Discrete and Continuous Models for Concurrent Systems

2. Concurrent Semantics of Petri Nets

Uli Fahrenberg

LMF, Université Paris-Saclay, France

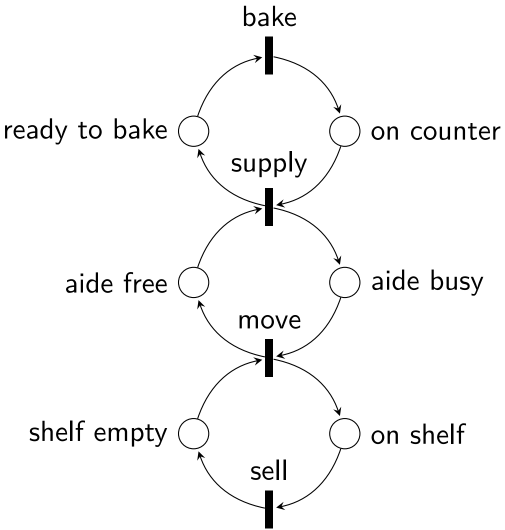
ISIMM, April 2026

Petri Nets

A Petri net (S, T, F):

- S set of places
- T set of transitions, $S \cap T = \emptyset$
- $F \subseteq S \times T \cup T \times S$ flow relation

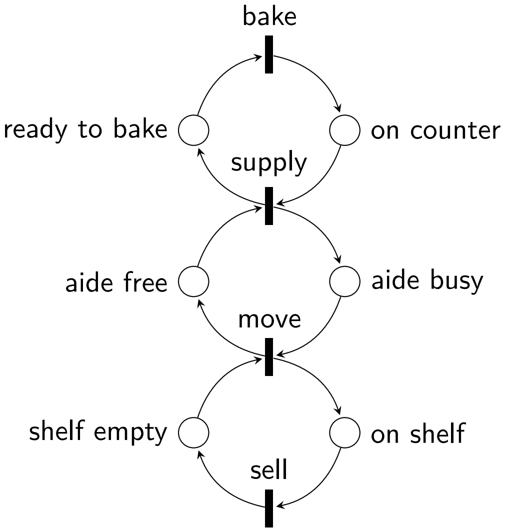
- very useful for modeling distributed or concurrent systems
- invented in 1962; ubiquitous in modeling



Petri Nets

A Petri net (S, T, F):

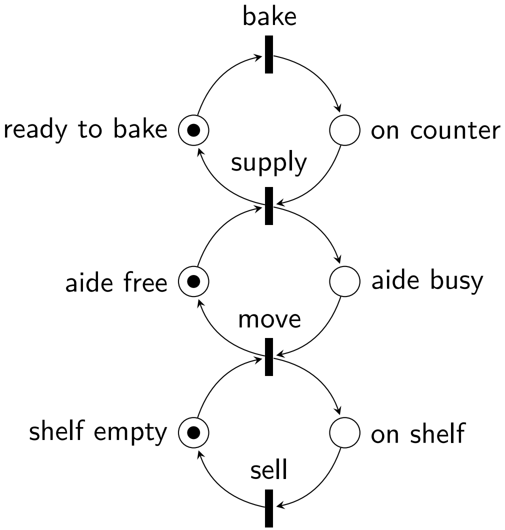
- S set of places
- T set of transitions, $S \cap T = \emptyset$
- $F : S \times T \cup T \times S \to \mathbb{N}$
  weighted flow relation



Petri Nets

A Petri net (S, T, F):

- S set of places
- T set of transitions, $S \cap T = \emptyset$
- $F : S \times T \cup T \times S \to \mathbb{N}$
  weighted flow relation
- marking: $S \to \mathbb{N}$:
  number of tokens per place
- preset of t: ${}^\bullet t = F(s, t)$
- postset of t: $t^\bullet(s) = F(t, s)$



Petri Nets

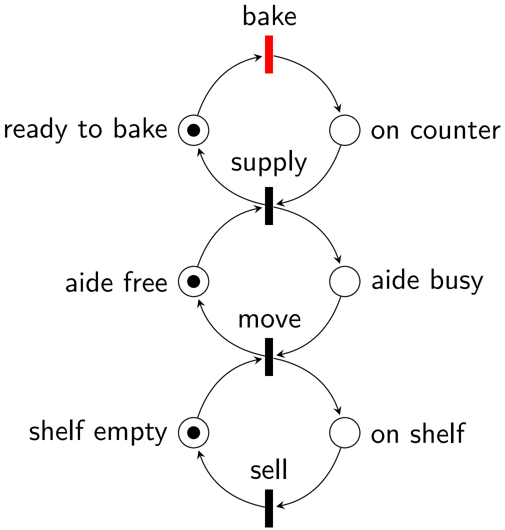
A Petri net (S, T, F):

- S set of places
- T set of transitions, $S \cap T = \emptyset$
- $F : S \times T \cup T \times S \to \mathbb{N}$
  weighted flow relation

- marking: $S \to \mathbb{N}$:
  number of tokens per place
- preset of t: ${}^\bullet t = F(s, t)$
- postset of t: $t^\bullet(s) = F(t, s)$

- compute by transforming markings:
  $$m' = m - {}^\bullet t + t^\bullet$$

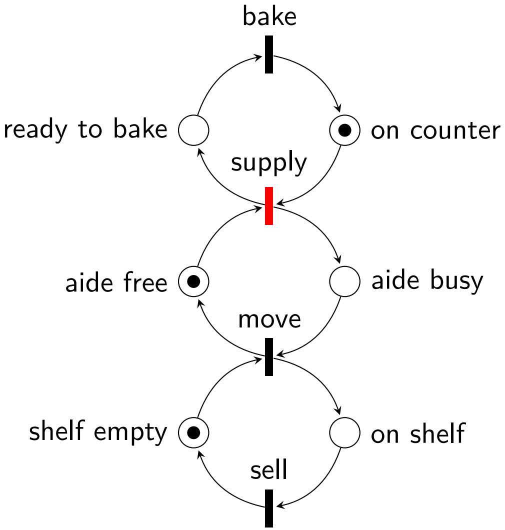
- only if ${}^\bullet t \leq m$



Petri Nets

A Petri net (S, T, F):

- S set of places
- T set of transitions, $S \cap T = \emptyset$
- $F : S \times T \cup T \times S \to \mathbb{N}$
  weighted flow relation
- marking: $S \to \mathbb{N}$:
  number of tokens per place
- preset of t: ${}^\bullet t = F(s, t)$
- postset of t: $t^\bullet(s) = F(t, s)$
- compute by transforming markings:
  $$m' = m - {}^\bullet t + t^\bullet$$
- only if ${}^\bullet t \leq m$



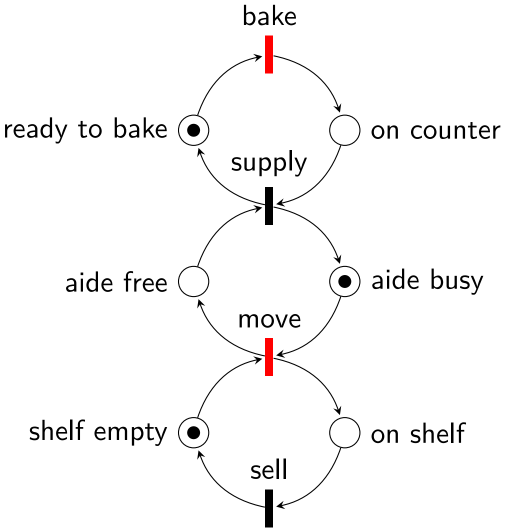
Petri Nets

A Petri net (S, T, F):

- S set of places
- T set of transitions, $S \cap T = \emptyset$
- $F : S \times T \cup T \times S \to \mathbb{N}$
  weighted flow relation

- marking: $S \to \mathbb{N}$:
  number of tokens per place
- preset of t: ${}^\bullet t = F(s, t)$
- postset of t: $t^\bullet(s) = F(t, s)$

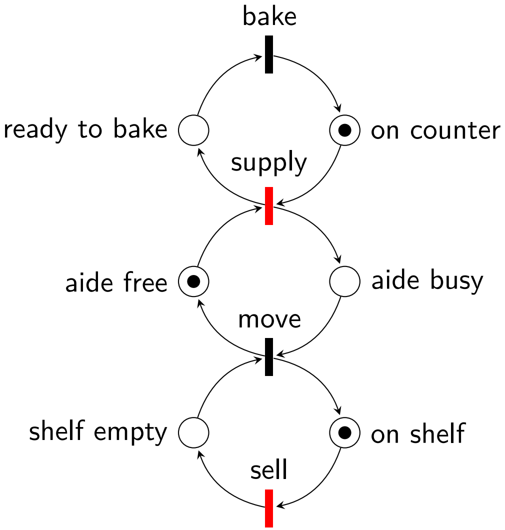
- compute by transforming markings:
  $$m' = m - {}^\bullet t + t^\bullet$$
- only if ${}^\bullet t \le m$



Petri Nets

A Petri net (S, T, F):

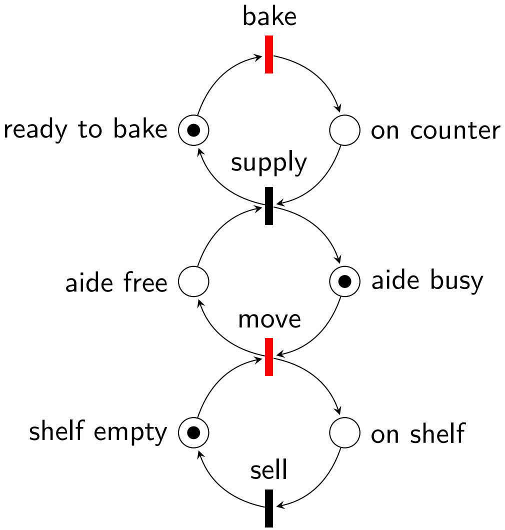
- S set of places
- T set of transitions, $S \cap T = \emptyset$
- $F : S \times T \cup T \times S \to \mathbb{N}$
  weighted flow relation
- marking: $S \to \mathbb{N}$:
  number of tokens per place
- preset of t: ${}^\bullet t = F(s, t)$
- postset of t: $t^\bullet(s) = F(t, s)$
- compute by transforming markings:
  $$m' = m - {}^\bullet t + t^\bullet$$
- only if ${}^\bullet t \leq m$



Petri Nets

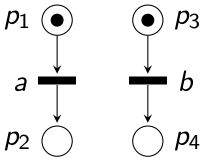
A Petri net (S, T, F):

- S set of places
- T set of transitions, $S \cap T = \emptyset$
- $F : S \times T \cup T \times S \to \mathbb{N}$
  weighted flow relation
- marking: $S \to \mathbb{N}$:
  number of tokens per place
- preset of t: ${}^\bullet t = F(s, t)$
- postset of t: $t^\bullet(s) = F(t, s)$
- compute by transforming markings:
  $$m' = m - {}^\bullet t + t^\bullet$$
- only if ${}^\bullet t \leq m$



Semantics of Petri Nets

Petri net (S, T, F): places S; transitions T;
weighted flows $F : S \times T \cup T \times S \to \mathbb{N}$

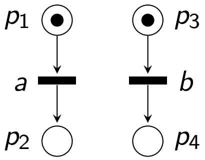


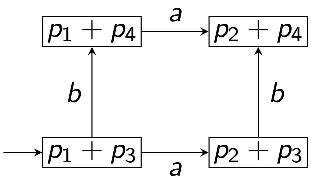
Semantics of Petri Nets

Petri net (S, T, F): places S; transitions T; weighted flows $F : S \times T \cup T \times S \to \mathbb{N}$

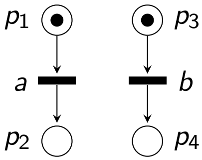
Interleaved semantics (reachability graph) (V, E):

- $V = \mathbb{N}^S$: all markings
- $E \subseteq V \times T \times V$: one transition at a time
- $E = \{(m, t, m') \mid {}^\bullet t \leq m, m' = m - {}^\bullet t + t^\bullet\}$
- initial marking $\Longrightarrow$ initial state; take reachable part



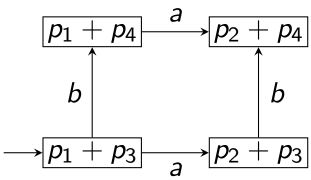


Semantics of Petri Nets



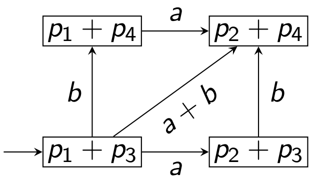
Petri net (S, T, F): places S; transitions T; weighted flows $F : S \times T \cup T \times S \to \mathbb{N}$

Interleaved semantics (reachability graph) (V, E):



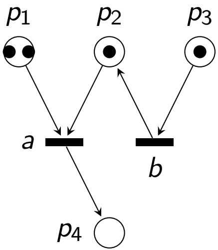
- $V = \mathbb{N}^S$: all markings
- $E \subseteq V \times T \times V$: one transition at a time
- $E = \{(m, t, m') \mid {}^\bullet t \le m, m' = m - {}^\bullet t + t^\bullet\}$
- initial marking $\implies$ initial state; take reachable part

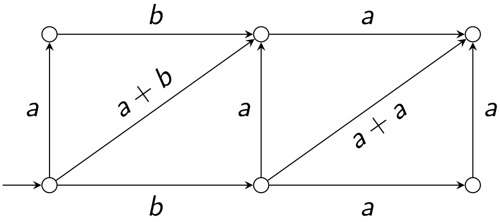
Concurrent step reachability graph (V, E'):



- $V = \mathbb{N}^S$
- $E' \subseteq V \times \mathbb{N}^T \times V$: multisets of transitions
- $E' = \{(m, U, m') \mid {}^\bullet U \le m, m' = m - {}^\bullet U + U^\bullet\}$

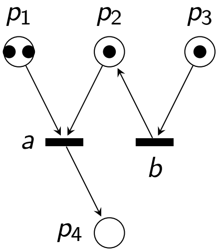
Another Example

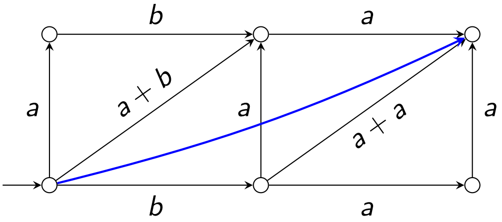




- after firing b, a is auto-concurrent

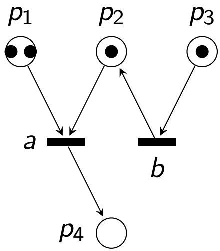
Another Example

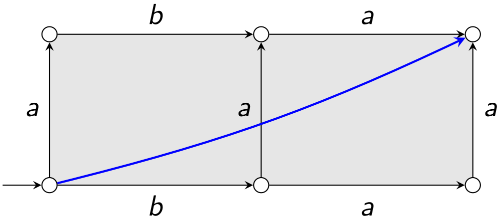




- after firing b, a is auto-concurrent
- semantics misses some behavoir?
  - start a – start b – finish b – start another a – etc.

Another Example





- after firing b, a is auto-concurrent
- semantics misses some behavoir?
  - start a – start b – finish b – start another a – etc.
- enter higher-dimensional automata
  - replace multi-transitions by squares

Higher-Dimensional Automata

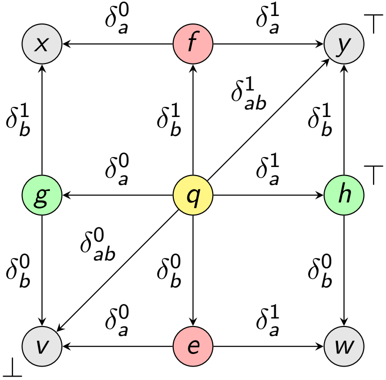
A conclist is a finite, totally ordered, Σ-labeled set. (a list of labeled events)

A precubical set X consists of:

- A set of cells X (cubes)
- Every cell $x \in X$ has a conclist $\text{ev}(x)$ (list of events active in x)
- We write $X[U] = \{x \in X \mid \text{ev}(x) = U\}$ for a conclist U (cells of type U)
- For every conclist U and $A \subseteq U$ there are:
  upper face map $\delta_A^1 : X[U] \to X[U \setminus A]$ (terminating events A)
  lower face map $\delta_A^0 : X[U] \to X[U \setminus A]$ ("unstarting" events A)
- Precube identities: $\delta_A^\mu \delta_B^\nu = \delta_B^\nu \delta_A^\mu$ for $A \cap B = \emptyset$ and $\mu, \nu \in \{0, 1\}$

A higher dimensional automaton (HDA) is a precubical set X with initial cells $\bot \subseteq X$ and accepting cells $\top \subseteq X$ (not necessarily vertices)

Example



$X[\emptyset] = \{v, w, x, y\}$

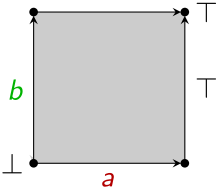
$X[a] = \{e, f\}$

$X[b] = \{g, h\}$

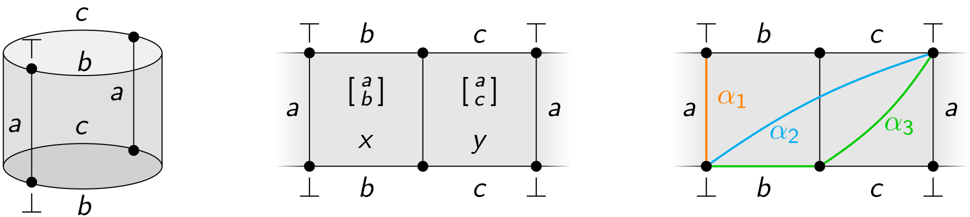
$X[\left[\begin{smallmatrix} a \\ b \end{smallmatrix}\right]] = \{q\}$

$\bot_X = \{v\}$

$\top_X = \{h, y\}$

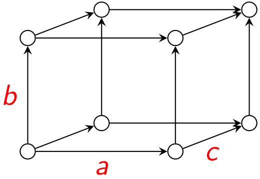


Another One

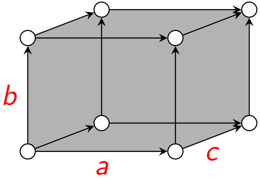


$$a \parallel (bc)^*$$

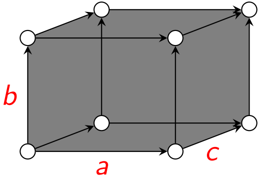
More Examples



no concurrency



two out of three



full concurrency

Higher-Dimensional Automata & Concurrency Theory

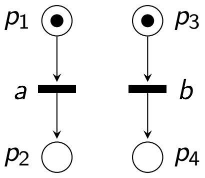
HDAs as a model for concurrency:

- points: states
- edges: transitions
- squares, cubes etc.: independency relations / concurrently executing events
- two-dimensional automata $\cong$ asynchronous transition systems

- Introduced in 1990
- Generalize all main models of concurrency proposed in the literature
- (event structures; Petri nets; communicating automata; etc.)

Concurrent Semantics of Petri Nets

Petri net (S, T, F): places S; transitions T;
weighted flows $F : S \times T \cup T \times S \to \mathbb{N}$



Interleaved semantics (V, E): $V = \mathbb{N}^S$; $E \subseteq V \times T \times V$

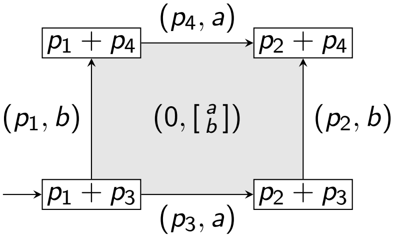
- $E = \{(m, t, m') \mid {}^\bullet t \le m, m' = m - {}^\bullet t + t^\bullet\}$

Concurrent semantics as HDA:
$\square = \square(T)$, $X = \mathbb{N}^S \times \square$, $\mathrm{ev}(m, \tau) = \tau$

- for $x = (m, \tau) \in X[\tau]$ with $\tau = (t_1, \dots, t_n)$:

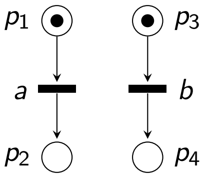
$$\delta^0_{t_i}(x) = (m + {}^\bullet t_i, (t_1, \dots, t_{i-1}, t_{i+1}, \dots, t_n))$$
$$\delta^1_{t_i}(x) = (m + t_i^\bullet, (t_1, \dots, t_{i-1}, t_{i+1}, \dots, t_n))$$

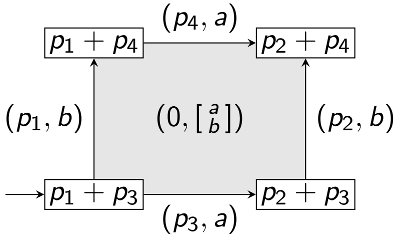
- initial marking $\Longrightarrow$ initial cell; take reachable part
- (no accepting cells)



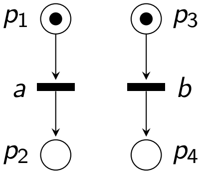
Event Order

- trouble with symmetry:
  have a cell $(0, \left[\begin{smallmatrix} a \\ b \end{smallmatrix}\right])$, but also $(0, \left[\begin{smallmatrix} b \\ a \end{smallmatrix}\right])$ (not shown)
- solution: fix an arbitrary order $\preccurlyeq$ on T
- and use $\square = \left\{ \left[\begin{smallmatrix} t_1 \\ \vdots \\ t_n \end{smallmatrix}\right] \;\middle|\; \forall i = 1, \dots, n-1 : t_i \preccurlyeq t_{i+1} \right\}$
  instead of $\square(T)$
- order $\preccurlyeq$ may be chosen (and re-chosen) at will
- here: lexicographic $a \prec b \prec \dots$

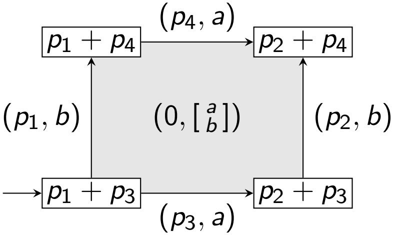


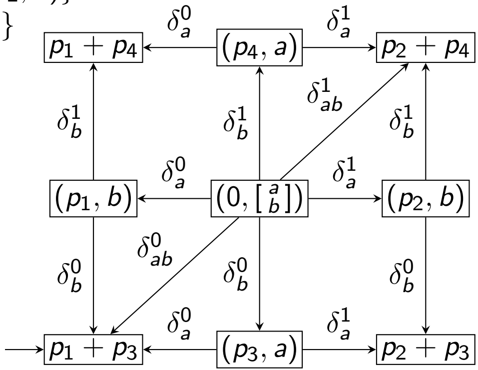


Example, Complete

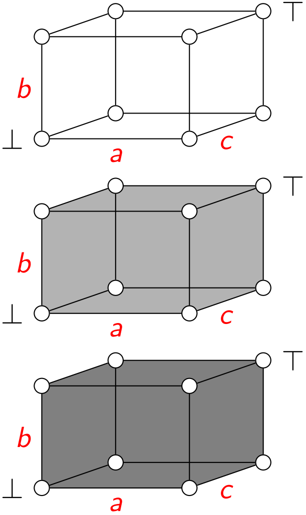


$$X[\emptyset] = \{p_1 + p_3, p_2 + p_3, p_1 + p_4, p_2 + p_4\}$$
$$X[a] = \{(p_3, a), (p_4, a)\}$$
$$X[b] = \{(p_1, b), (p_2, b)\}$$
$$X[[\,{}^{a}_{b}\,]] = \{(0, [\,{}^{a}_{b}\,])\}$$





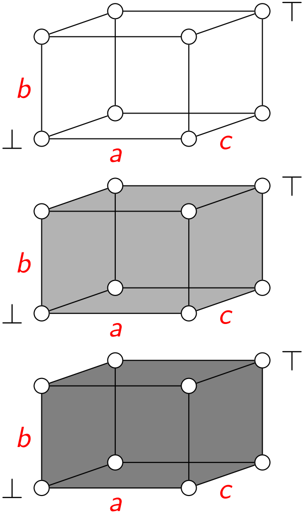
Languages of HDAs: Examples



$$L_1 = \{abc, acb, bac, bca, cab, cba\}$$

$$L_3 = \left\{ \begin{bmatrix} a \\ b \\ c \end{bmatrix}, \dots \right\}$$

Languages of HDAs: Examples



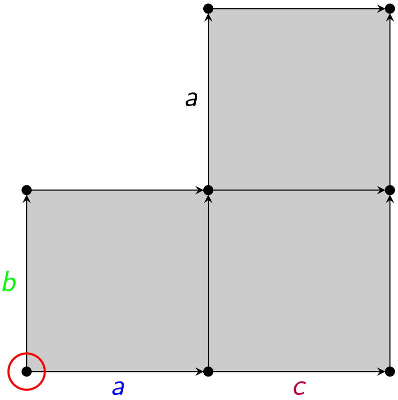
$$L_1 = \{abc, acb, bac, bca, cab, cba\}$$

$$L_2 = \left\{ \begin{bmatrix} a \\ b \to c \end{bmatrix}, \begin{bmatrix} a \\ c \to b \end{bmatrix}, \begin{bmatrix} b \\ a \to c \end{bmatrix}, \begin{bmatrix} b \\ c \to a \end{bmatrix}, \begin{bmatrix} c \\ a \to b \end{bmatrix}, \begin{bmatrix} c \\ b \to a \end{bmatrix} \right\} \cup L_1 \cup \dots$$

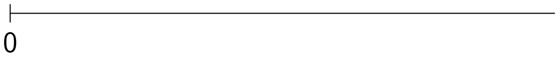
sets of pomsets

$$L_3 = \left\{ \begin{bmatrix} a \\ b \\ c \end{bmatrix} \right\} \cup L_2$$

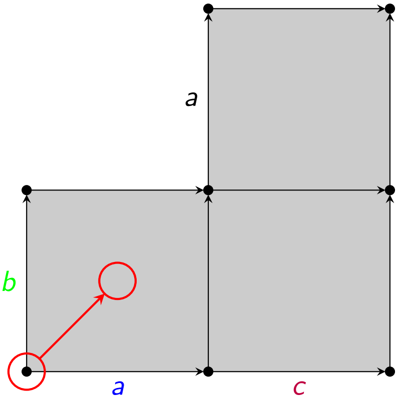
Event Ipomset of a Path

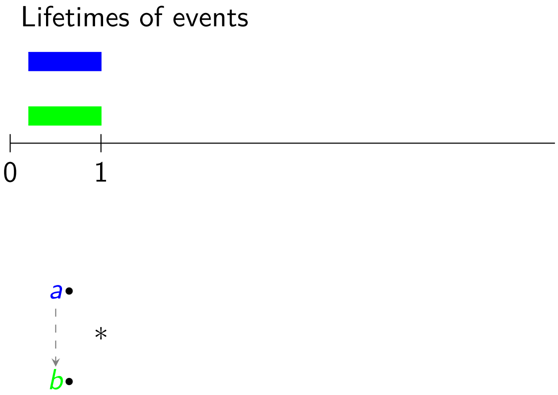


Lifetimes of events

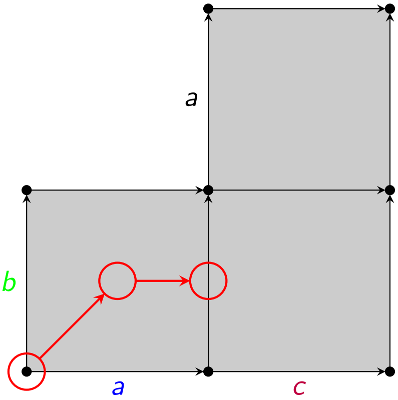


Event Ipomset of a Path

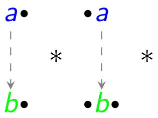




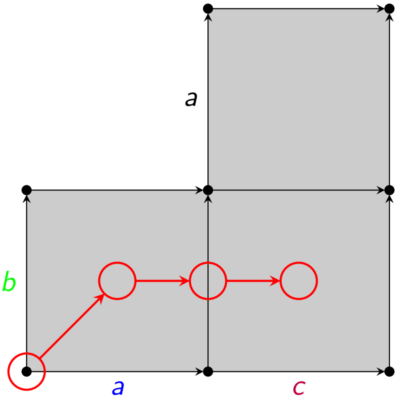
Event Ipomset of a Path

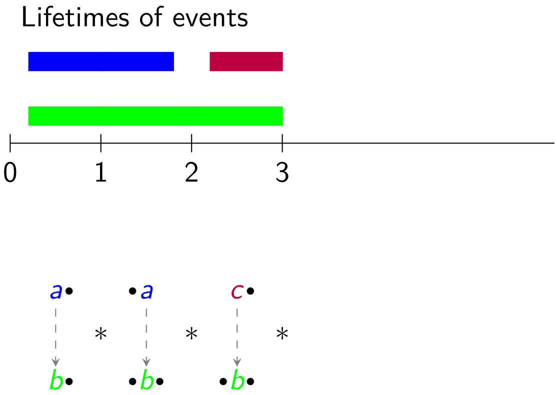


Lifetimes of events

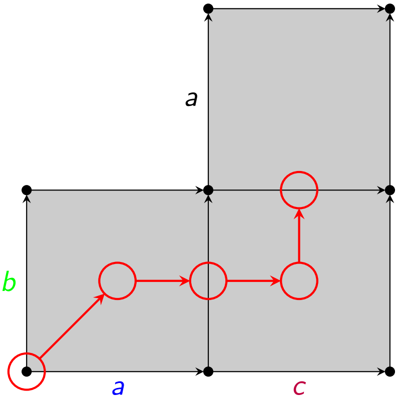


Event Ipomset of a Path

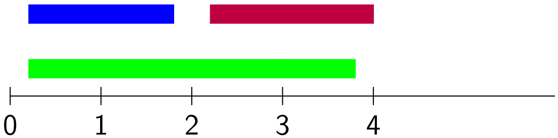


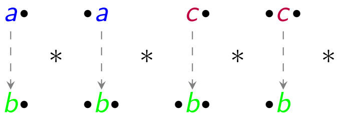


Event Ipomset of a Path

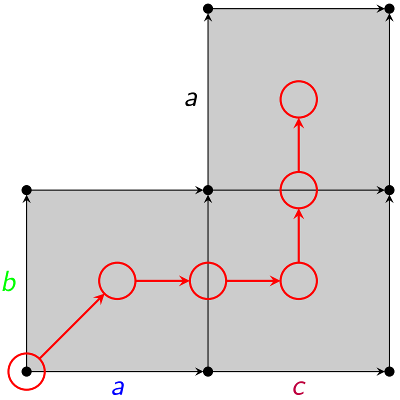


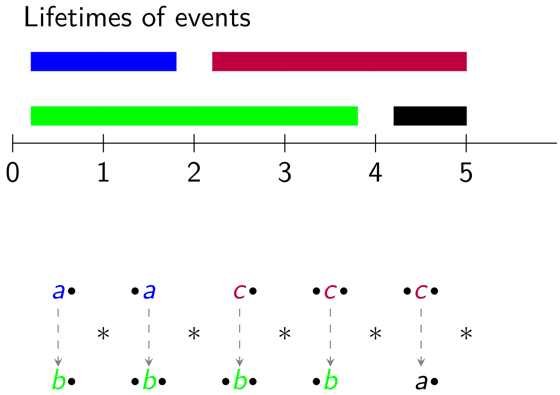
Lifetimes of events



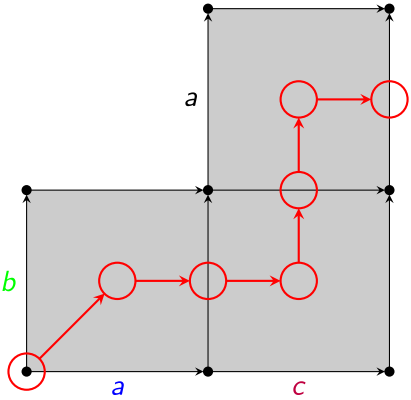


Event Ipomset of a Path

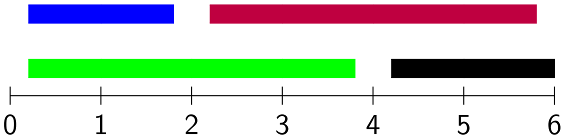


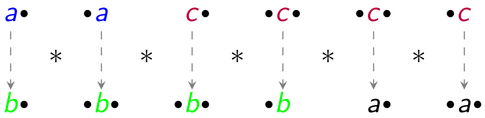


Event Ipomset of a Path

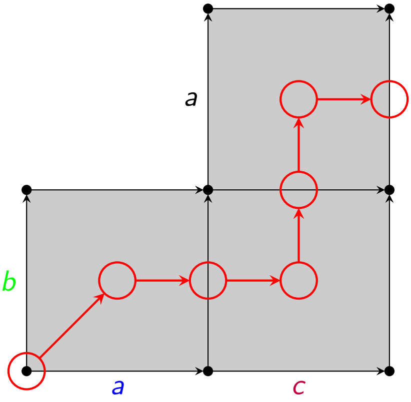


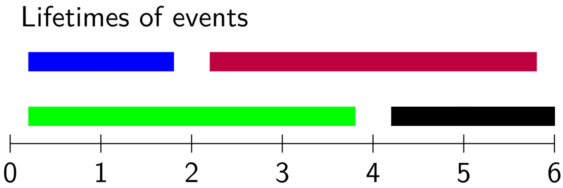
Lifetimes of events

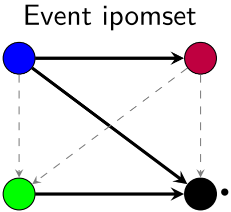




Event Ipomset of a Path







(not series-parallel!)

Selected Bibliography

- J.Desel, W.Reisig. *Place/transition Petri nets*. Lectures on Petri nets, Springer 1998
- J.Esparza. *Lecture notes on Petri nets*. 2019

- R.J.van Glabbeek. *On the expressiveness of higher dimensional automata*. Theor.Comput.Sci. 2006
- A.Amrane, H.Bazille, U.F., L.Hélouët, P.Schlehuber-Caissier. *Petri nets and higher-dimensional automata*. arxiv 2025