Discrete and Continuous Models for Concurrent Systems

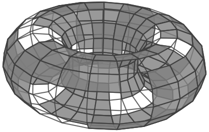
2. Concurrent Semantics of Petri Nets

Uli Fahrenberg

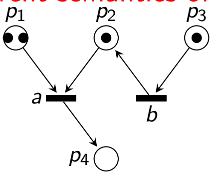
LMF, Université Paris-Saclay, France

POPL 2026 Tutorial, Rennes, France

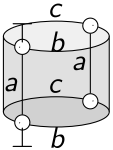
1. The geometry of concurrency



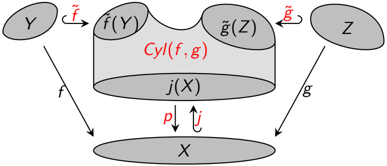
2. Concurrent semantics of Petri nets



3. Languages of higher-dimensional aut.



4. Advanced topics



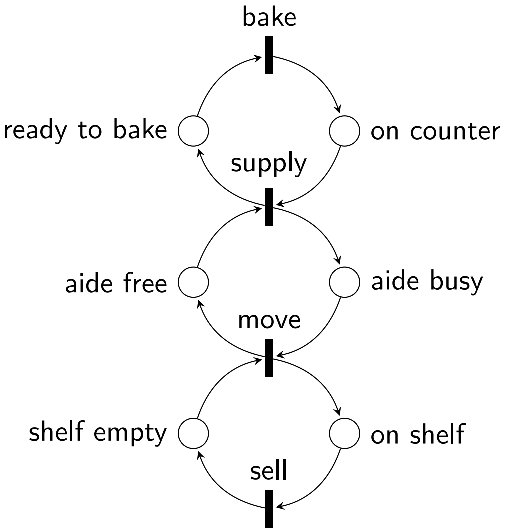
1. Introduction
2. Petri Nets
3. Higher-Dimensional Automata
4. Concurrent Semantics of Petri Nets

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

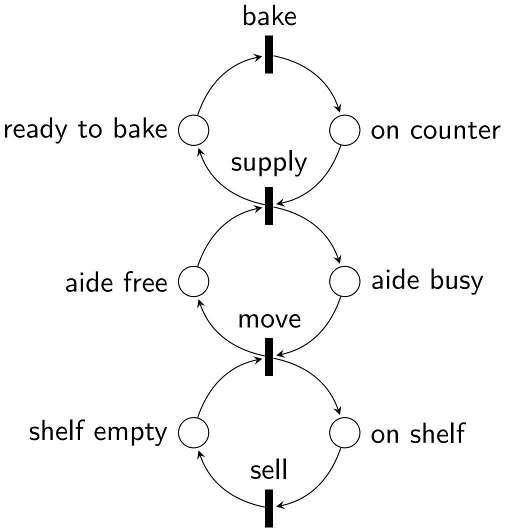
- very useful for modeling distributed or concurrent systems
- invented in 1962; ubiquitous in modeling



Petri Nets

A Petri net (S, T, F):

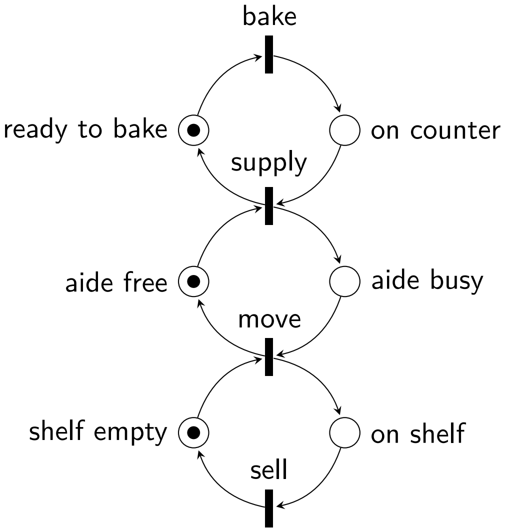
- S set of places
- T set of transitions, $S \cap T = \emptyset$
- $F : S \times T \cup T \times S \to \mathbb{N}$
  weighted flow relation



Petri Nets

A Petri net (S, T, F):

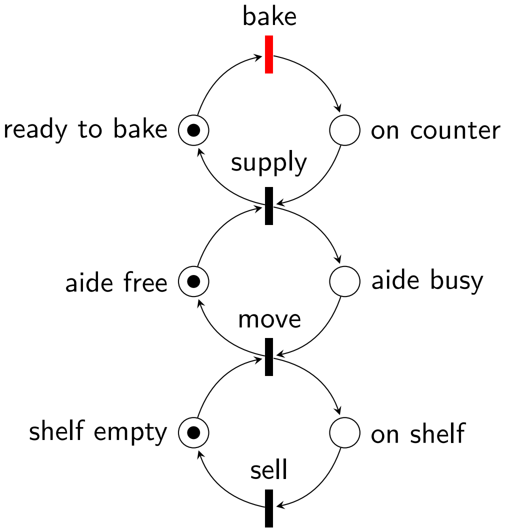
- S set of places
- T set of transitions, $S \cap T = \emptyset$
- $F : S \times T \cup T \times S \to \mathbb{N}$
  weighted flow relation
- marking: $S \to \mathbb{N}$:
  number of tokens per place
- preset of t: ${}^\bullet t = F(s, t)$
- postset of t: $t^\bullet(s) = F(t, s)$



Petri Nets

A Petri net (S, T, F):

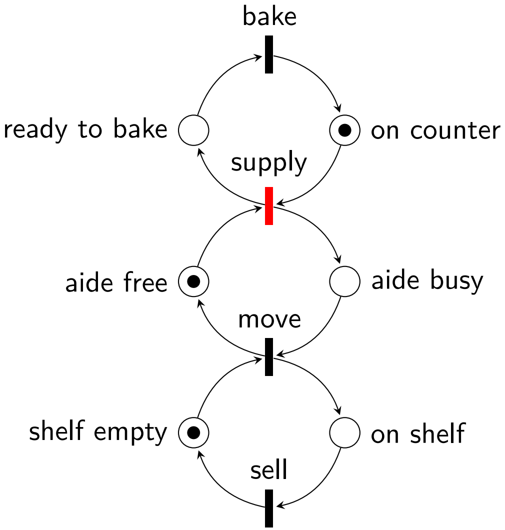
- S set of places
- T set of transitions, $S \cap T = \emptyset$
- $F : S \times T \cup T \times S \to \mathbb{N}$
  weighted flow relation
- marking: $S \to \mathbb{N}$:
  number of tokens per place
- preset of t: ${}^\bullet t = F(s, t)$
- postset of t: $t^\bullet(s) = F(t, s)$
- compute by transforming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

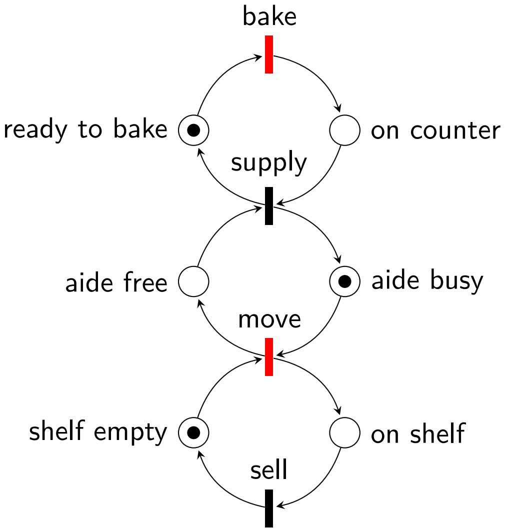
A Petri net (S, T, F):

- 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:
  $$m' = m - {}^\bullet t + t^\bullet$$

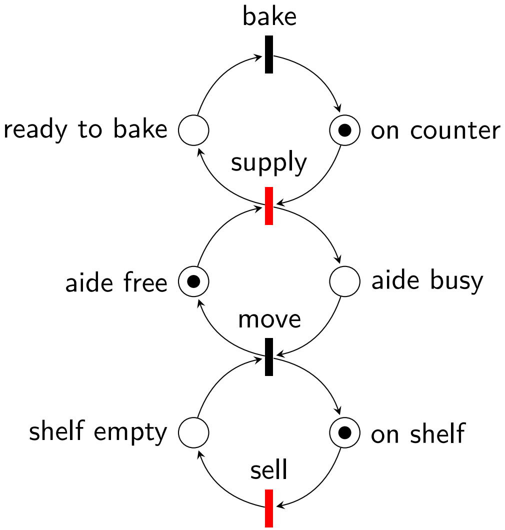
- only if ${}^\bullet t \leq m$



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

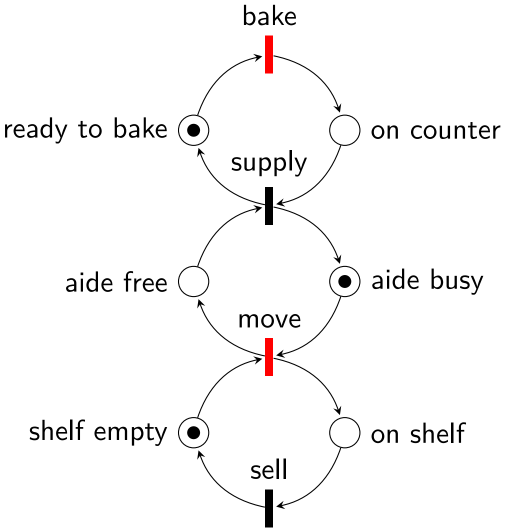
A Petri net (S, T, F):

- 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: $t^\bullet(s) = F(t, s)$

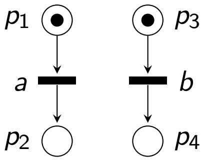
- compute by transforming markings:
  $$m' = m - {}^\bullet t + t^\bullet$$

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

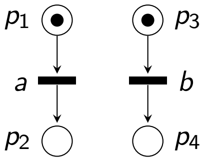


Semantics of Petri Nets

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



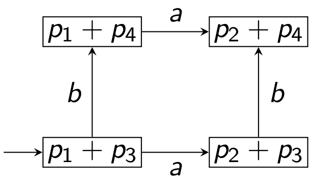
Semantics of Petri Nets



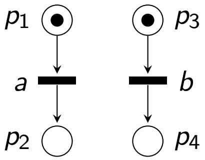
Petri net (S, T, F): places S; transitions T;
weighted flows $F : S \times T \cup T \times S \to \mathbb{N}$

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 $\implies$ initial state; take reachable part



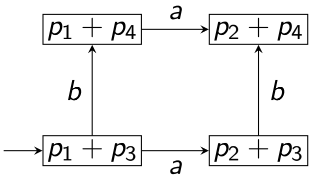
Semantics of Petri Nets



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

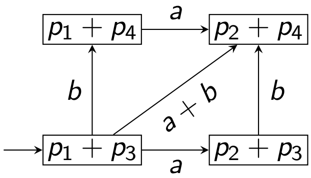
Interleaved semantics (reachability graph) (V, E):

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

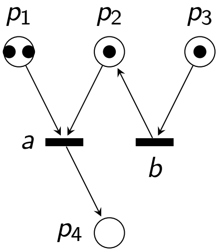


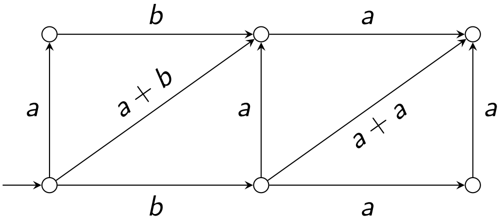
Concurrent step reachability graph (V, E'):

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



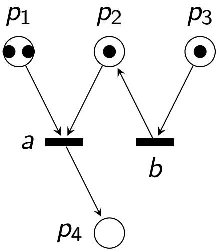
Another Example

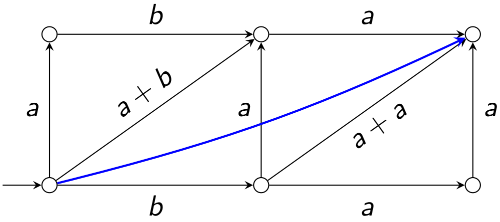




- after firing b, a is auto-concurrent

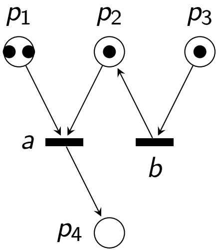
Another Example

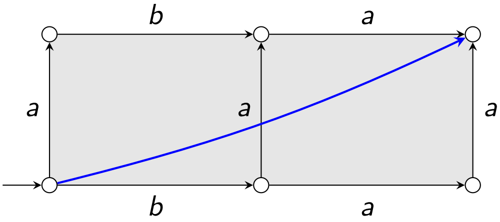




- after firing b, a is auto-concurrent
- semantics misses some behavoir?
  - start a – start b – finish b – start another a – etc.

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

1. Introduction
2. Petri Nets
3. Higher-Dimensional Automata
4. Concurrent Semantics of Petri Nets

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 $\delta^1_A : X[U] \to X[U \setminus A]$ (terminating events A)
  lower face map $\delta^0_A : X[U] \to X[U \setminus A]$ ("unstarting" events A)
- Precube identities: $\delta^\mu_A \delta^\nu_B = \delta^\nu_B \delta^\mu_A$ 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

$$\begin{tikzcd}
x & f \arrow[l, "\delta_a^0"'] \arrow[r, "\delta_a^1"] & y \; \top \\
g \arrow[u, "\delta_b^1"] \arrow[d, "\delta_b^0"'] & q \arrow[l, "\delta_a^0"'] \arrow[r, "\delta_a^1"] \arrow[u, "\delta_b^1"] \arrow[d, "\delta_b^0"] \arrow[ur, "\delta_{ab}^1"] \arrow[dl, "\delta_{ab}^0"'] & h \; \top \arrow[u, "\delta_b^1"] \arrow[d, "\delta_b^0"] \\
\perp \; v & e \arrow[l, "\delta_a^0"'] \arrow[r, "\delta_a^1"] & w
\end{tikzcd}$$

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

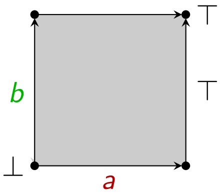
$$X[a] = \{e, f\}$$

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

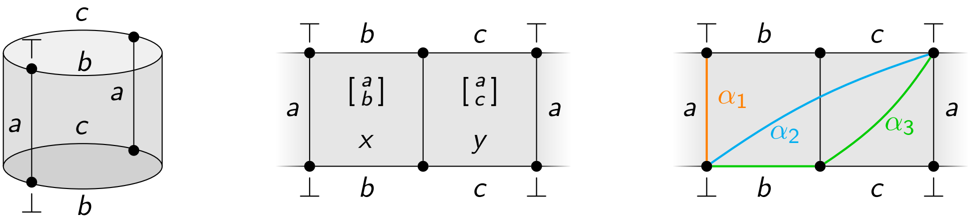
$$X[[\,{}^{a}_{b}\,]] = \{q\}$$

$$\perp_X = \{v\}$$

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

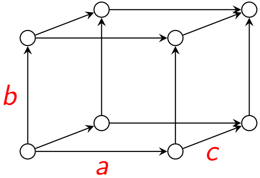


Another One

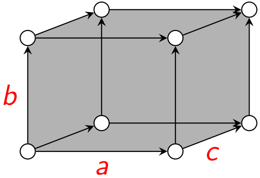


$$a \parallel (bc)^*$$

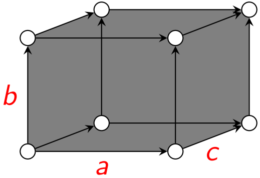
More Examples



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.)

1 Introduction

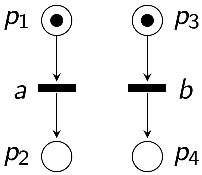
2 Petri Nets

3 Higher-Dimensional Automata

4 Concurrent Semantics of Petri Nets

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

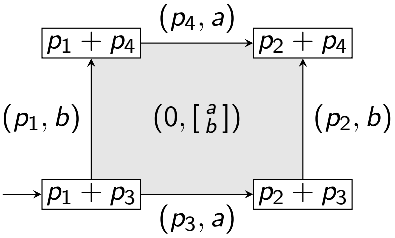
- $E = \{(m, t, m') \mid {}^\bullet t \le m, m' = m - {}^\bullet t + t^\bullet\}$

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 $\tau = (t_1, \dots, t_n)$:

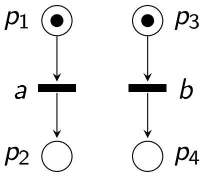
$$\delta^0_{t_i}(x) = (m + {}^\bullet t_i, (t_1, \dots, t_{i-1}, t_{i+1}, \dots, t_n))$$
$$\delta^1_{t_i}(x) = (m + t_i^\bullet, (t_1, \dots, t_{i-1}, t_{i+1}, \dots, t_n))$$

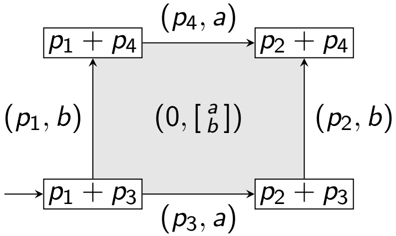
- initial marking $\implies$ initial cell; take reachable part
- (no accepting cells)



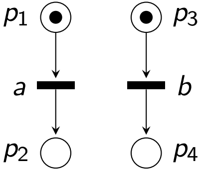
Event Order

- 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 $\square = \left\{ \begin{bmatrix} t_1 \\ \vdots \\ t_n \end{bmatrix} \;\middle|\; \forall i = 1, \dots, n-1 : t_i \preccurlyeq t_{i+1} \right\}$
  instead of $\square(T)$
- order $\preccurlyeq$ may be chosen (and re-chosen) at will
- here: lexicographic $a \prec b \prec \dots$

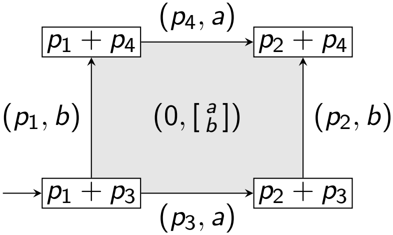


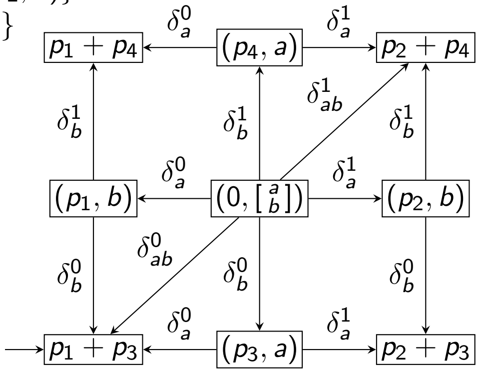


Example, Complete



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

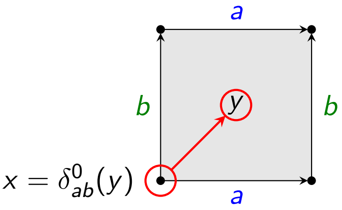




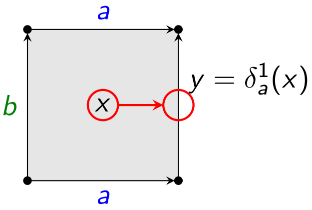
Computations of HDAs

An HDA computes by starting and terminating events in sequence:

upstep $x \nearrow y$, starting $[\begin{smallmatrix} a \\ b \end{smallmatrix}]$:



downstep $x \searrow y$, terminating a:



Idea: Use this to define an automata-like operational semantics for HDAs

- an ST-automaton (def. next slide) has
  - transitions which start and terminate events
  - states which remember which events are currently running

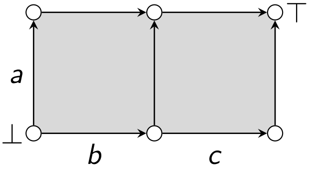
ST-Automata

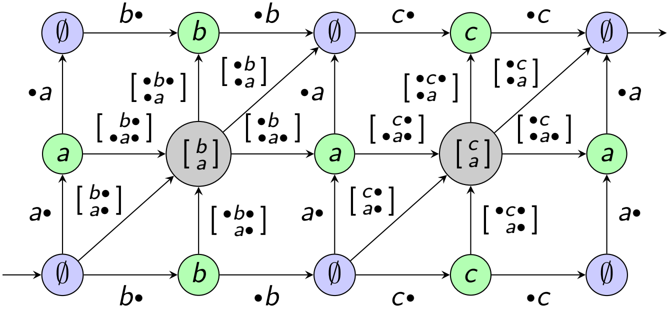
- a starter (A, U): conclist U, subset $A \subseteq U$
- a terminator (U, B): conclist U, subset $B \subseteq U$
- starting A; terminating B: written ${}_A{\uparrow}U$ resp. $U{\downarrow}_B$
- Let ST denote the (infinite) set of starters and terminators

An ST-automaton $(Q, \bot, \top, E, \lambda)$:

- Q set of states; $\bot, \top \subseteq Q$ initial resp. accepting states
- $E \subseteq Q \times \mathsf{ST} \times Q$ transitions
- $\lambda : Q \to \square$ state labeling, such that for all $(p, x, q) \in E$:
  - if $x = {}_A{\uparrow}U$, then $\lambda(p) = U \setminus A$ and $\lambda(q) = U$;
  - if $x = U{\downarrow}_B$, then $\lambda(p) = U$ and $\lambda(q) = U \setminus B$.

Translation



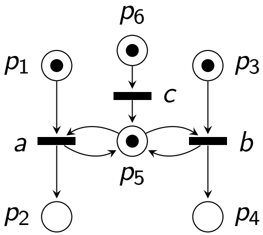


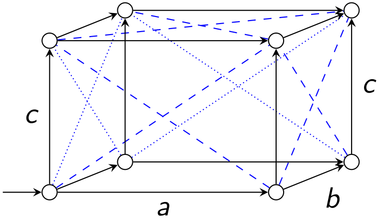
from HDA $(X, \perp, \top)$ to ST-automaton $(Q, \perp, \top, E, \lambda)$:

- $Q = X$, $\lambda = \text{ev}$, $E = \{\delta_A^0(x) \xrightarrow{A\uparrow\text{ev}(x)} x \mid A \subseteq \text{ev}(x)\} \cup \{x \xrightarrow{\text{ev}(x)\downarrow_A} \delta_A^1(x) \mid A \subseteq \text{ev}(x)\}$

from ST-automata to HDAs: complicated; we've lost geometric information

One Last Example





- initially, p_5 is a mutex place: it disables concurrency of a and b
- after c fires, p_5 holds two tokens, so a and b become concurrent

- semantically, a hollow cube without bottom face
- the five faces:

front:	$(p_3, [\,{}^{a}_{c}\,])$,	back:	$(p_4, [\,{}^{a}_{c}\,])$
left:	$(p_1, [\,{}^{b}_{c}\,])$,	right:	$(p_2, [\,{}^{b}_{c}\,])$
top:	$(0, [\,{}^{a}_{b}\,])$		

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*. PETRI NETS 2025
- A.Amrane, H.Bazille, E.Clement, U.F., K.Ziemiański. *Presenting interval pomsets with interfaces*. RAMiCS 2024