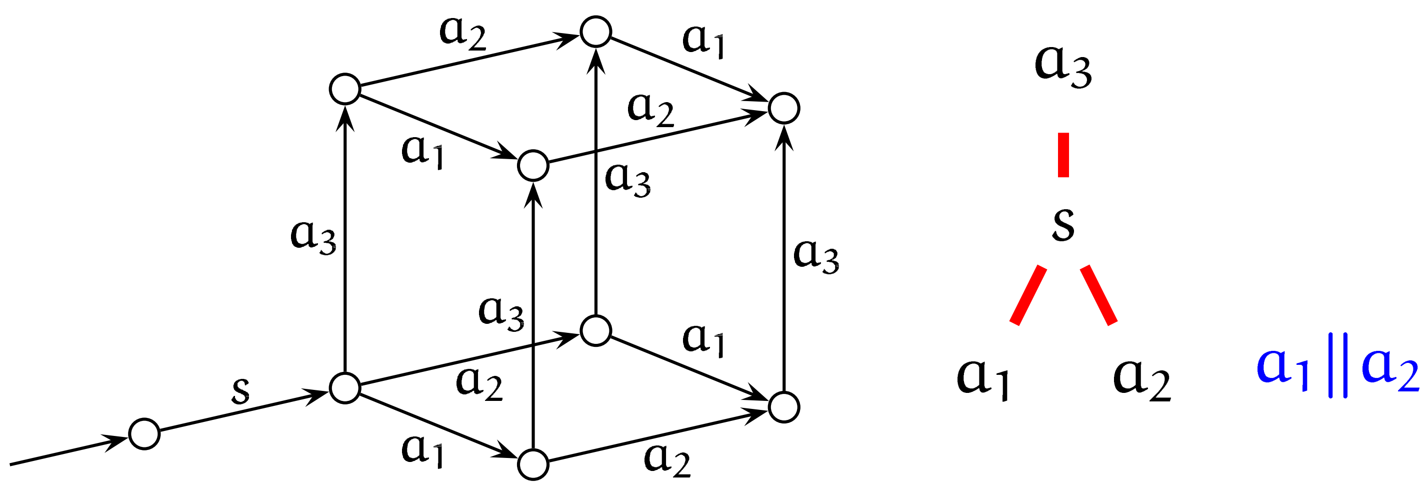


How many reachable global states are there?

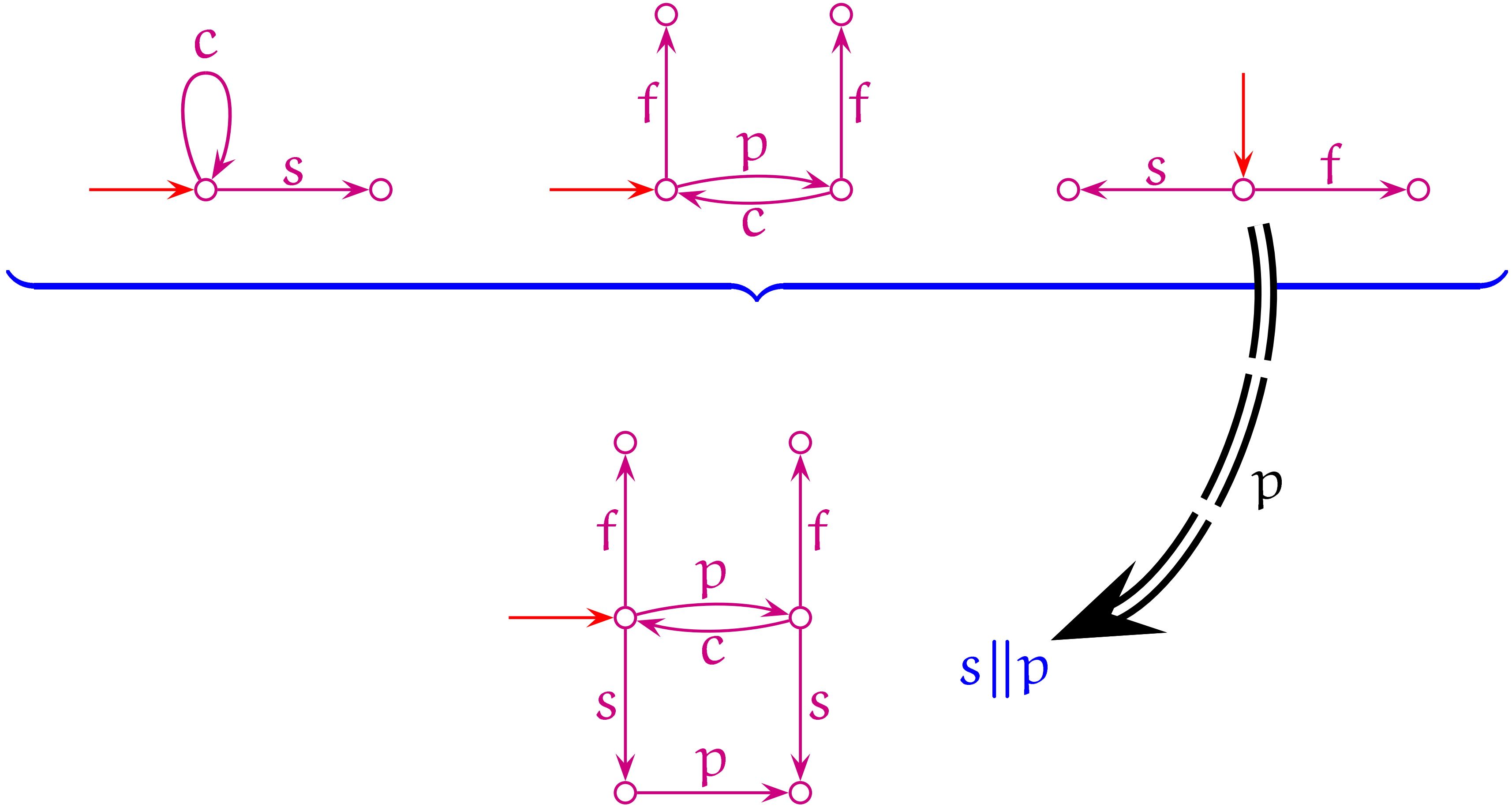
Synchronized product



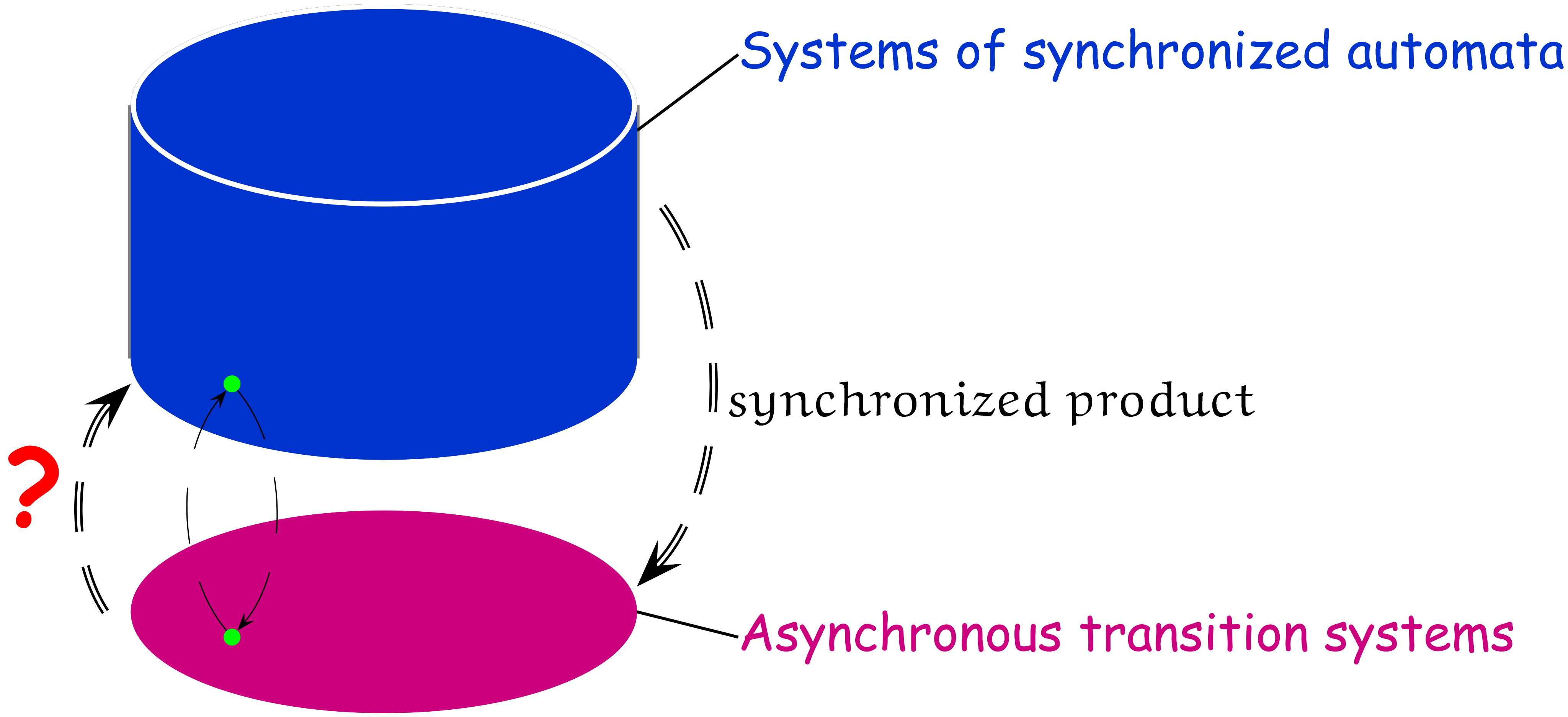
How many reachable global states are there?



From systems of processes to asynchronous systems

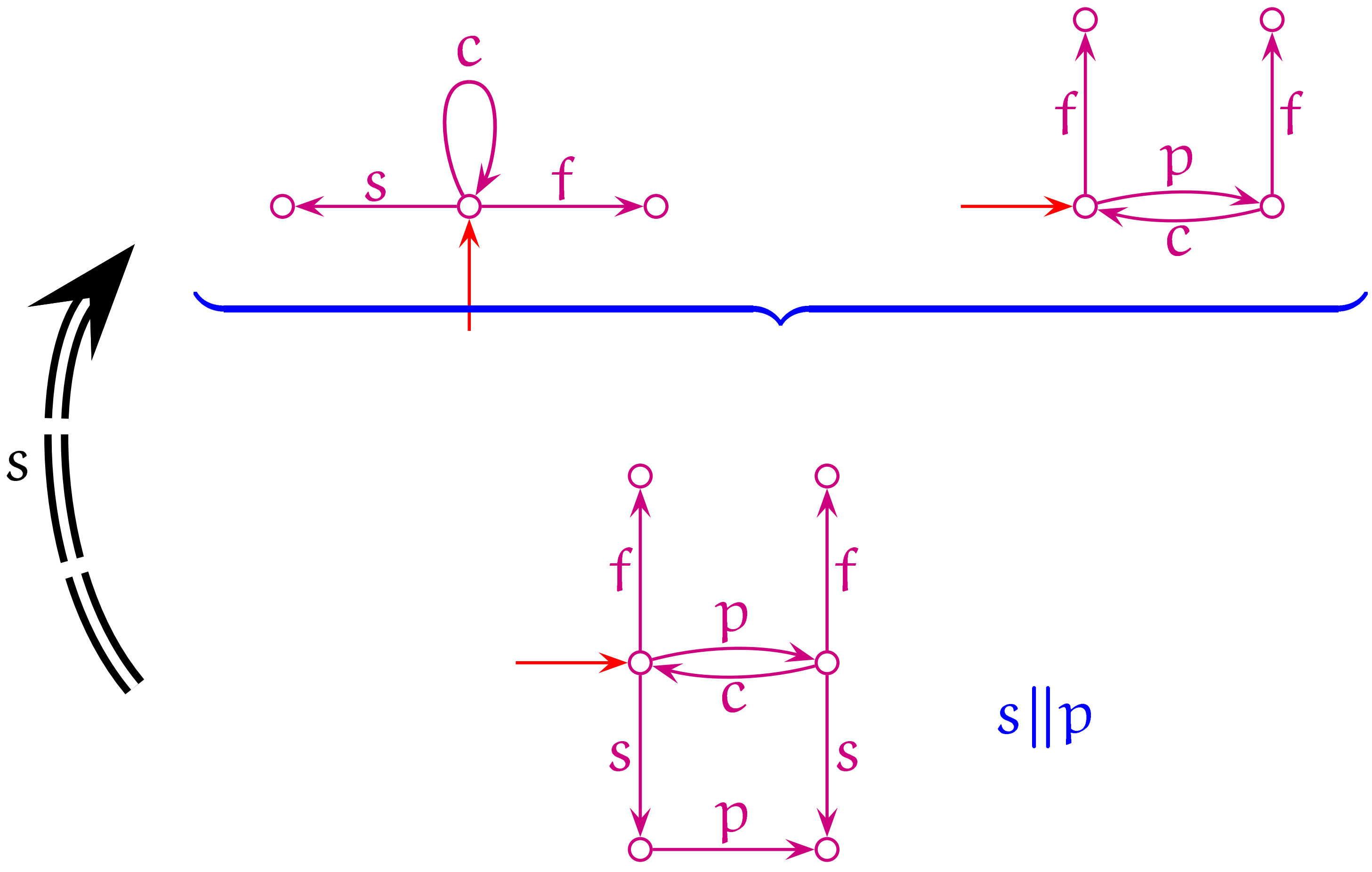


From synchronized automata to asynchronous transition systems (and back)



Theory of regions for synchronized automata

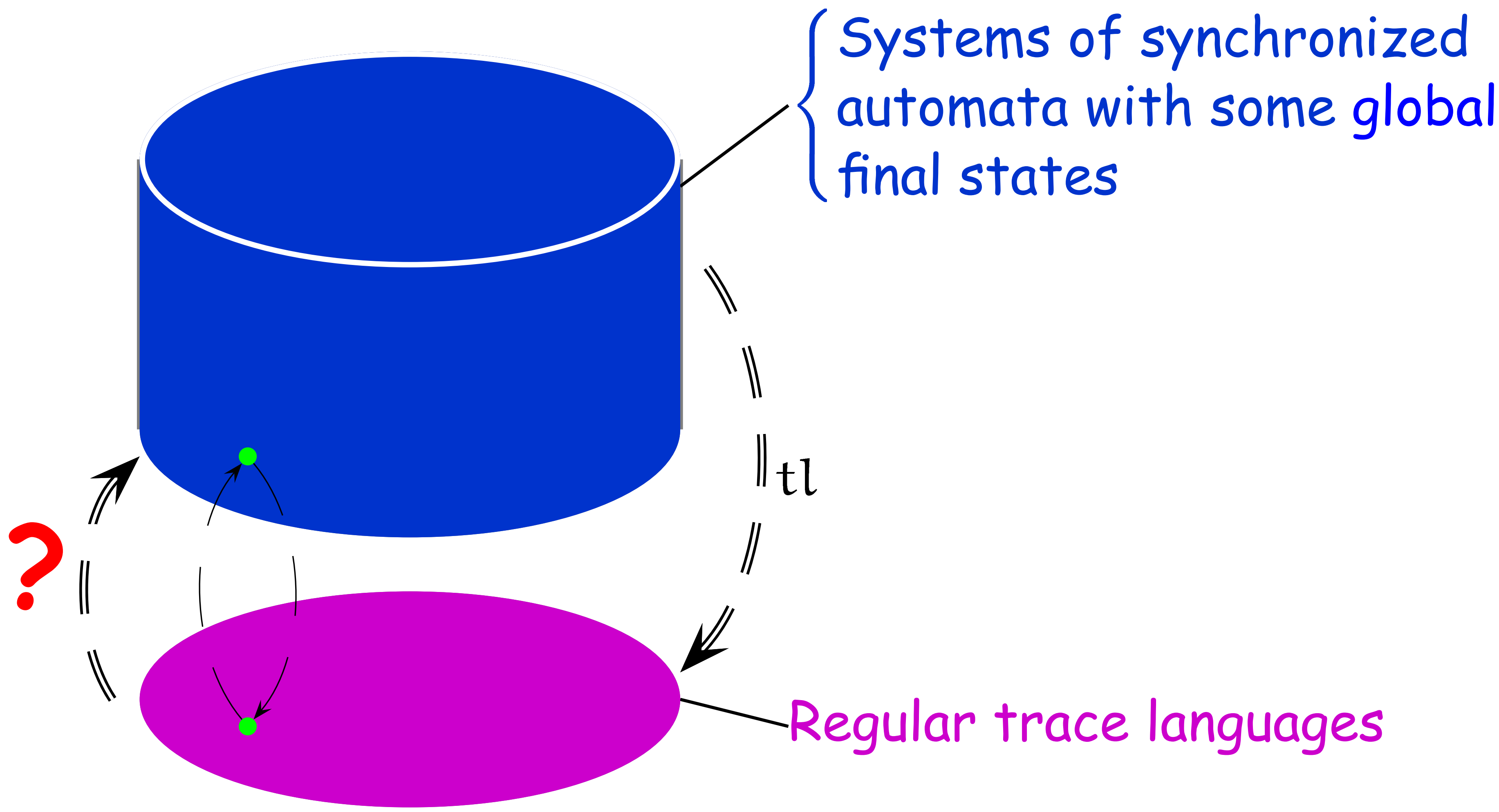
Synthesis of systems of processes



Decompositions depend on process alphabets

Is the structure of the automaton that important?

Synthesis problem for regular trace languages



We'll come back to this problem later

- ✔ *Synchronized product (a la Arnold-Nivat)*
- ☛ *Traces are labelled partial orders*

Mazurkiewicz traces

An independence relation over Σ is a binary, symmetric and irreflexive relation $\| \subseteq \Sigma \times \Sigma$.

The associated trace equivalence is the least equivalence relation $\sim$ over $\Sigma^\star$ such that $\forall u, v \in \Sigma^\star, \forall a, b \in \Sigma : a \| b \Rightarrow u.ab.v \sim u.ba.v$.

A trace $[u]$ is the equivalence class of a word $u \in \Sigma^\star$.

A trace language $\mathcal{L} \subseteq \mathbb{M}(\Sigma, \|)$ is identified to a subset $L \subseteq \Sigma^\star$ such that $u \sim v \wedge u \in L \Rightarrow v \in L$.

A trace language is

- prefix-closed if $u \leqslant v \wedge v \in L \Rightarrow u \in L$.
- coherent if $u.a \in L \wedge u.b \in L \wedge a \| b \Rightarrow u.ab \in L$.
- regular if it is accepted by a finite asynchronous transition system provided with some final states.

Pratt's pomsets

A pomset over an alphabet Σ is a triple $t = (E, \preccurlyeq, \xi)$ where $(E, \preccurlyeq)$ is a finite partial order and ξ is a mapping from E to Σ.
We denote by $\mathbb{P}(\Sigma)$ the class of all pomsets over Σ.
In this talk we consider only pomsets without auto-concurrency:

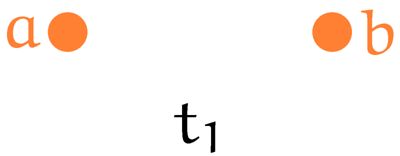
$$\xi(x) = \xi(y) \text{ implies } (x \preccurlyeq y \text{ or } y \preccurlyeq x)$$

An order extension of $t = (E, \preccurlyeq, \xi)$ is a pomset $t' = (E, \preccurlyeq', \xi)$ such that $\preccurlyeq \subseteq \preccurlyeq'$. A linear extension of t is an order extension that is linearly ordered. We denote by $LE(t) \subseteq \Sigma^\star$ the set of linear extensions of t.

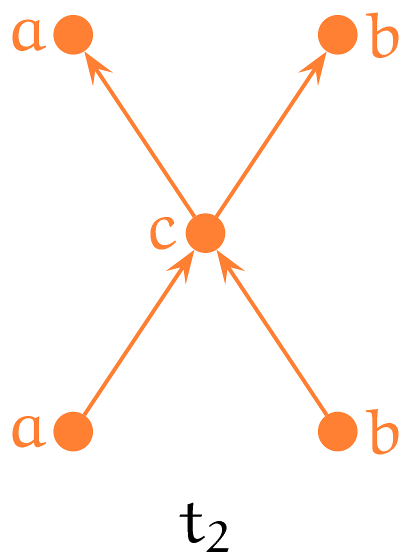
Theorem [Spilzrajn]

Two pomsets t and t' are isomorphic if and only if $LE(t) = LE(t')$.

Two examples



$\mathsf{LE}(t_1) = \{ab, ba\}$



$\mathsf{LE}(t_2) = \{abcab, bacab, abcba, bacba\}$

Mazurkiewicz traces are pomsets

Let $(\Sigma, \|)$ be an independence alphabet.
A trace $[u]$ is an equivalence class of words.

For any word $u \in \Sigma^\star$, $[u] = \mathsf{LE}(t)$ for a (unique) pomset $t = (E, \preccurlyeq, \xi)$.
Moreover t satisfies

MP$_1$: $\forall e_1, e_2 \in E$: $\xi(e_1) \not\| \xi(e_2) \Rightarrow e_1 \preccurlyeq e_2$ **or** $e_2 \preccurlyeq e_1$;

MP$_2$: $\forall e_1, e_2 \in E$: $e_1 \longrightarrow\!\!\prec e_2 \Rightarrow \xi(e_1) \not\| \xi(e_2)$.

where $x \longrightarrow\!\!\prec y$ if $x \prec y$ and $x \prec z \preccurlyeq y$ implies $y = z$.

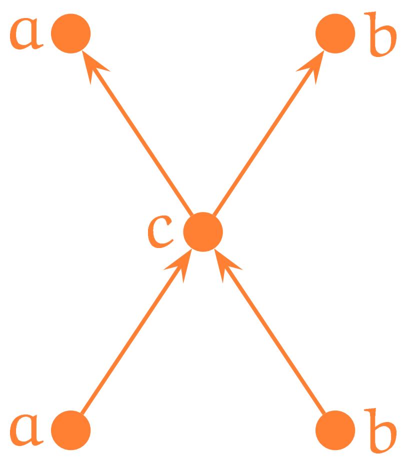
Conversely any pomset t satisfying these two axioms is such that $\mathsf{LE}(t) = [u]$ for some word u.

Connected traces

Consider $\Sigma = \{a, b, c\}$ with $a \| b$, $a \not\| c$ and $b \not\| c$.

$[ab] = \{ab, ba\}$

$[abcab] = \{abcab, bacab, abcba, bacba\}$



A trace is connected if its associated partial order is a connected graph.

How to get partial orders from traces?

How to get events from traces?

- Events are projectivity equivalence classes of prime intervals
- Events are prime traces
- Traces are heaps of pieces

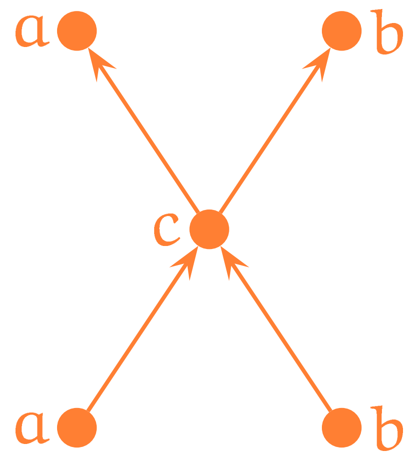
Heaps of pieces (1/5)

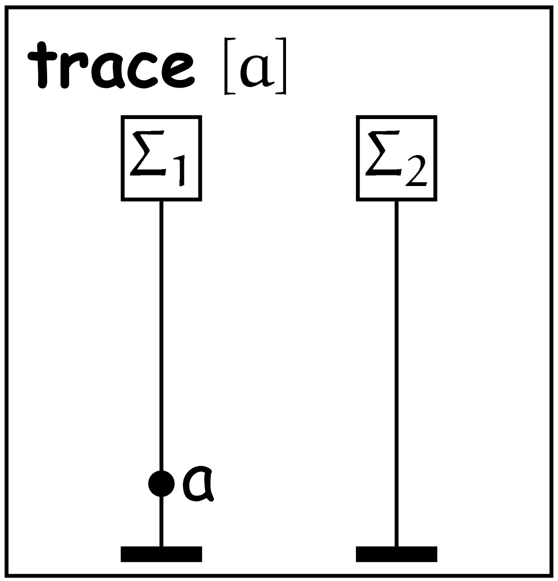
A distributed alphabet is a family of subsets $\Sigma_1, \ldots, \Sigma_k \subseteq \Sigma$.
It induces an independence relation: $a \not\parallel b$ iff $\exists i \in [1, k], \{a, b\} \in \Sigma_i$ (considered e.g. in the synchronized product).

Example

Consider $\Sigma = \{a, b, c\}$ with $a \| b$, $a \not\parallel c$ and $b \not\parallel c$.
We put $\Sigma_1 = \{a, c\}$ and $\Sigma_2 = \{b, c\}$.

How about $[abcab]$?





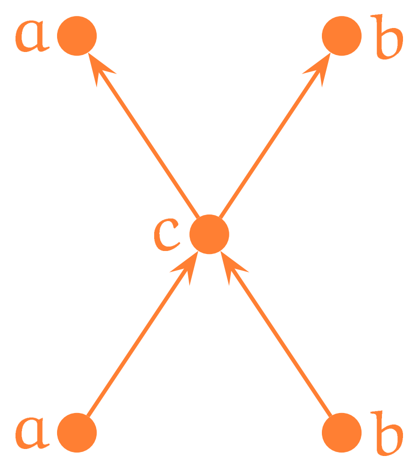
Heaps of pieces (2/5)

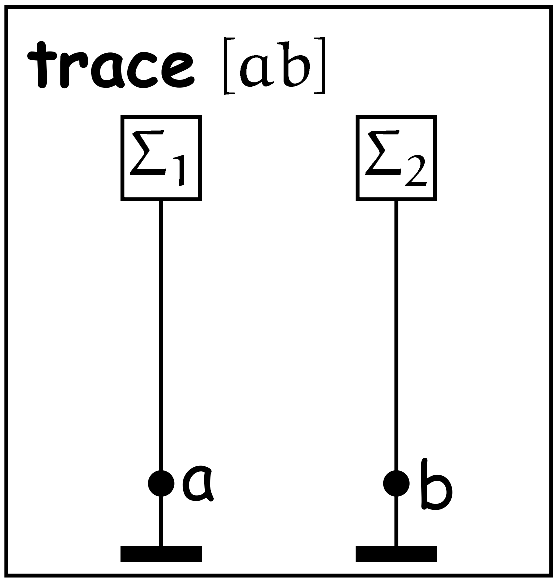
A distributed alphabet is a family of subsets $\Sigma_1, \ldots, \Sigma_k \subseteq \Sigma$.
It induces an independence relation: $a \not\| b$ iff $\exists i \in [1,k], \{a,b\} \in \Sigma_i$
(considered e.g. in the synchronized product).

Example

Consider $\Sigma = \{a,b,c\}$ with $a \| b$, $a \not\| c$ and $b \not\| c$.
We put $\Sigma_1 = \{a,c\}$ and $\Sigma_2 = \{b,c\}$.

How about $[abcab]$?





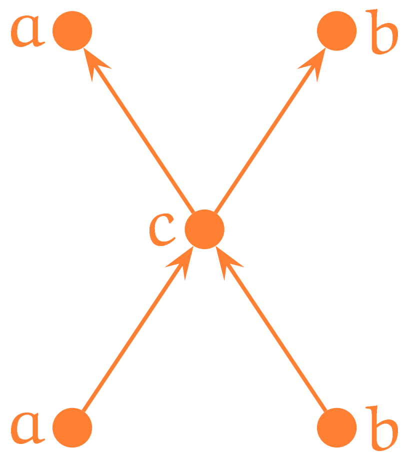
Heaps of pieces (3/5)

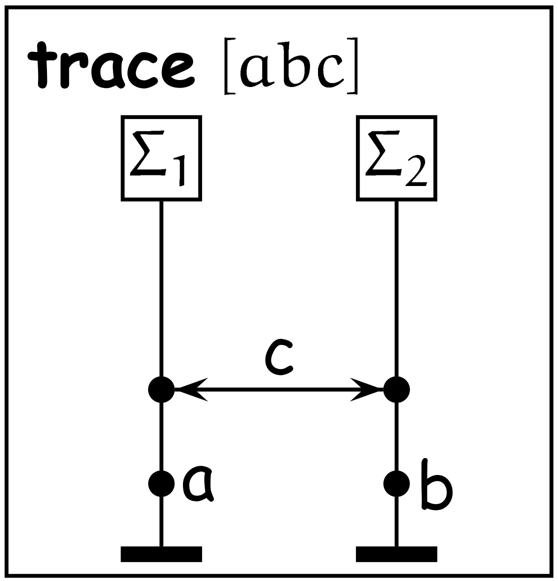
A distributed alphabet is a family of subsets $\Sigma_1, \ldots, \Sigma_k \subseteq \Sigma$.
It induces an independence relation: $a \not\parallel b$ iff $\exists i \in [1,k], \{a,b\} \in \Sigma_i$
(considered e.g. in the synchronized product).

Example

Consider $\Sigma = \{a, b, c\}$ with $a \| b$, $a \not\parallel c$ and $b \not\parallel c$.
We put $\Sigma_1 = \{a, c\}$ and $\Sigma_2 = \{b, c\}$.

How about $[abcab]$?





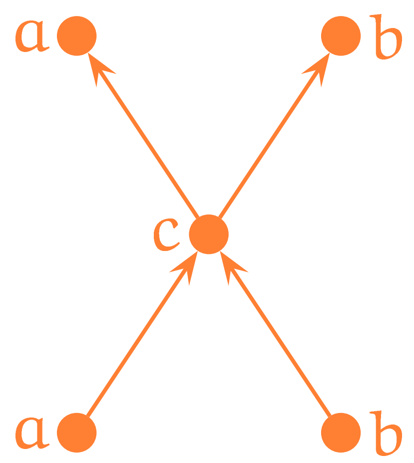
Heaps of pieces (4/5)

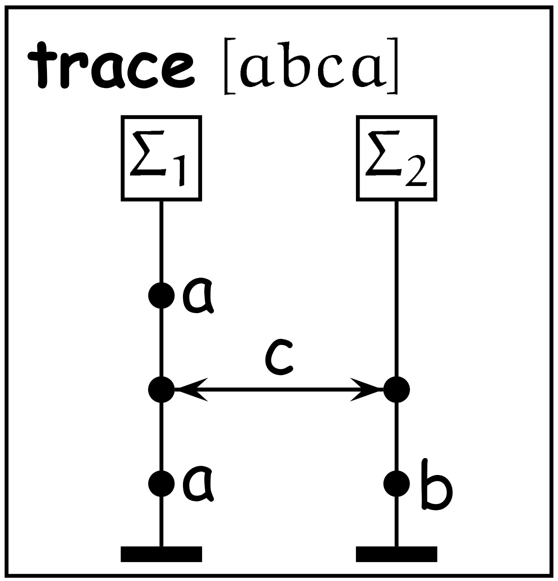
A distributed alphabet is a family of subsets $\Sigma_1,\ldots,\Sigma_k \subseteq \Sigma$.
It induces an independence relation: $a \not\parallel b$ iff $\exists i \in [1,k], \{a,b\} \in \Sigma_i$
(considered e.g. in the synchronized product).

Example

Consider $\Sigma = \{a,b,c\}$ with $a \parallel b$, $a \not\parallel c$ and $b \not\parallel c$.
We put $\Sigma_1 = \{a,c\}$ and $\Sigma_2 = \{b,c\}$.

How about $[abcab]$?





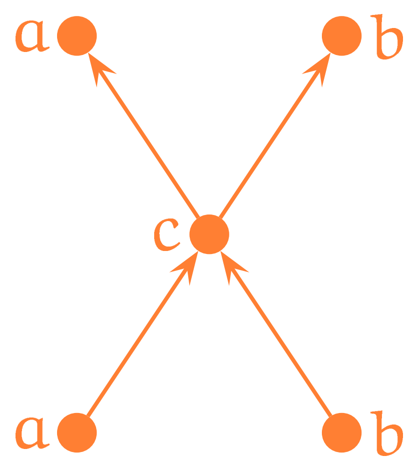
Heaps of pieces (5/5)

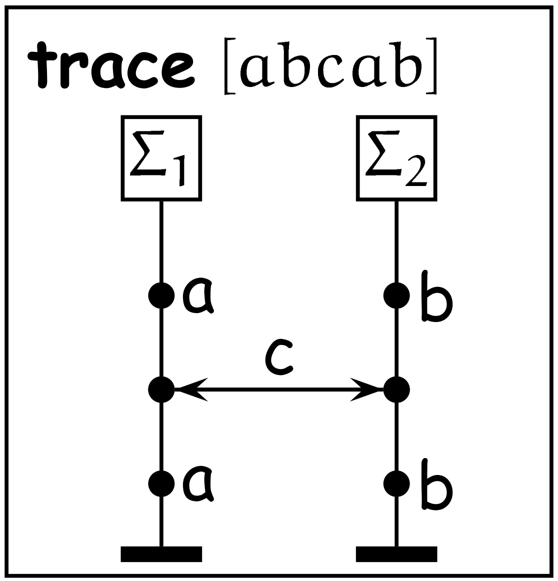
A distributed alphabet is a family of subsets $\Sigma_1,\ldots,\Sigma_k \subseteq \Sigma$.
It induces an independence relation: $a \not\parallel b$ iff $\exists i \in [1,k], \{a,b\} \in \Sigma_i$
(considered e.g. in the synchronized product).

Example

Consider $\Sigma = \{a,b,c\}$ with $a \| b$, $a \not\parallel c$ and $b \not\parallel c$.
We put $\Sigma_1 = \{a,c\}$ and $\Sigma_2 = \{b,c\}$.

How about $[abcab]$*?*





✔ *Synchronized product (a la Arnold-Nivat)*

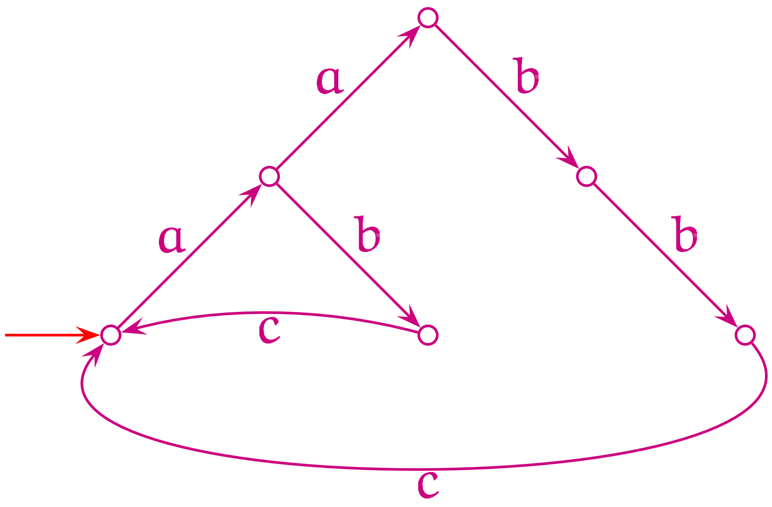
✔ *Traces are labelled partial orders*

☛ *Trace monoid*

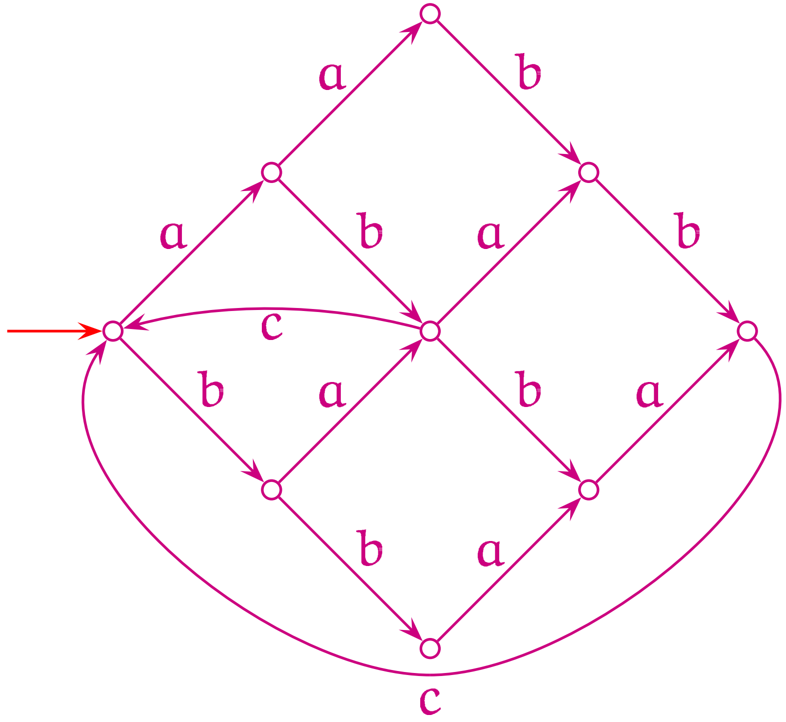
Two key examples

We consider $\Sigma = \{a, b, c\}$ with $a \| b$, $a \not\| c$ and $b \not\| c$.

- $(((a \cdot b) + (a \cdot a \cdot b \cdot b)) \cdot c)^\star$

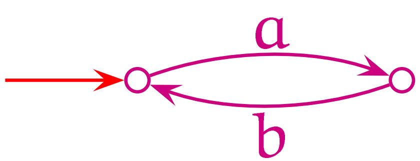


Associated words in $\Sigma^\star$



Associated traces in $\mathbb{M}(\Sigma, \|)$

- $(a \cdot b)^\star$



$L = \{u \in \{a, b\}^\star \mid |u|_a = |u|_b\}$ is not regular!

Trace monoid

We put $[u] \cdot [v] = [u.v]$. This operation

- admits the empty trace as unit: $1 \cdot [u] = [u] \cdot 1 = [u]$, and
- is associative: $[u] \cdot ([v] \cdot [w]) = ([u] \cdot [v]) \cdot [w]$

$\mathbb{M}(\Sigma, \|)$ is called the trace monoid over $(\Sigma, \|)$.

Rational subsets of monoids

Let $(\mathbb{M}, \cdot)$ be a monoid with unit $\mathbf{1}$.
For any $\mathcal{L}, \mathcal{L}' \subseteq \mathbb{M}$, $\mathcal{L} \cdot \mathcal{L}' := \{x \cdot x' \mid x \in \mathcal{L} \wedge x' \in \mathcal{L}'\}$.
We let $\mathcal{L}^0 := \{\mathbf{1}\}$ and for any $n \in \mathbb{N}$, $\mathcal{L}^{n+1} := \mathcal{L}^n \cdot \mathcal{L}$;
then $\mathcal{L}^\star := \bigcup_{n \in \mathbb{N}} \mathcal{L}^n$ — also denoted $\langle \mathcal{L} \rangle_{\mathbb{M}}$.

A subset of $\mathbb{M}$ is rational if it can be obtained from the finite subsets of $\mathbb{M}$ by means of unions, products and iterations.

Usually, $\mathbb{M}$ is the free monoid $\Sigma^\star$.
In this talk, we consider the trace monoid $\mathbb{M}(\Sigma, \|)$ and the monoid of basic MSCs $\mathrm{b}\mathbb{MSC}$.

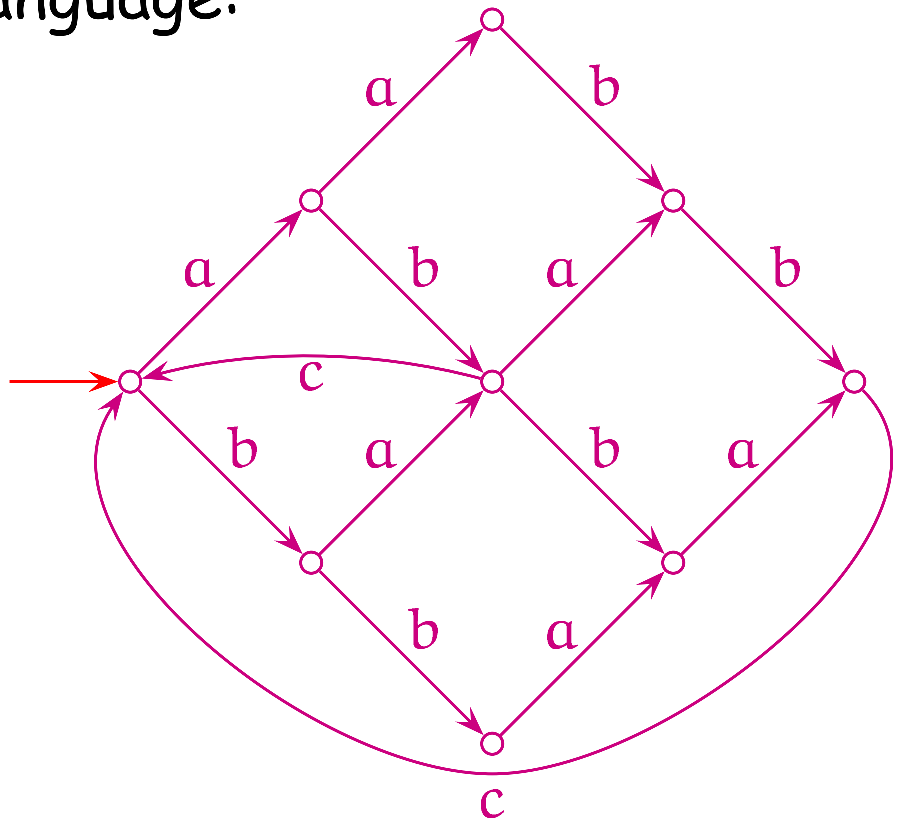
Some classical results and a big question

- It is undecidable whether a rational expression describes a regular language [Sakarovitch92].
- A rational expression is c-rational if iteration $\star$ occurs over sets of connected traces only.
  C-rational expressions describe regular languages [Ochmański85].
- Any regular language is described by some c-rational expression [Ochmański85].
- Still open: Decide whether the iteration of some given regular language is regular?

- ✔ *Synchronized product (a la Arnold-Nivat)*
- ✔ *Traces are labelled partial orders*
- ✔ *Trace monoid*
- ☛ *Back to the synthesis problem*

From regular trace languages to synchronized products?

Let $\Sigma = \{a, b, c\}$ with $a \| b$, $a \not\| c$ and $b \not\| a$.
The c-rational expression $(((a \cdot b) + (a \cdot a \cdot b \cdot b)) \cdot c)^{\star}$ corresponds to a regular trace language:



However [Zielonka87]

no synchronized product of automata accepts L*!*

Let Σ and Σ' be two alphabets and $\lambda : \Sigma' \to \Sigma$ a mapping.
Then λ extends to $\lambda : \mathbb{P}(\Sigma) \to \mathbb{P}(\Sigma')$.

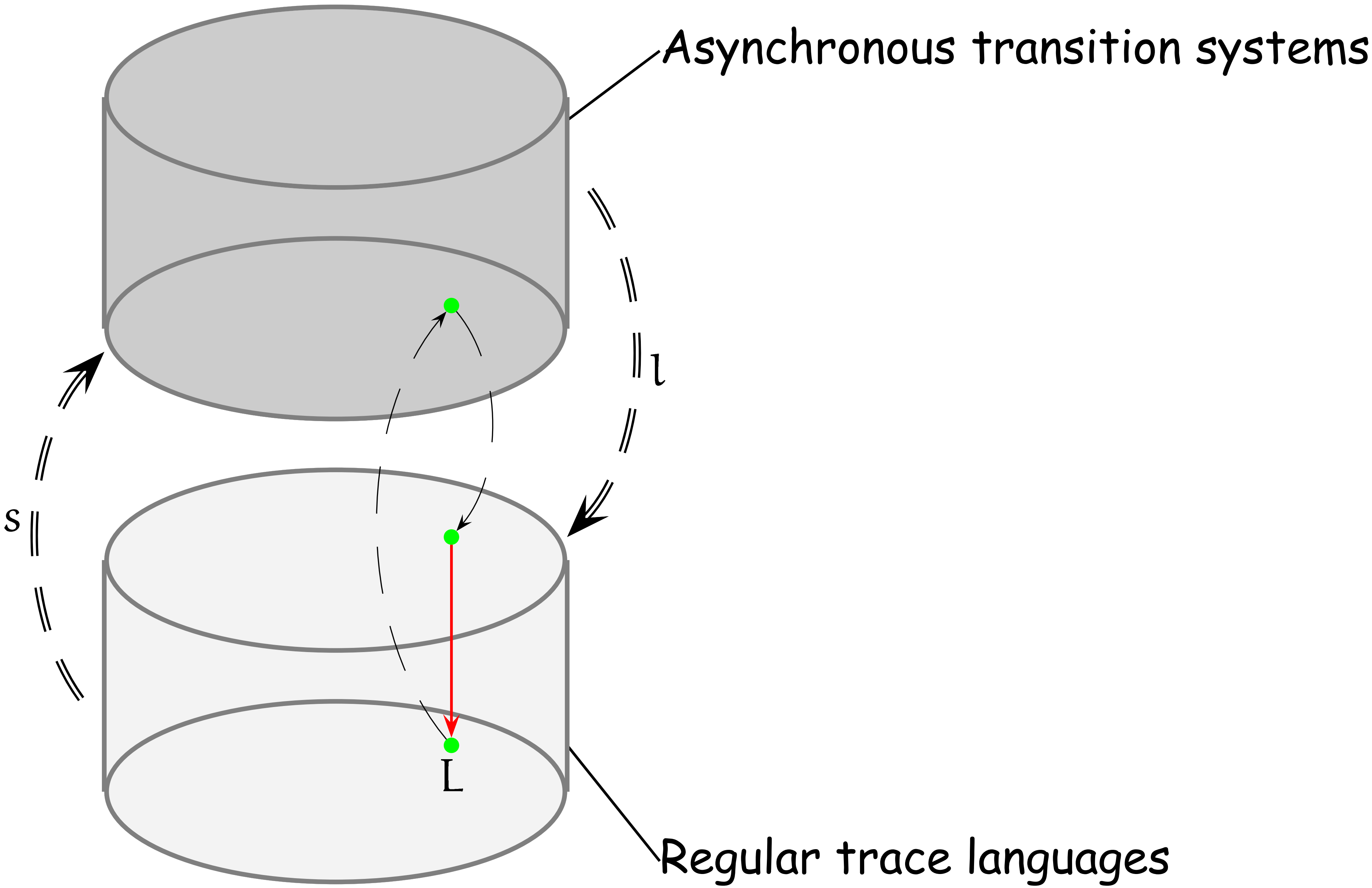
Definition

Let $\mathcal{L} \subseteq \mathbb{P}(\Sigma)$ and $\mathcal{L}' \subseteq \mathbb{P}(\Sigma')$ be two sets of pomsets over Σ and Σ' respectively. We say that $\mathcal{L}'$ is a refinement of $\mathcal{L}$ up to λ if the labelling λ induces a bijection from $\mathcal{L}'$ onto $\mathcal{L}$.

Theorem [Zielonka87]

For any regular trace language L over $(\Sigma, \|)$ there is a system of finite automata $\mathcal{S} = (\mathcal{A}_i)_{i \in 1 \leqslant i \leqslant n}$ over Σ' with particular final global states such that $\mathrm{tl}(\mathcal{S})$ is a refinement of L up to some labelling $\lambda : \Sigma' \to \Sigma$.

From regular trace languages to synchronized products!



EPIT32 — April 26-30, 2004 — CIRM

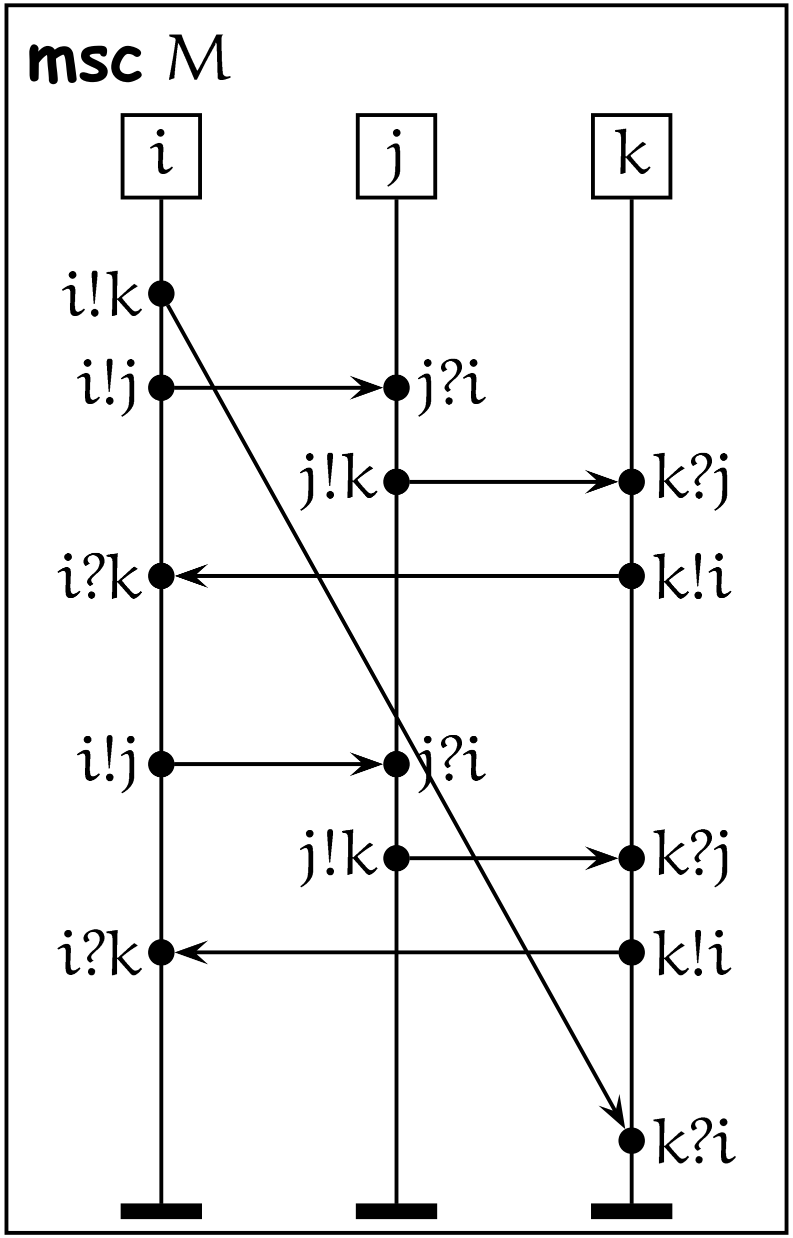
Introduction to message sequence charts

Rémi Morin

May 19, 2004

Communicating systems

Basic MSCs are labelled partial orders



Finite set of processes (called instances).

No non-deterministic choice:

- Lamport's diagrams;
- Nielsen, Plotkin & Winskel's elementary event structures;
- Pratt's pomsets.

Reliable channels.

Complete specification.

In my opinion, Basic Message Sequence Charts are pomsets

Let $\mathcal{I}$ be a finite set of instances and Λ be a set of messages.
$\forall i \in \mathcal{I}$, $\Sigma_i := \Sigma_i^{int} \uplus \{i!^x j, i?^x j \mid j \in \mathcal{I} \setminus \{i\}, x \in \Lambda\}$.
Then $\Sigma_{\mathcal{I}} := \biguplus_{i \in \mathcal{I}} \Sigma_i$ and $\forall a \in \Sigma_{\mathcal{I}}, \mathbf{Ins}(a) = i :\Leftrightarrow a \in \Sigma_i$.
For any pomset $(E, \preccurlyeq, \xi)$ over $\Sigma_{\mathcal{I}}$, $\mathbf{Ins}(e) := \mathbf{Ins}(\xi(e))$.

A basic MSC is a pomset $M = (E, \preccurlyeq, \xi)$ over $\Sigma_{\mathcal{I}}$ such that

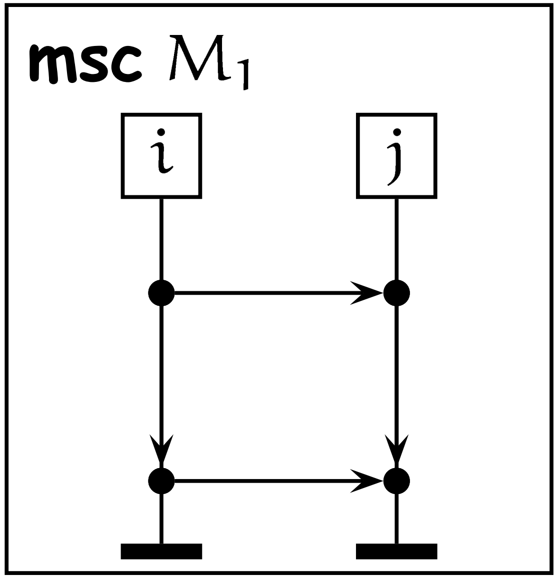
M_1: $\forall e, f \in E$: $\mathbf{Ins}(e) = \mathbf{Ins}(f) \Rightarrow (e \preccurlyeq f \vee f \preccurlyeq e)$

M_2: $\#^{i!^x j}(E) = \#^{j?^x i}(E)$ for any channel (i, j, x)

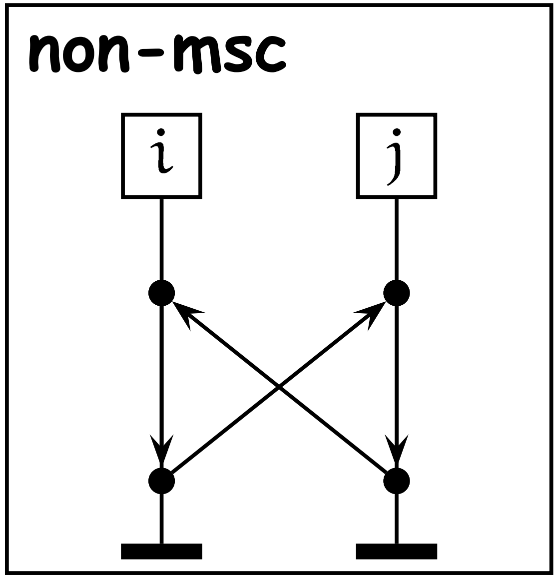
M_3: $\big(\xi(e) = i!^x j \wedge \xi(f) = j?^x i \wedge \#^{i!^x j}(\downarrow e) = \#^{j?^x i}(\downarrow f)\big) \Rightarrow e \preccurlyeq f$

M_4: $[e \longrightarrow\!\!\!\!\prec f \wedge \mathbf{Ins}(e) \neq \mathbf{Ins}(f)]$
$\Rightarrow \big[\xi(e) = i!^x j \wedge \xi(f) = j?^x i \wedge \#^{i!^x j}(\downarrow e) = \#^{j?^x i}(\downarrow f)\big]$.

Other examples

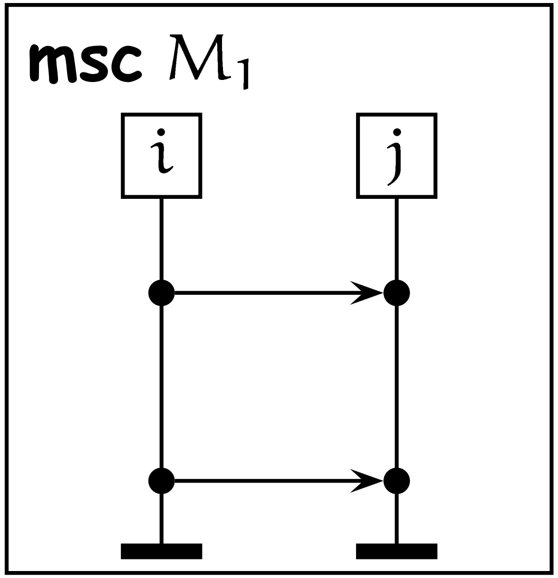


Local total order

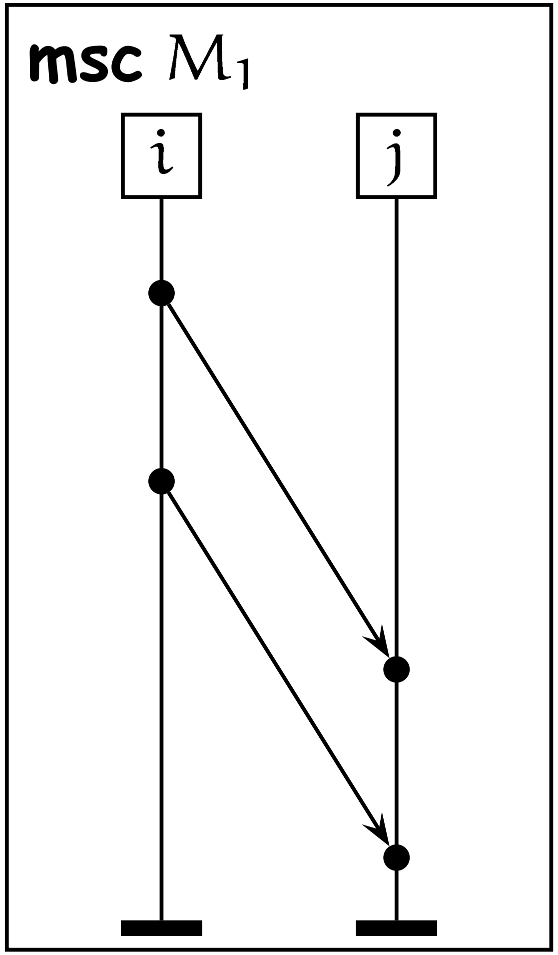


MSCs are pomsets

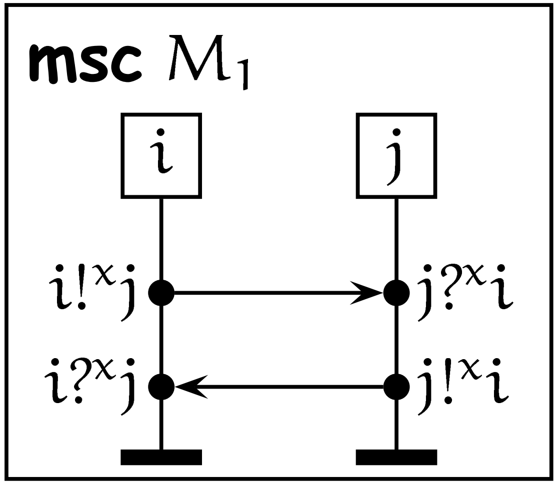
MSCs are pomsets !

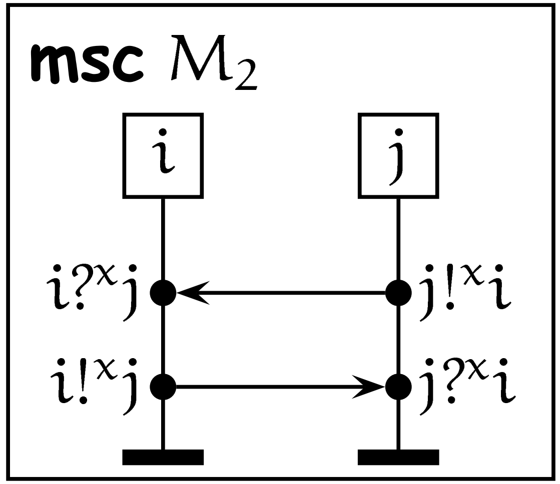


=

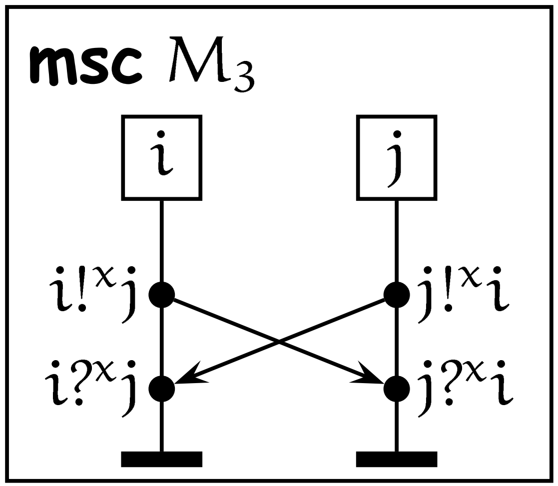


Basic MSCs are not easy to realize!

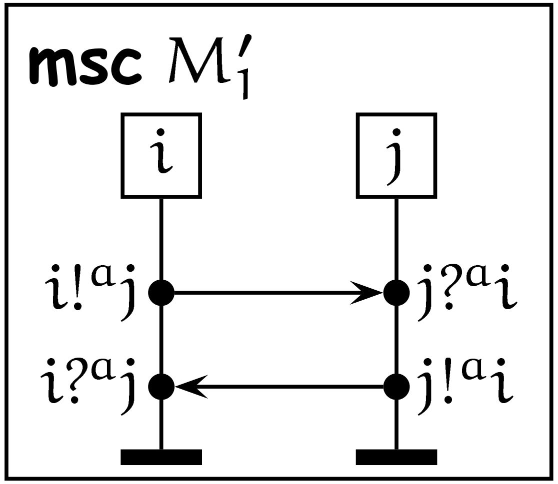


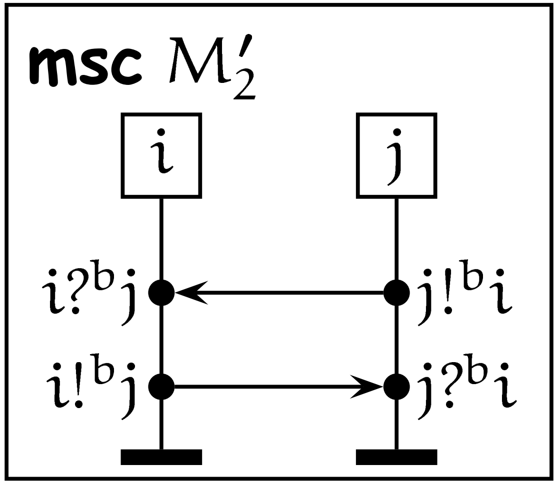


$\{M_1, M_2\}$ is not realizable (with local final states)!



But refinement can help again!

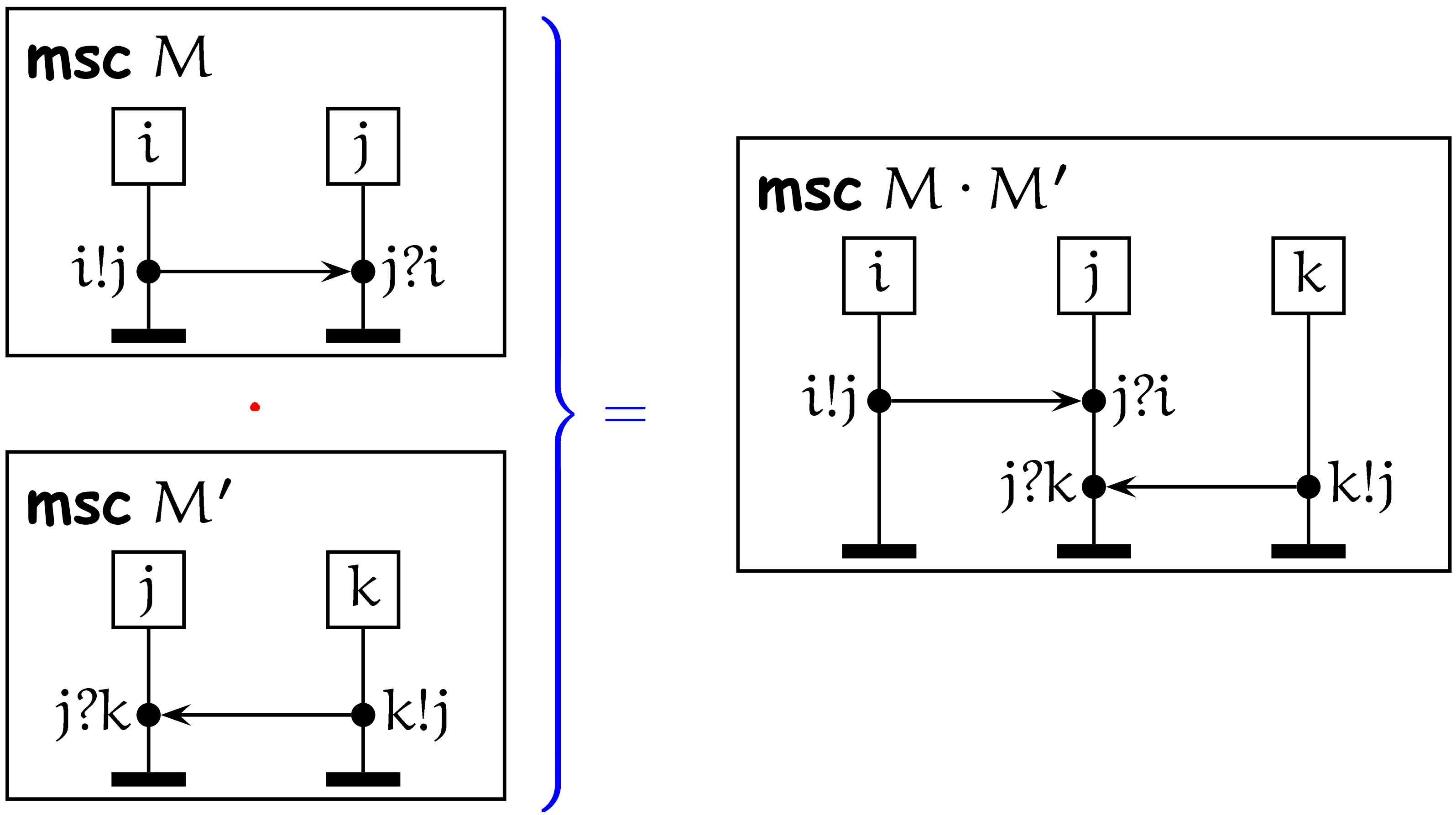




$\{M'_1, M'_2\}$ is realizable up to $a, b \mapsto x$ (with deadlock)

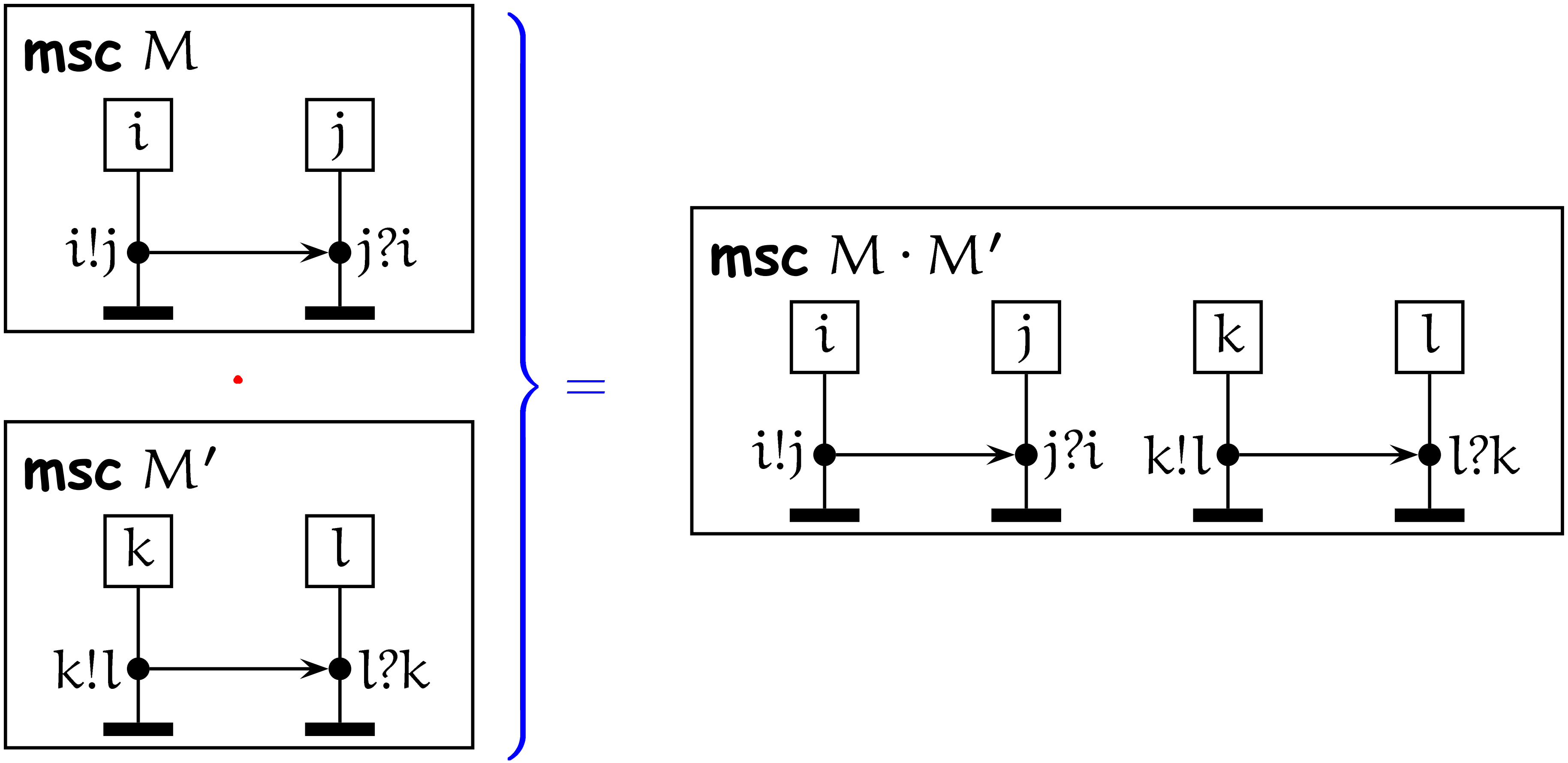
- ✔ *Communicating systems*
- ☛ *High-level MSCs*

Composition of basic MSCs



A HMSC is a rational expression of bMSC.

Parallel composition (for free)



N.B. If $\mathcal{I}_M \cap \mathcal{I}_{M'} = \emptyset$ then $M \cdot M' = M' \cdot M$.

Composition of basic MSCs

The asynchronous concatenation of two basic MSCs $M_1 = (E_1, \preccurlyeq_1, \xi_1)$ and $M_2 = (E_2, \preccurlyeq_2, \xi_2)$ is $M_1 \cdot M_2 = (E, \preccurlyeq, \xi)$ where $E = E_1 \uplus E_2$, $\xi = \xi_1 \cup \xi_2$ and the partial order $\preccurlyeq$ is the transitive closure of

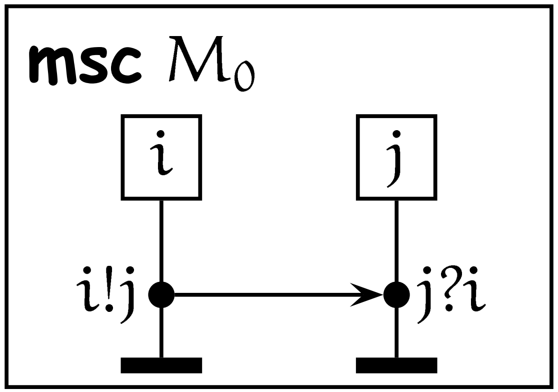
$$\preccurlyeq_1 \cup \preccurlyeq_2 \cup \{(e_1, e_2) \in E_1 \times E_2 \mid \mathbf{Ins}(e_1) = \mathbf{Ins}(e_2)\}.$$

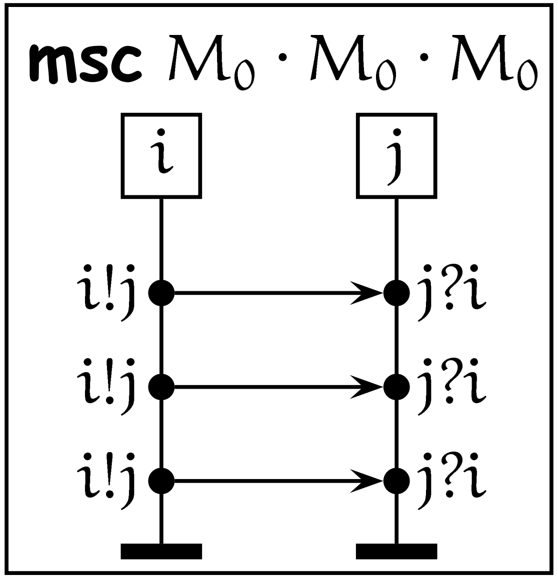
$\{M_0\}^*$ *is not channel-bounded*

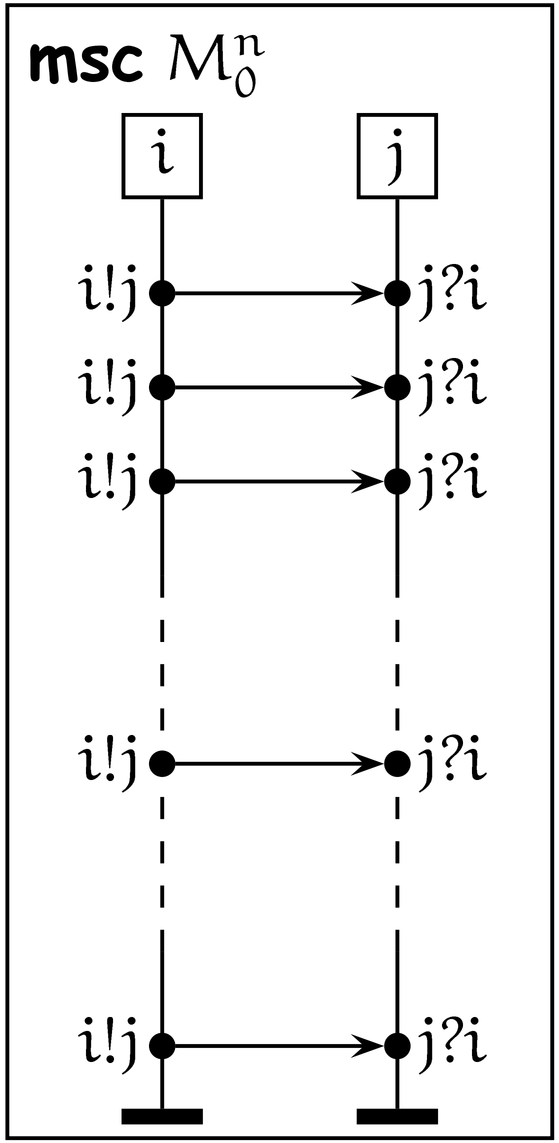
The channel-width of a basic MSC M is

$$\max_{i,j \in \mathcal{I}, i \neq j} \{ |v|_{i!j} - |v|_{j?i} \mid v \leqslant u \in \mathsf{LE}(M) \}.$$

$\mathcal{L} \subseteq \mathsf{bMSC}$ is channel-bounded by $B \in \mathbb{N}$ if each $M \in \mathcal{L}$ has a channel-width $\leqslant B$.

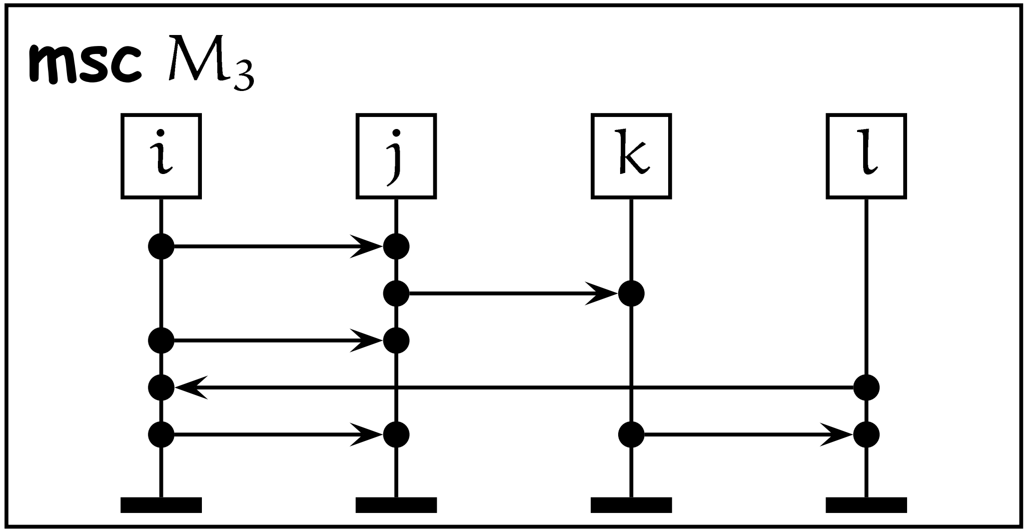


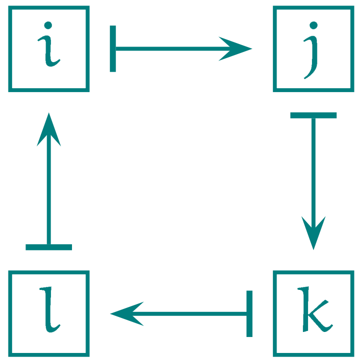




What is the channel-width of M_0^3?

Channel-boundedness is easily decidable





Theorem [Morin01 ???]
A HMSC is channel-bounded if and only if the iteration operation occurs only over sets of locally strongly connected MSCs (each connected component of the communication graph is strongly connected).

We can compute these communication graphs !

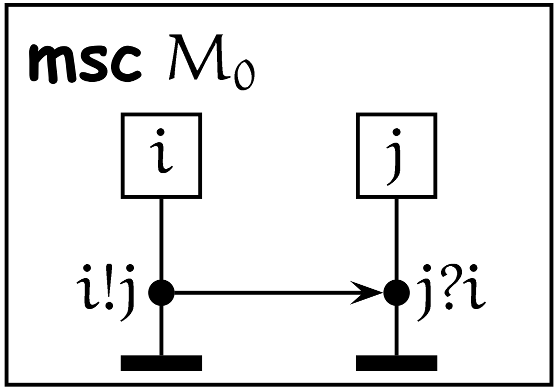
$\{M_0\}^\star$ *is not regular*

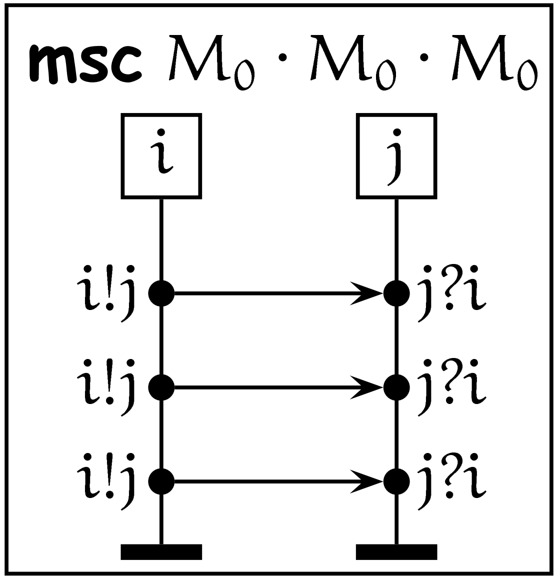
Definition

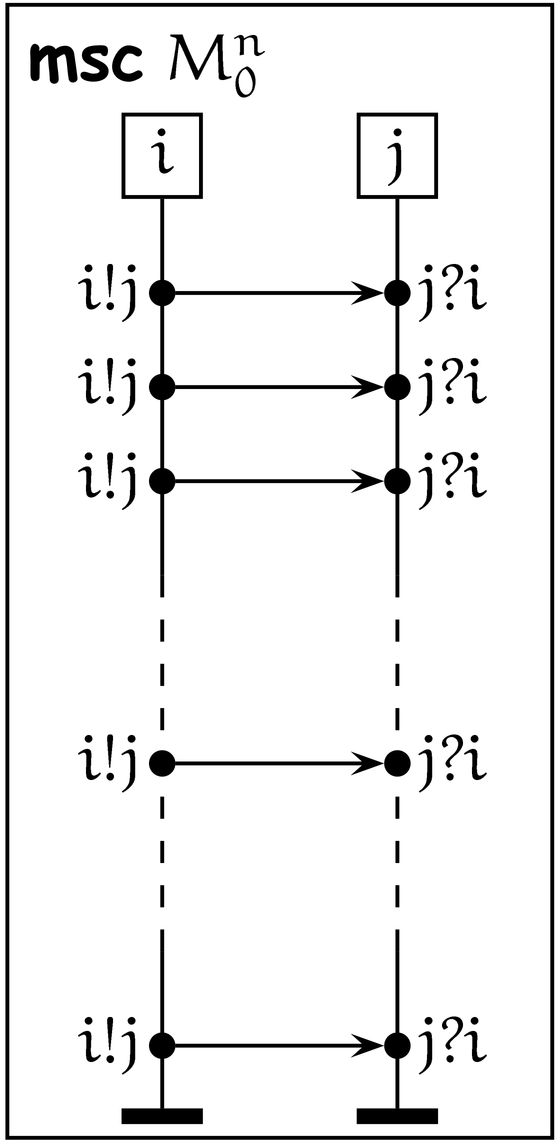
$\mathcal{L} \subseteq$ bMSC is regular [Henriksen et al. 00] if $\bigcup_{M \in \mathcal{L}} LE(M)$ is regular in $\Sigma_{\mathcal{I}}^\star$.

Proposition [Henriksen et al. 00]

Regular $\Rightarrow$ channel-bounded.







Proof of "Regular ⇒ Channel-bounded"

Let $\mathcal{A} = (Q, \imath, \rightarrow, F)$ be a finite automaton recognizing $\mathsf{LE}(\mathcal{L})$.
We have

$$\forall i \neq j \in \mathcal{I}, \forall u \in \Sigma_{\mathcal{I}}^{\star} : \imath \xrightarrow{u} q \in F \Rightarrow |u|_{i!j} = |u|_{j?i}.$$

We may assume

$$\forall q \in Q, \exists u, v \in \Sigma_{\mathcal{I}}^{\star} : \imath \xrightarrow{u} q \xrightarrow{v} q' \in F.$$

Therefore,

$$\forall q \in Q, \forall u_1, u_2 \in \Sigma_{\mathcal{I}}^{\star} : \imath \xrightarrow{u_1} q \wedge \imath \xrightarrow{u_2} q \Rightarrow \forall i \neq j \in \mathcal{I} : |u_1|_{i!j} - |u_1|_{j?i} = |u_2|_{i!j} - |u_2|_{j?i} =: \kappa_{i,j}(q).$$

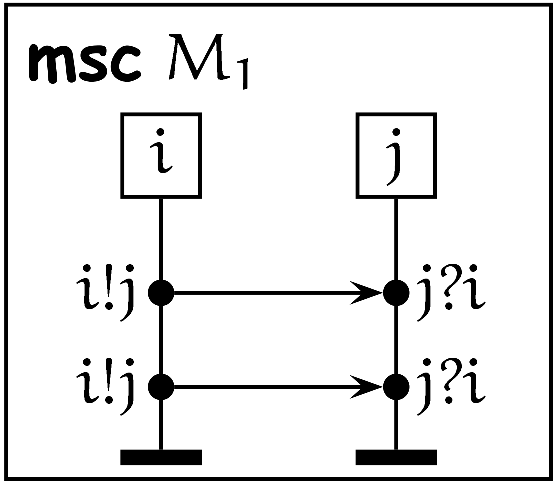
Since Q is finite, there is $B \in \mathbb{N}$ such that

$$\forall q \in Q, \forall i \neq j \in \mathcal{I} : \kappa_{i,j}(q) \leqslant B$$

hence $\forall u \in \mathsf{Pref}(\mathsf{LE}(\mathcal{L})), |u|_{i!j} - |u|_{j?i} \leqslant B$.

- ✔ *Communicating systems*
- ✔ *High-level MSCs*
- ☛ *Connections with Mazurkiewicz traces*

MSCs have a dynamic independence relation



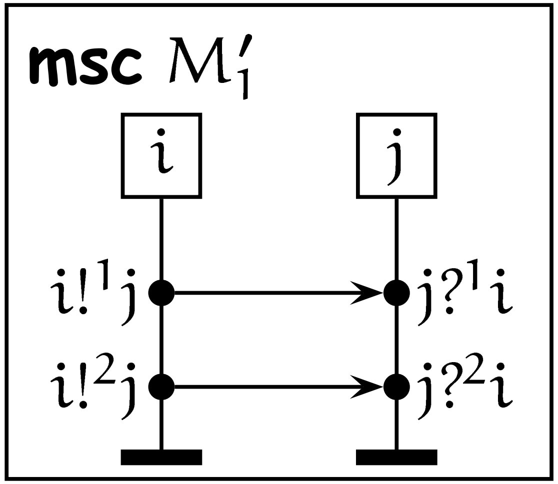
$$i!j.i!j.j?i.j?i \sim i!j.j?i.i!j.j?i$$

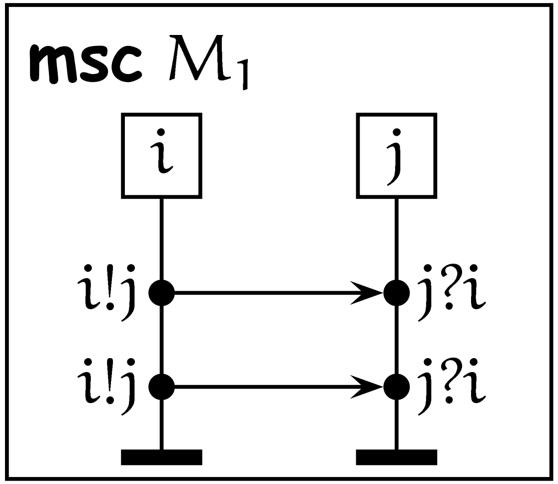
$$(i!j, \{i!j, j?i\}) \in I$$

$$i!j.j?i.i!j.j?i \not\sim i!j.j?i.j?i.i!j$$

$$(i!j.j?i, \{i!j, j?i\}) \notin I$$

But they may be seen as traces up to some refinement!





M_1' is a trace!

$i!^1j, i!^2j \mapsto i!j$

$j?^1i, j?^2i \mapsto j?i$

Refinement of bounded MSCs by traces

Lemma (Kuske's counting lemma)

Let B be a positive integer.
We consider $\Sigma = \Sigma_{\mathcal{I}} \times [0, B-1]$ together with $\| \subseteq \Sigma \times \Sigma$ such that

$$(a, n) \not\| (a', n') \text{ if } \mathbf{Ins}(a) = \mathbf{Ins}(a') \text{ or } [\,\{a, a'\} = \{i!j, j?i\} \wedge n = n'\,].$$

For any basic MSC $M = (E, \preccurlyeq, \xi)$ with channel-width at most B, the pomset $\widehat{\pi_1}(M) := (E, \preccurlyeq, \xi_t)$ such that

$$\xi_t(e) = (\xi(e),\ \#_M^{\xi(e)}(\downarrow e) \quad \mathbf{mod}\ B)$$

is a trace over $(\Sigma, \|)$.

Büchi's theorem for MSCs

Theorem [Henriksen et al.: MFCS 2000]

A language $\mathcal{L} \in \mathrm{b}\mathbb{MSC}$ is regular if, and only if, it is channel-bounded and definable in MSO-logic.

Proof. Assume $\mathcal{L}$ to be regular (and channel-bounded by B).
Then $\widehat{\pi_1}(\mathcal{L})$ is MSO-definable over $\mathbb{M}(\Sigma, \|)$ since $\mathsf{LE}(\widehat{\pi_1}(\mathcal{L}))$ is regular [Thomas 90]. Therefore so is $\mathcal{L}$ as well.
Assume $\mathcal{L}$ to be MSO-definable and channel-bounded by B.
Then $\widehat{\pi_1}(\mathcal{L})$ is MSO-definable, hence $\mathsf{LE}(\widehat{\pi_1}(\mathcal{L}))$ is regular and $\pi_1(\mathsf{LE}(\widehat{\pi_1}(\mathcal{L}))) = \mathsf{LE}(\mathcal{L})$ is regular too.

Useful result for unbounded languages

Two basic MSCs M and M' are independent and we write $M \| M'$ if $\mathcal{I}_M \cap \mathcal{I}_{M'} = \emptyset$. In that case $M \cdot M' = M' \cdot M$.
A basic MSC M is prime if $M = M_1 \cdot M_2 \Rightarrow M_1 = 1 \vee M_2 = 1$.
Any basic MSC is a product of primes.

This decomposition is unique up to permutations of independent primes.

Lemma (Freezing lemma)
Let Γ be a subset of prime bMSCs.
Then

$$\begin{aligned} \mathfrak{R}_\Gamma : \mathbb{M}(\Gamma, \|) &\to \langle \Gamma \rangle_{\mathrm{bMSC}} \\ a \in \Gamma &\mapsto a \in \mathrm{bMSC} \end{aligned}$$

is an isomorphism.

Recognizable sets in monoids

Let $(\mathbb{M}, \cdot)$ be a monoid with unit 1.
A subset $\mathcal{L}$ of $\mathbb{M}$ is recognizable if there exists a finite monoid $\mathbb{M}'$ and a monoid morphism $\eta : \mathbb{M} \to \mathbb{M}'$ such that $\mathcal{L} = \eta^{-1} \circ \eta(\mathcal{L})$.

Equivalently, $\mathcal{L}$ is recognizable if and only if the collection of all sets $\mathcal{L}/x = \{y \in \mathbb{M} \mid x \cdot y \in \mathcal{L}\}$ is finite.

Recognizability is undecidable

Theorem

Recognizability of a HMSC is undecidable.

Proof. For any finite independence alphabet $(\Sigma, \|)$, we can find a finite family of prime bMSCs $\Gamma = (M_a)_{a \in \Sigma}$ such that

$$a \| b \Leftrightarrow \mathcal{I}_{M_a} \cap \mathcal{I}_{M_b} = \emptyset.$$

Then $\psi : \mathbb{M}(\Sigma, \|) \to \langle \Gamma \rangle_{\mathrm{bMSC}}$ defined by $\psi(a) = M_a$ is an isomorphism.

Therefore, $\mathcal{L} \subseteq \mathbb{M}(\Sigma, \|)$ is recognizable iff $\psi(\mathcal{L})$ is recognizable.

EPIT32 — April 26-30, 2004 — CIRM

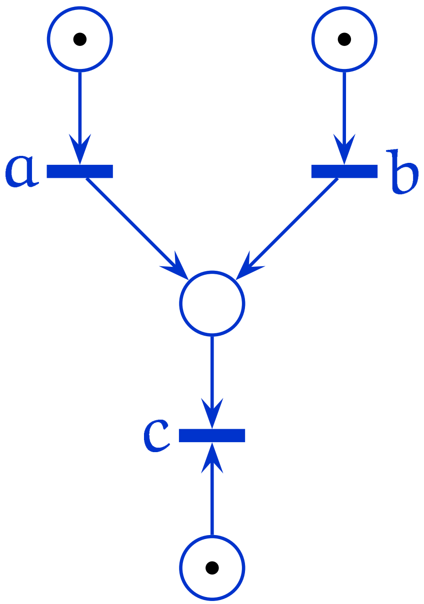
Some generalizations to dynamic traces

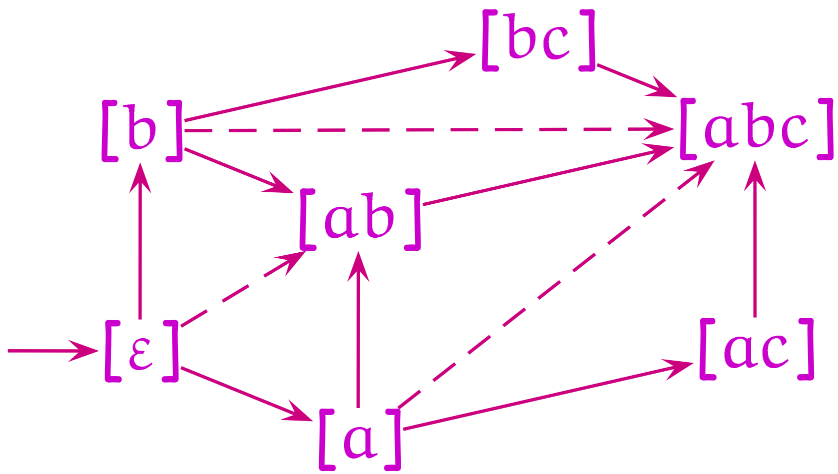
Rémi Morin

May 19, 2004

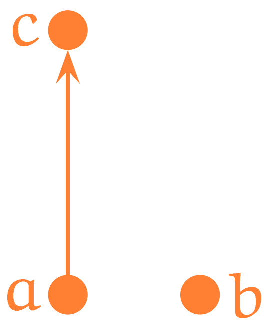
Dynamic independence relations

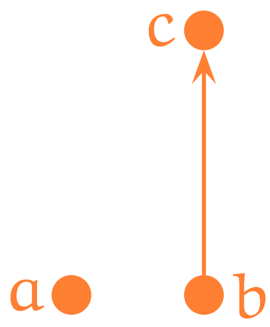
Pomsets for dynamic traces ?





$[abc] = \mathsf{LE}(t)$ *for some pomset* t?





Traces for dynamic independence relations ?

Definition

A *dynamic independence relation* on Σ is a subset I of $\Sigma^\star \times \wp_f(\Sigma)$.
The *trace equivalence* $\sim_I$ is the least equivalence on $\Sigma^\star$ such that

TE$_1$: $\forall u_1, u_2 \in \Sigma^\star, \forall a \in \Sigma, u_1 \sim_I u_2 \Rightarrow u_1.a \sim_I u_2.a$;

TE$_2$: $\forall (u,p) \in I, \forall z_1, z_2 \in \mathsf{Lin}(p), u.z_1 \sim_I u.z_2$.

A dynamic trace $[u]$ is the equivalence class of some word u.

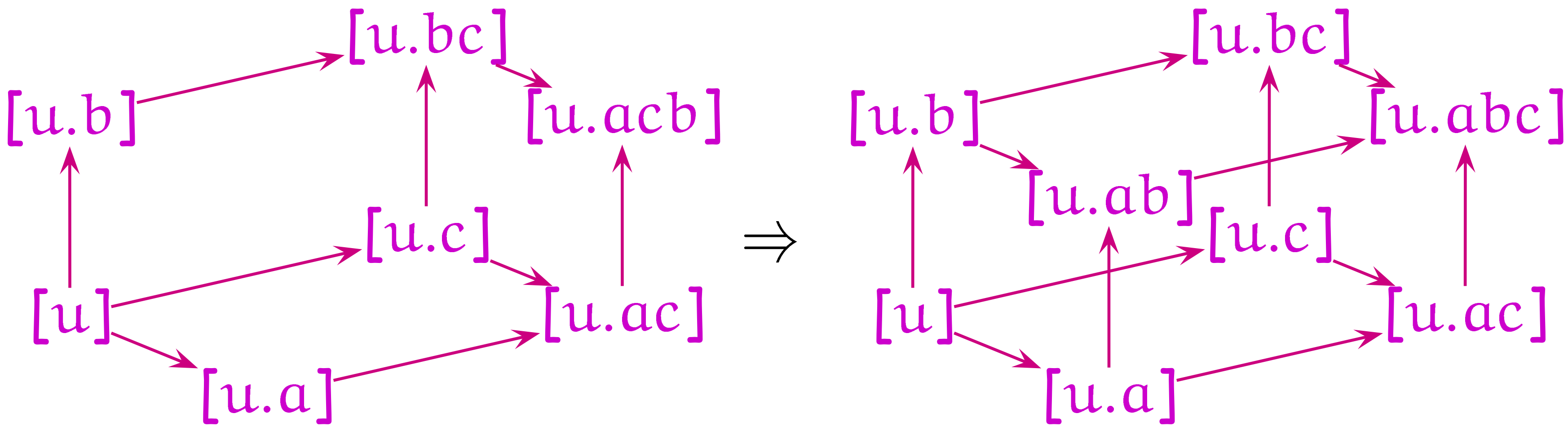
In this talk: $u \sim v \wedge (u,p) \in I$ implies $(v,p) \in I$.

✔ *Dynamic independence relations*

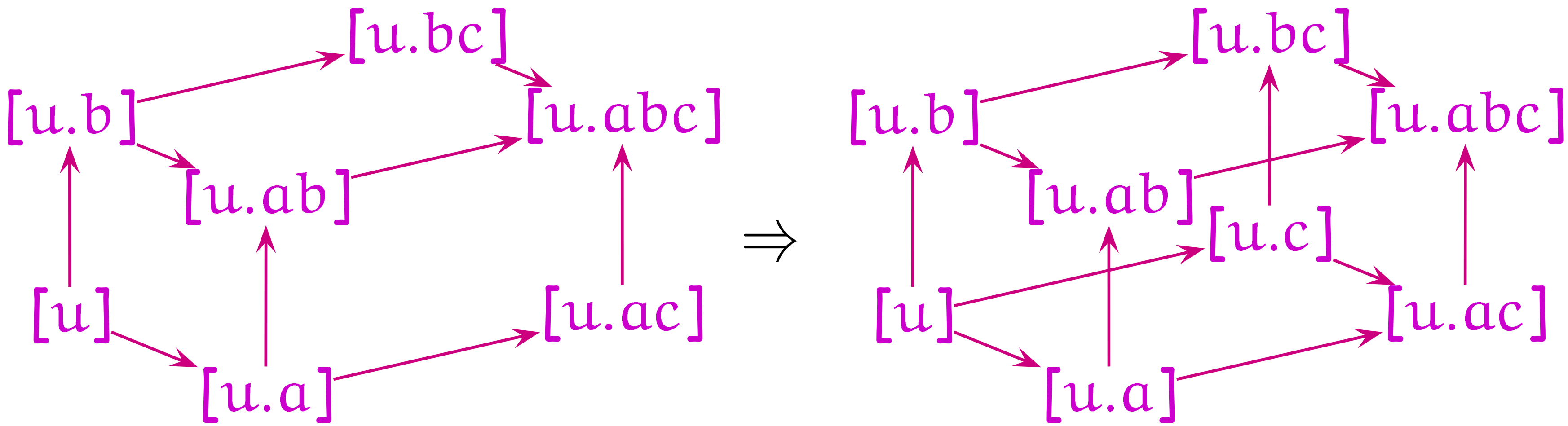
☛ *Dynamic traces as partial orders?*

Cube properties c_1 and c_2

A right-congruence satisfies C_1 if, and only if, for any word $u \in \Sigma^\star$ and all distinct actions $a, b, c \in \Sigma$:



A right-congruence satisfies C_2 if, and only if, for any word $u \in \Sigma^\star$ and all distinct actions $a, b, c \in \Sigma$:



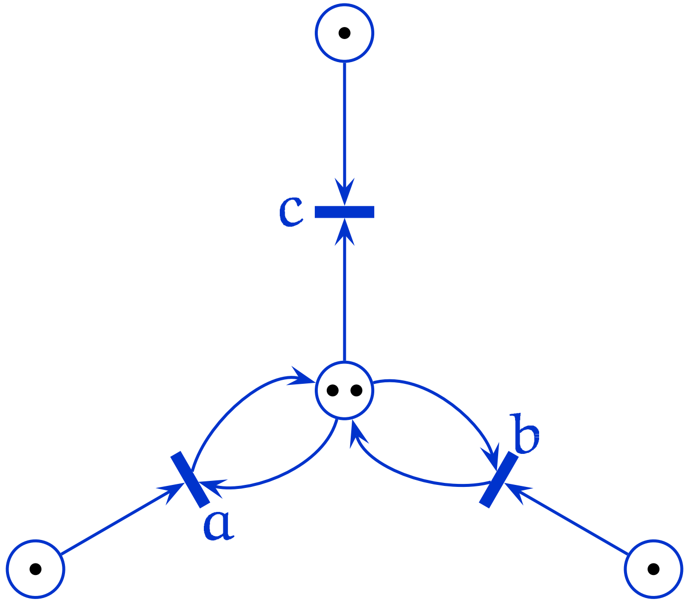
Theorem [Bracho, Droste, Kuske 97]

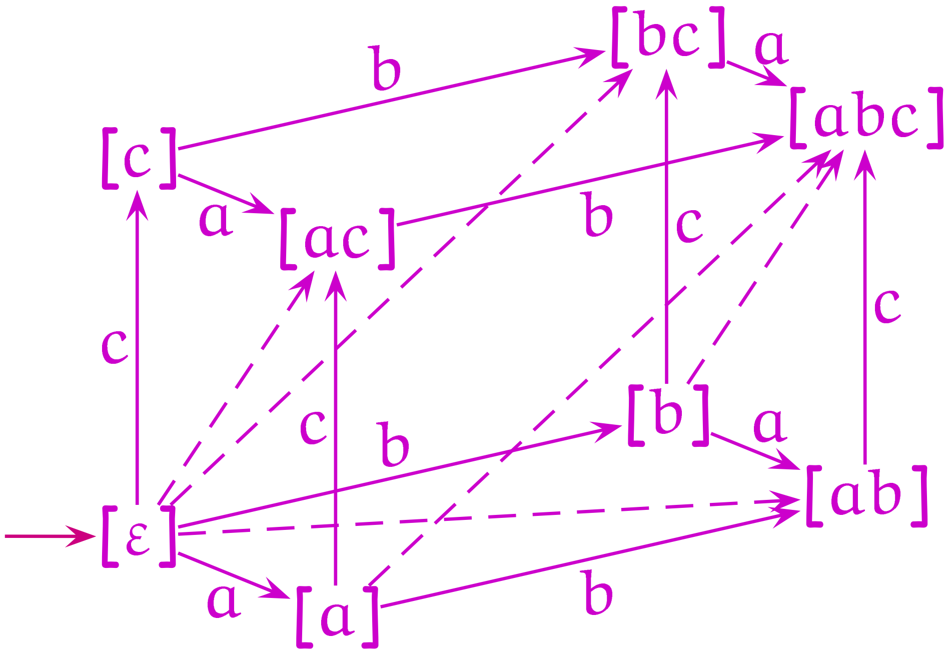
For any dynamic independence relation I, the following conditions are equivalent:

(i) $\sim_{\mathrm{I}}$ satisfies C_1 and C_2.

(ii) $\forall u \in \Sigma^\star, \exists\, !\, t \in \mathbb{P}(\Sigma), [u] = \mathsf{LE}(t)$.

Dynamic traces vs. true concurrency





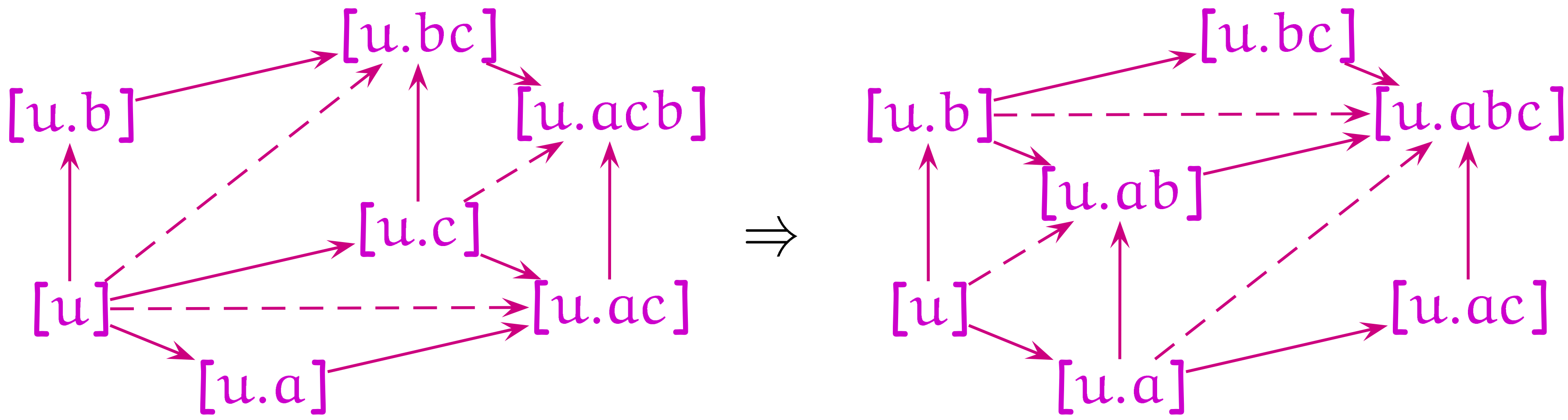
a● ●b ●c

[abc] = **LE**(t) !!!

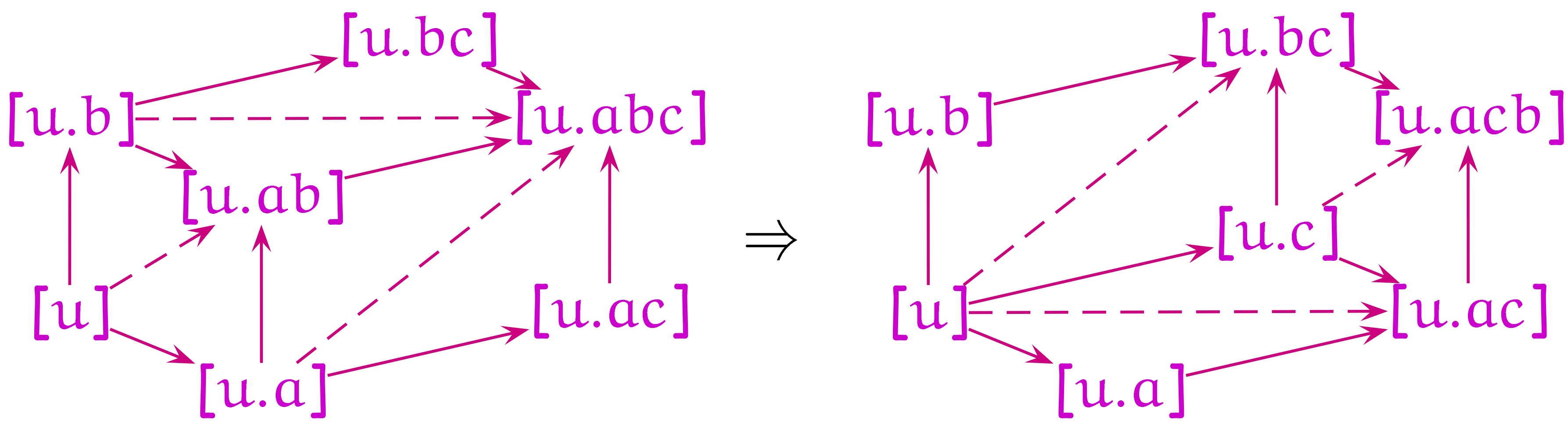
Alternative cube properties

A dynamic independence relation I satisfies the two alternative cube properties [Droste, Kuske] if

AC_1: $\forall u \in \Sigma^*, \forall a, b, c \in \Sigma$ distinct:



AC_2: $\forall u \in \Sigma^*, \forall a, b, c \in \Sigma$ distinct:



Usefull aspects of the alternative cube properties

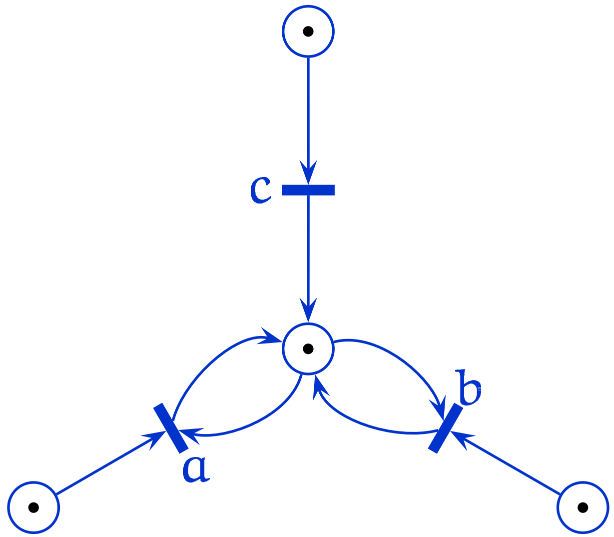
Proposition

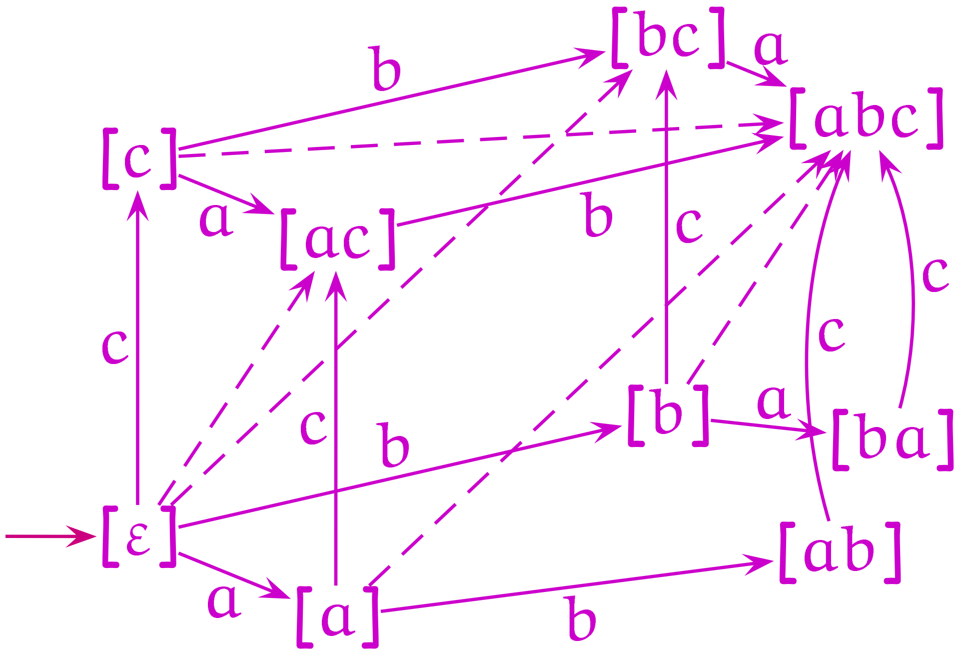
We assume that I is binary ($|p| \leqslant 2$) and satisfies AC_1 **<u>or</u>** AC_2.

1. $\forall u \in \Sigma^\star, \forall a \in \Sigma$: $u.a \sim u'.a \Rightarrow u \sim u'$.
2. $\forall u \in \Sigma^\star, \forall a, b \in \Sigma$: $u.ab \sim u.ba \Rightarrow (u, \{a, b\}) \in \mathrm{I}$.
3. $\forall z, u, u' \in \Sigma^\star$: $z.u \sim z.u' \Rightarrow z.u \sim_z z.u'$.

Note that (2) shows that AC_n implies C_n.

Counter-example for Property 1





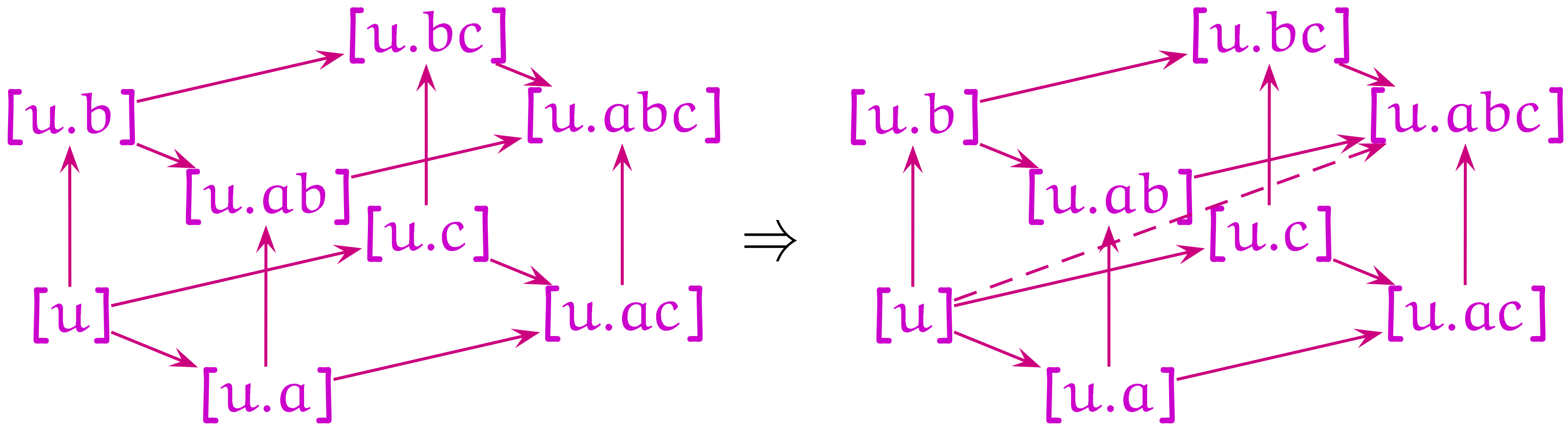
✔ *Dynamic independence relations*

✔ *Dynamic traces as partial orders?*

☛ *Stable independence relations*

Property c_0

A right congruence satisfies C_0 if the following requirement is fulfilled for all $u \in \Sigma^\star$ and all distincts actions $a_1, \ldots, a_n \in \Sigma$:
If $u.a_1 \ldots a_n \sim u.a_{\sigma(1)} \ldots a_{\sigma(n)}$ **for all permutations** $\sigma : [1,n] \to [1,n]$
then $(u, \{a_1, \ldots, a_n\}) \in I$



Definition

A dynamic independence relation I is *stable* if it satisfies the 3 properties C_0, C_1, and C_2.

Stable independence relations are CCI sets of pomsets

Let I be a stable independence relation.
Let $\mathcal{P}_I$ be the pomsets corresponding to all traces. We have

1. $\Sigma^\star = \bigcup_{t \in \mathcal{P}_I} \mathsf{LE}(t)$
2. For all $t_1, t_2 \in \mathcal{P}_I$, $\mathsf{LE}(t_1) \cap \mathsf{LE}(t_2) \neq \emptyset$ implies $t_1 = t_2$.
3. $\mathcal{P}_I$ is prefix-closed.

i.e. $\mathcal{P}_I$ is a CCI set of P-traces [Arnold91].

Lemma

$(u, p) \in I$ if and only if there exists some pomset $t = (E, \preccurlyeq, \xi)$ in $\mathcal{P}_I$ and a downward-closed set $E' \subseteq E$ such that $u \in \mathsf{LE}(E', \preccurlyeq_{|E'}, \xi_{|E'})$ and $p = \xi(\min_{\preccurlyeq}(E \setminus E'))$.

CCI sets of pomsets are stable independence relations

Lemma

Let $\mathcal{P}$ be a CCI set of pomsets. Let I be the dynamic independence relation such that $(u, p) \in I$ if there exists some pomset $t = (E, \preccurlyeq, \xi)$ in $\mathcal{P}$ and a downward-closed set $E' \subseteq E$ such that $u \in \mathbf{LE}(t_{|E'})$ and $p = \xi(\mathbf{min}_{\preccurlyeq}(E \setminus E'))$.
Then I is stable and $\mathcal{P}_I = \mathcal{P}$.

Projections with finite alphabets

Definition

A dynamic independence relation I is *regular* if

- Σ is finite,
- For any $p \in \wp_f(\Sigma)$, $\{u \in \Sigma^\star \mid (u, p) \in \mathrm{I}\}$ is regular.

Theorem [Arnold 91]

Any regular stable (dynamic) independence relation admits a refinement as a regular prefix-closed (static) trace language.

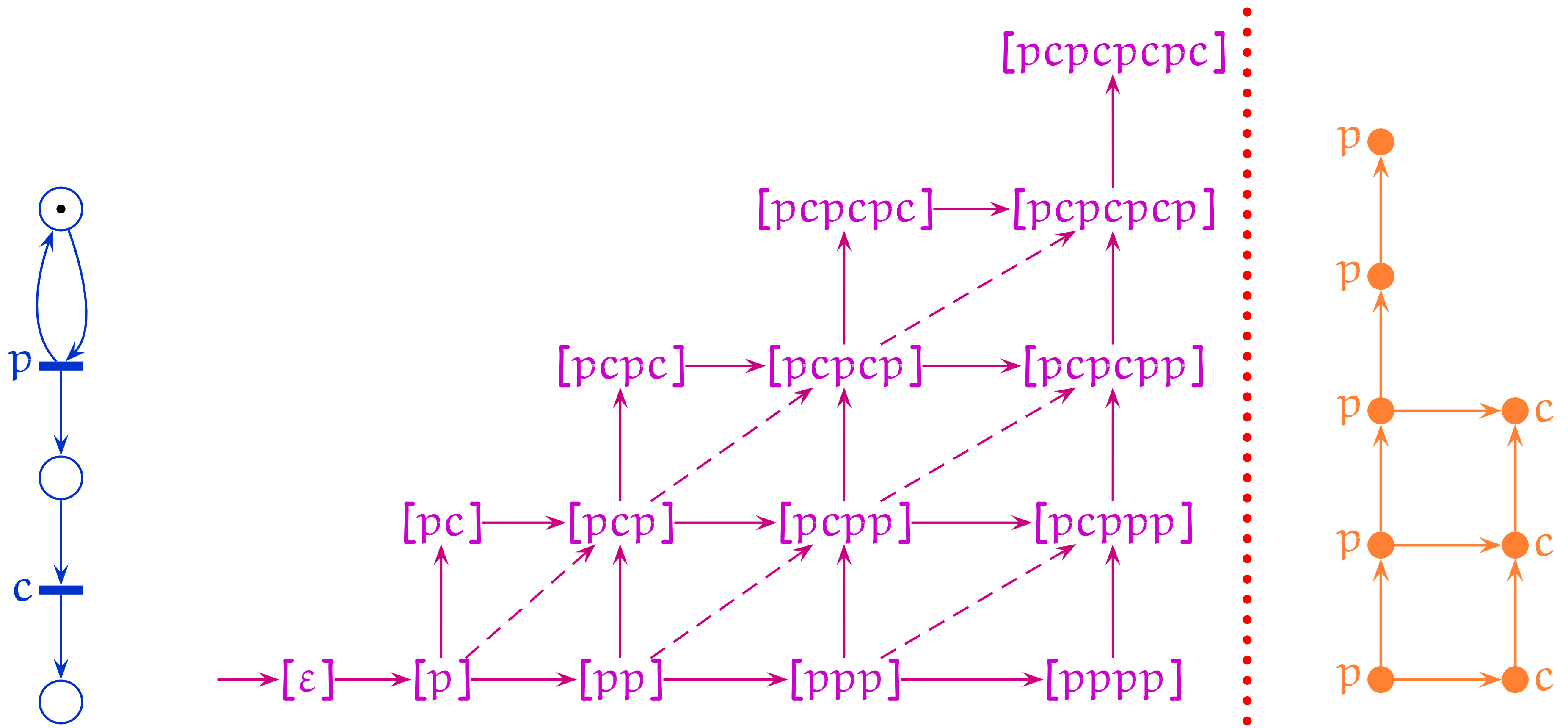
This generalizes Kuske's counting lemma!

Refinements of non-regular stable independencies (1/2)

Theorem [Kuske 97 + Winskel 81]

Any stable (dynamic) independence relation admits a refinement as a prefix-closed (static) trace language (over a possibly infinite alphabet).

Refinements of non-regular stable independencies (2/2)

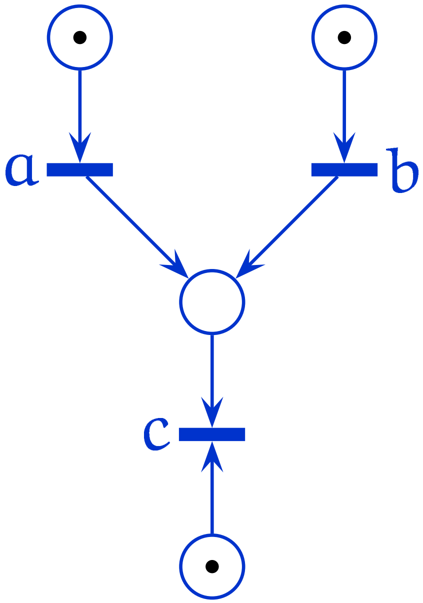


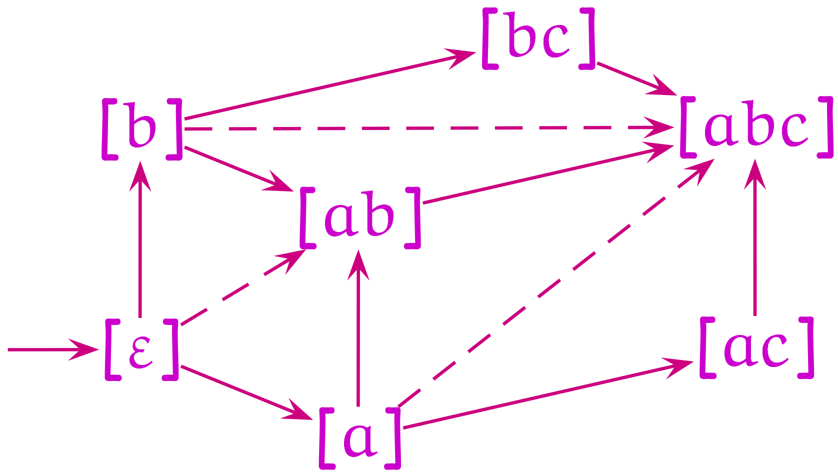
Theorem

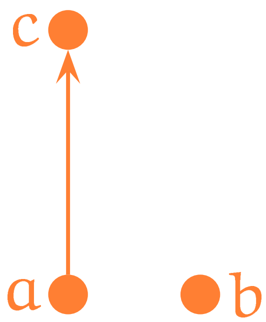
The Producer-Consumer system defines a stable independence relation over a finite alphabet which it admits no refinement as prefix-closed trace language over a finite alphabet.

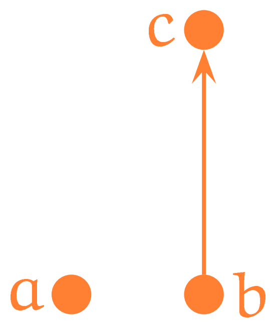
- ✔ *Dynamic independence relations*
- ✔ *Dynamic traces as partial orders?*
- ✔ *Stable independence relations*
- ☛ *Beyond cube axioms*

Pomsets for dynamic traces ?









$$[abc] = \mathsf{LE}(t_1) \cup \mathsf{LE}(t_2)$$

Usual natural restrictions

Definition

A dynamic independence relation I is *complete* if

Cpl$_1$: $(u, p) \in I \wedge p' \subseteq p \Rightarrow (u, p') \in I$;

Cpl$_2$: $(u, p) \in I \wedge p' \subseteq p \wedge v \in \mathsf{Lin}(p') \Rightarrow (u.v, p \setminus p') \in I$;

Cpl$_3$: $(u, \{a, b\}) \in I \wedge (u.ab.v, p) \in I \Rightarrow (u.ba.v, p) \in I$;

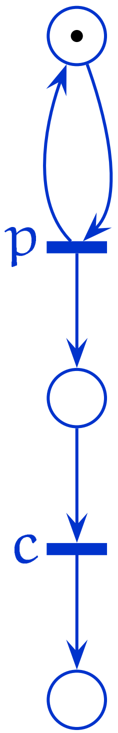
Cpl$_4$: $(u.a, \emptyset) \in I \Rightarrow (u, \{a\}) \in I$.

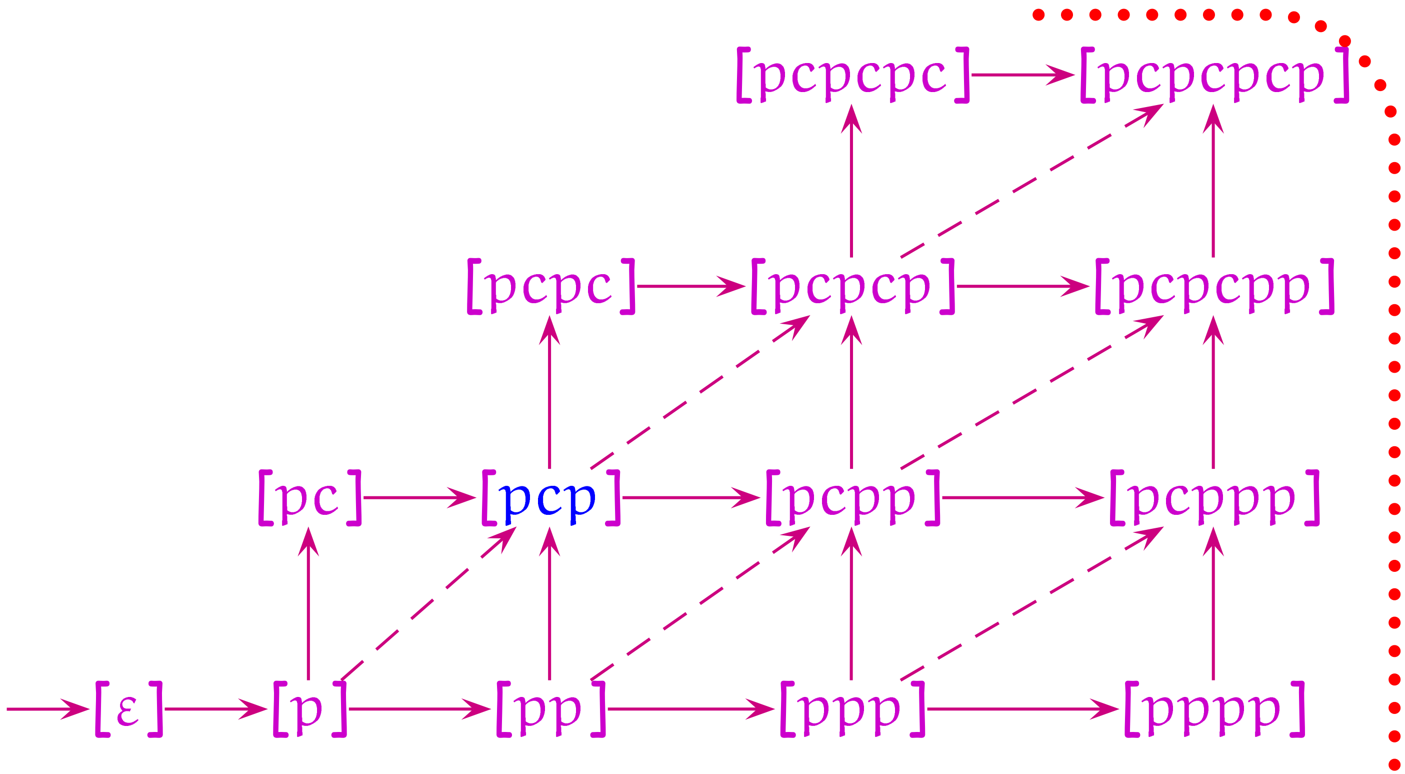
Definition

A *dynamic trace language* over Σ is a structure $\mathcal{L} = (\Sigma, I, L)$ where I is a complete dynamic independence relation on Σ and the set of (prefix-closed) *sequential observations* $L \subseteq \Sigma^\star$ such that

- $u \in L \wedge u \sim_I v \Rightarrow v \in L$
- $u \in L \Leftrightarrow (u, \emptyset) \in I$.

Local trace language (Σ, I, L) *of the Producer-Consumer system*





The alphabet is $\Sigma = \{p, c\}$.

The sequential executions are $L = \{u \in \Sigma^\star \mid \forall v \leqslant u, |v|_p \geqslant |v|_c\}$.

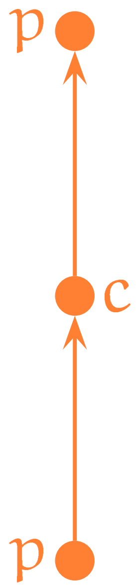
The dynamic independence relation $I \subseteq \Sigma^\star \times \wp_f(\Sigma)$ is such that

$$(u, \{p, c\}) \in I \Leftrightarrow u \in L \wedge |u|_p \geqslant |u|_c + 1.$$

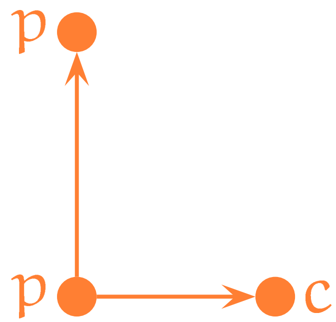
N.B. The Producer-Consumer system is not regular

since $L = \{u \in \Sigma^\star \mid \forall v \leqslant u, |v|_p \geqslant |v|_c\}$ is not regular.

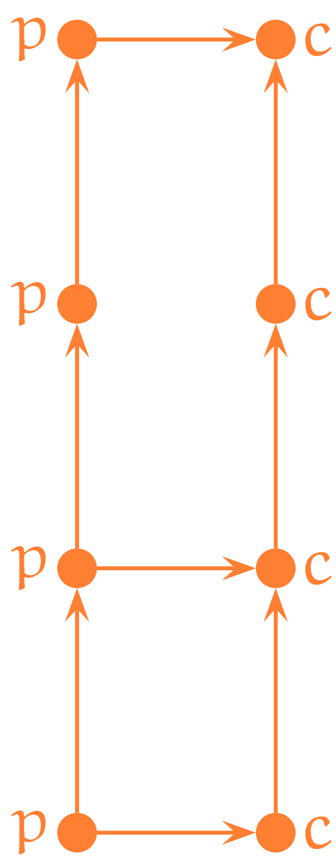
How to describe dynamic traces with pomsets ?



is a process
(but not a proper process)



is a proper process
because $(p, \{p, c\}) \in I$.



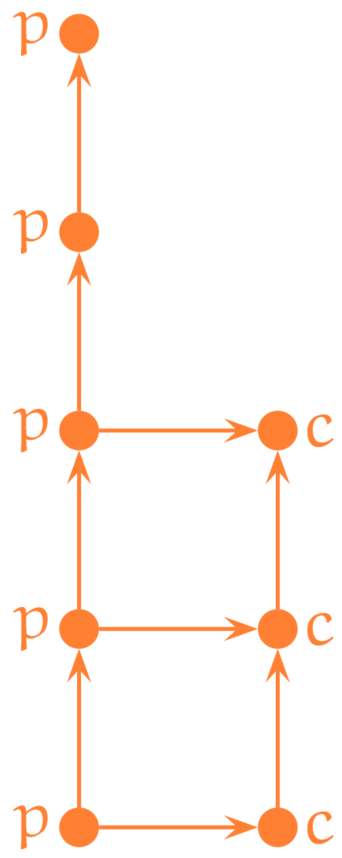
is not a process
since $(pcpc, \{p, c\}) \notin I$.

Processes and proper processes

Definition

A process of $\mathcal{L}$ is a pomset $t = (E, \preccurlyeq, \xi)$ such that for all downward-closed set E' and for all linear order extensions u of $(E', \preccurlyeq_{|E'}, \xi_{|E'})$, we have

$$(u, \xi(\min_{\preccurlyeq}(E \setminus E'))) \in I.$$

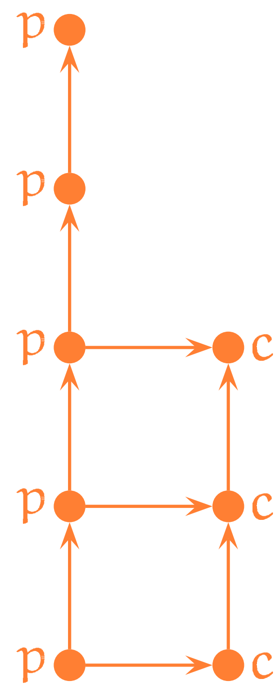


Definition

A proper process of $\mathcal{L}$ is a process which is not an order extension of another process of $\mathcal{L}$.

The proper processes of the Producer-Consumer system are MSO-definable by

$$\forall x, P_p(x) \vee P_c(x)$$
$$\forall x_1, x_2, (P_p(x_1) \wedge P_p(x_2)) \Rightarrow (x_1 \preccurlyeq x_2 \vee x_2 \preccurlyeq x_1)$$
$$\forall x, y, P_p(y) \wedge x \preccurlyeq y \Rightarrow P_p(x)$$
$$\forall y_1, y_2, (P_c(y_1) \wedge P_c(y_2)) \Rightarrow (y_1 \preccurlyeq y_2 \vee y_2 \preccurlyeq y_1)$$
$$\forall y, (P_c(y) \Rightarrow \exists x(P_p(x) \wedge x \longrightarrow\!\!\!\!\prec y))$$
$$\forall x, z, ((P_p(x) \wedge P_c(z) \wedge x \preccurlyeq z) \Rightarrow \exists y, (P_c(y) \wedge x \longrightarrow\!\!\!\!\prec y))$$



Remark

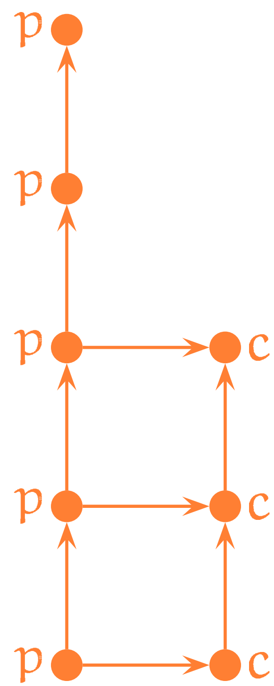
The Producer-Consumer system is definable, but not regular!

What is missing ? The boundedness property

Definition [Kuske 98]

Let $t = (E, \preccurlyeq, \xi)$ be a pomset and $k \in \mathbb{N} \setminus \{0\}$.
A k-chain covering of t is a family $(C_i)_{i \in [1,k]}$ of subsets of E such that

1. each C_i is a chain in $(E, \preccurlyeq)$;
2. $E = \bigcup_{i \in [1,k]} C_i$;
3. if $x \prec\!\!\cdot\, y$ then $\{x, y\} \subseteq C_i$ for some $i \in [1, k]$.



A class of processes $\mathcal{P}$ is bounded if there exists $k \in \mathbb{N} \setminus \{0\}$ such that each process of $\mathcal{P}$ admits a k-chain covering.

Remark

The Producer-Consumer system is not bounded.

Büchi's theorem for dynamic trace languages!

Theorem [Kuske Morin 00]

Let $\mathcal{L} = (\Sigma, I, L)$ be a dynamic trace language with Σ finite.
Let $\mathfrak{p}(\mathcal{L})$ denote the class of all proper processes of $\mathcal{L}$.
$\mathcal{L}$ is regular if and only if $\mathfrak{p}(\mathcal{L})$ is MSO-definable *and bounded*.

"Regular $\Rightarrow$ bounded" relies on Ramsey's theorem.

Ramsey's theorem

Let S be a finite set. Then there is $R(S) \in \mathbb{N}$ such that for any mapping d of the two-elements subsets of $\{1, 2, ..., R(S)\}$ into S, there exist three distinct elements $i, j, k \in \{1, 2, ..., R(S)\}$ such that $d(B) = d(C)$ for any two-elements subsets B and C of $\{i, j, k\}$.

Questions?

Thank you!