Petri Nets, Concurrency, and Real Time

Uli Fahrenberg

EPITA Research Laboratory (LRE), Rennes/Paris, France

LMF Informel, March 2025

Petri Nets, Concurrency, and Real Time

. . . and Higher-Dimensional Automata

Uli Fahrenberg

EPITA Research Laboratory (LRE), Rennes/Paris, France

LMF Informel, March 2025

Higher-dimensional automata:

- Languages of Higher-Dimensional Automata [MSCS 2021]
- Posets with Interfaces as a Model for Concurrency [I&C 2022]
- Kleene Theorem for Higher-Dimensional Automata [LMCS 2024]
- Myhill-Nerode Theorem for Higher-Dimensional Automata [FI 2024]
- Decision and Closure Properties for Higher-Dimensional Automata [TCS 2025]

. . . and Petri nets:

- Petri Nets and Higher-Dimensional Automata [Petri Nets 2025]

. . . and real time?

- Higher-Dimensional Timed and Hybrid Automata [LITES 2022]
- Languages of Higher-Dimensional Timed Automata [Petri Nets 2024]

Nice people

- Eric Goubault, Paris
- Christian Johansen, Gjøvik
- Georg Struth, Sheffield
- Krzysztof Ziemiański, Warsaw

- Amazigh Amrane (Paris), Hugo Bazille (Rennes), Emily Clement (Paris), Jérémy Dubut (Paris), Marie Fortin (Paris), Loïc Hélouët (Rennes), Jérémy Ledent (Paris), Philipp Schlehuber-Caissier (Paris), Safa Zouari (Gjøvik), . . .

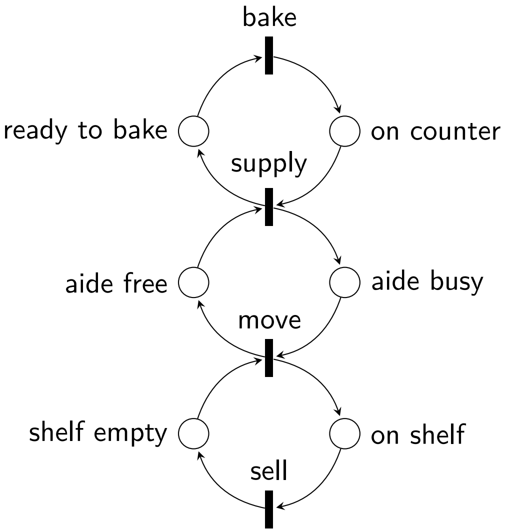
- See also `https://ulifahrenberg.github.io/pomsetproject/`

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

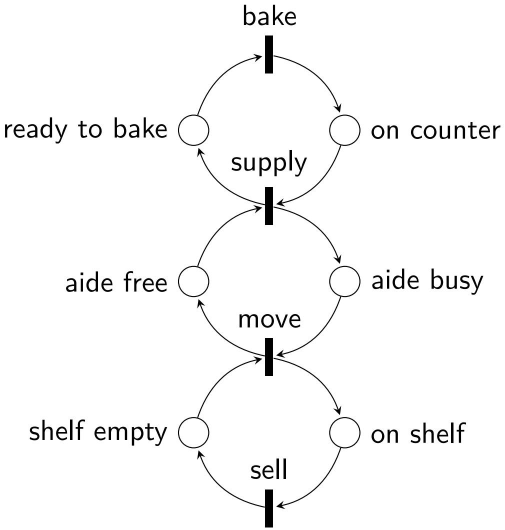
- very useful for modeling distributed or concurrent systems
- invented in 1962; ubiquitous in modeling



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

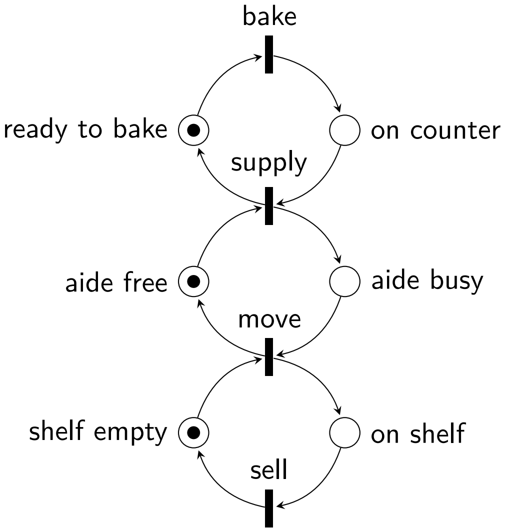


Petri Nets

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

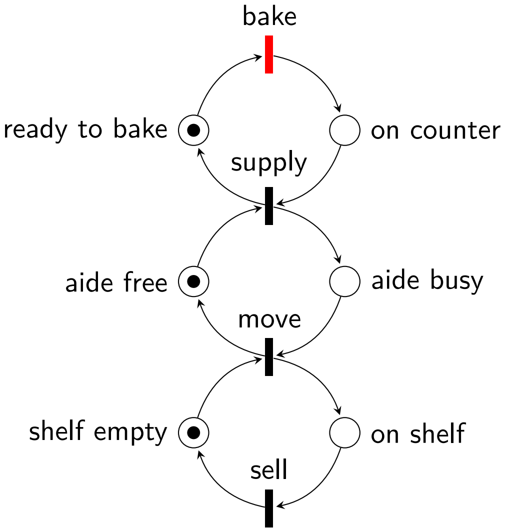
- marking: $S \to \mathbb{N}$:
  number of tokens per place
- preset of t: ${}^\bullet t = F(s, t)$
- postset of t: $t^\bullet(s) = F(t, s)$



Petri Nets

A Petri net (S, T, F):

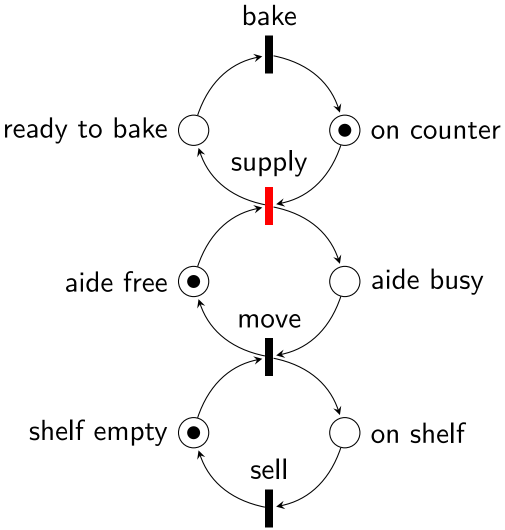
- S set of places
- T set of transitions, $S \cap T = \emptyset$
- $F : S \times T \cup T \times S \to \mathbb{N}$
  weighted flow relation
- marking: $S \to \mathbb{N}$:
  number of tokens per place
- preset of t: ${}^\bullet t = F(s, t)$
- postset of t: $t^\bullet(s) = F(t, s)$
- compute by transforming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



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 ${}^\bullet t \le m$



Petri Nets

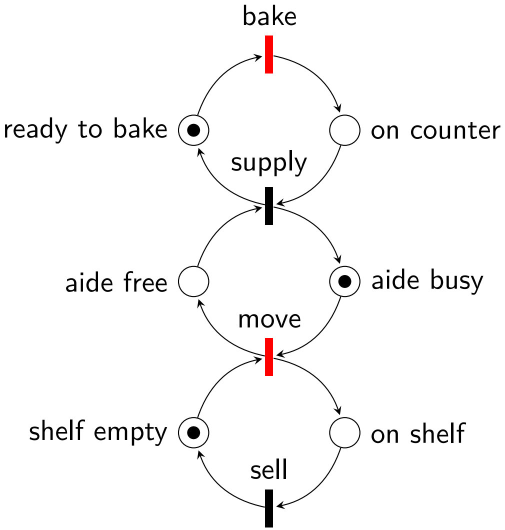
A Petri net (S, T, F):

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

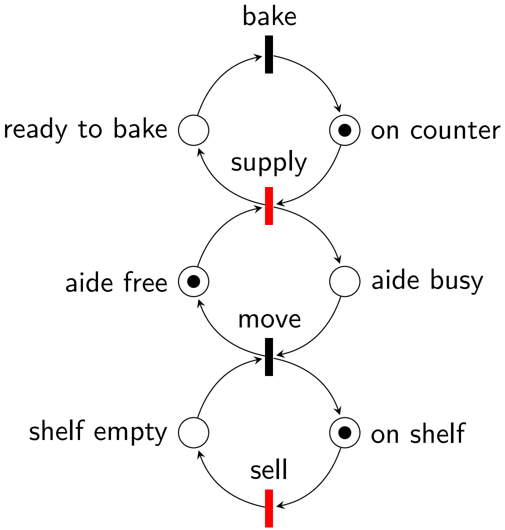
- only if ${}^\bullet t \leq m$



Petri Nets

A Petri net (S, T, F):

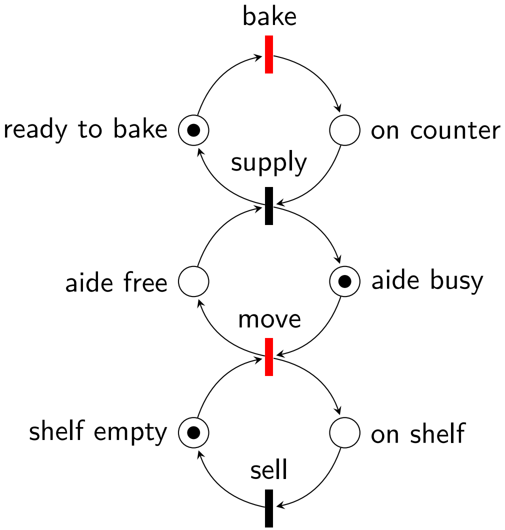
- S set of places
- T set of transitions, $S \cap T = \emptyset$
- $F : S \times T \cup T \times S \to \mathbb{N}$
  weighted flow relation
- marking: $S \to \mathbb{N}$:
  number of tokens per place
- preset of t: ${}^\bullet t = F(s, t)$
- postset of t: $t^\bullet(s) = F(t, s)$
- compute by transforming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



Petri Nets

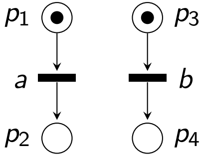
A Petri net (S, T, F):

- 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 $F : S \times T \cup T \times S \to \mathbb{N}$

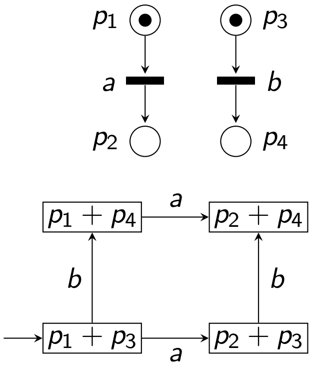


Semantics of Petri Nets

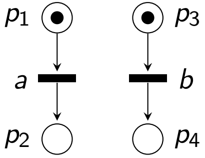
Petri net (S, T, F): places S; transitions T;
weighted flows $F : S \times T \cup T \times S \to \mathbb{N}$

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

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



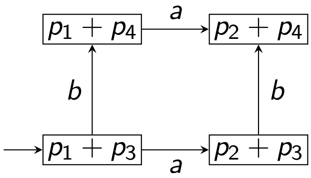
Semantics of Petri Nets



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

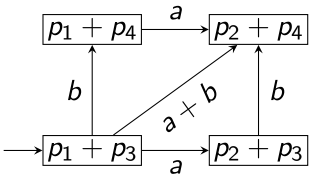
Interleaved semantics (reachability graph) (V, E):

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

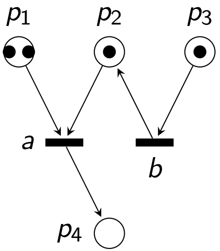


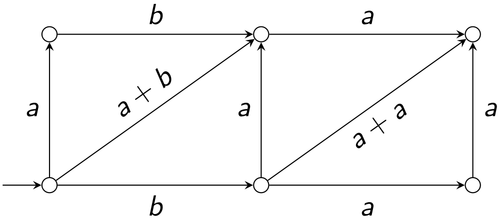
Concurrent step reachability graph (V, E'):

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



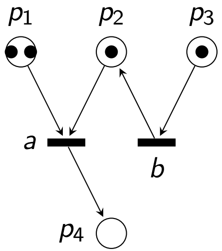
Another Example

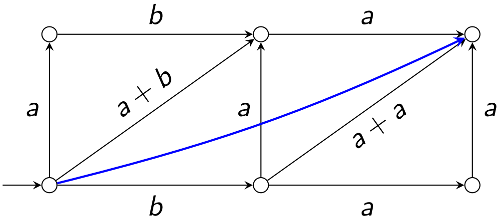




- after firing b, a is auto-concurrent

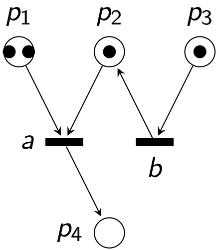
Another Example

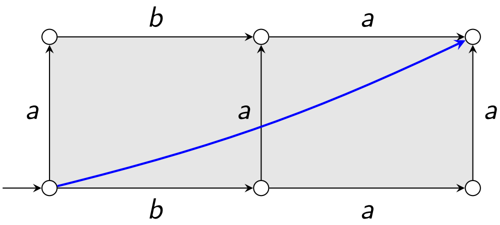




- after firing b, a is auto-concurrent
- semantics misses some behavoir?
  - start a – start b – finish b – start another a – etc.

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

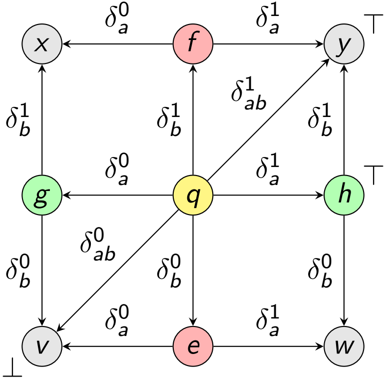
A conclist is a finite, totally ordered, Σ-labeled set. (a list of labeled events)

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 $\delta^1_A : X[U] \to X[U \setminus A]$ (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 $\perp \subseteq X$ and accepting cells $\top \subseteq X$ (not necessarily vertices)

Example



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

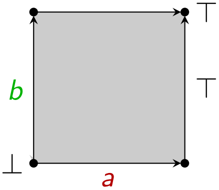
$$X[a] = \{e, f\}$$

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

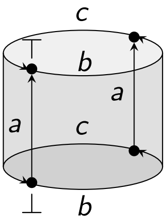
$$X[[\begin{smallmatrix} a \\ b \end{smallmatrix}]] = \{q\}$$

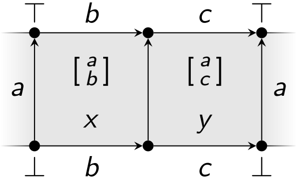
$$\perp_X = \{v\}$$

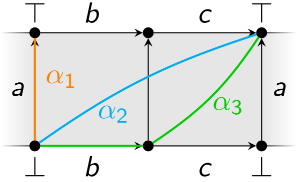
$$\top_X = \{h, y\}$$



Another One

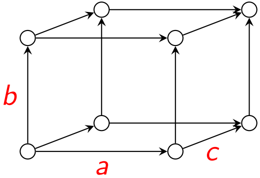




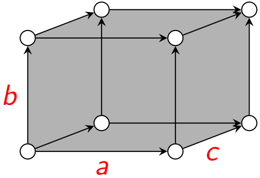


$$a \parallel (bc)^*$$

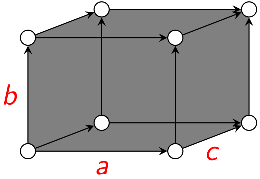
More Examples



no concurrency



two out of three



full concurrency

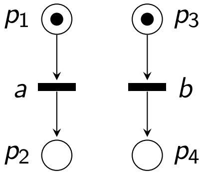
Higher-Dimensional Automata & Concurrency Theory

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.)

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

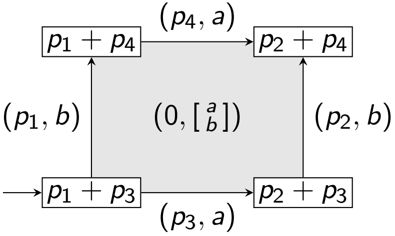
- $E = \{(m, t, m') \mid {}^\bullet t \le m, m' = m - {}^\bullet t + t^\bullet\}$

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

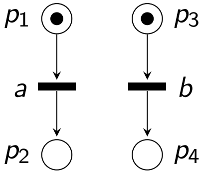
$$\delta^0_{t_i}(x) = (m + {}^\bullet t_i, (t_1, \dots, t_{i-1}, t_{i+1}, \dots, t_n))$$
$$\delta^1_{t_i}(x) = (m + t_i^\bullet, (t_1, \dots, t_{i-1}, t_{i+1}, \dots, t_n))$$

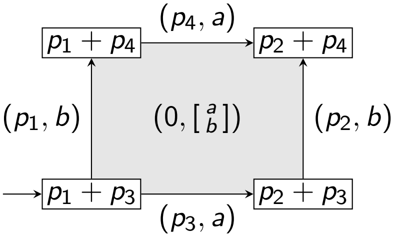
- initial marking $\implies$ initial cell; take reachable part
- (no accepting cells)



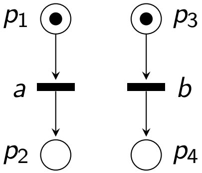
Event Order

- 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 $a \prec b \prec \dots$

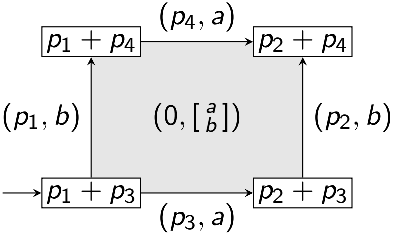


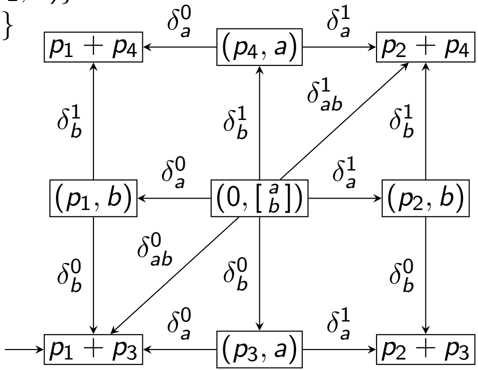


Example, Complete

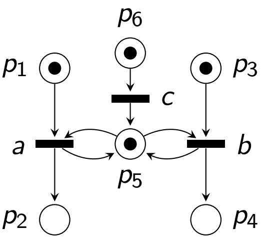


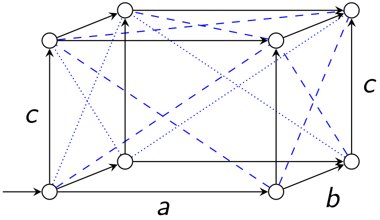
$$X[\emptyset] = \{p_1 + p_3, p_2 + p_3, p_1 + p_4, p_2 + p_4\}$$
$$X[a] = \{(p_3, a), (p_4, a)\}$$
$$X[b] = \{(p_1, b), (p_2, b)\}$$
$$X[[\begin{smallmatrix} a \\ b \end{smallmatrix}]] = \{(0, [\begin{smallmatrix} a \\ b \end{smallmatrix}])\}$$





One Last Example

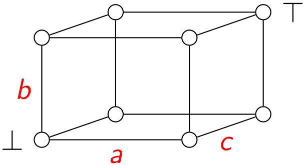




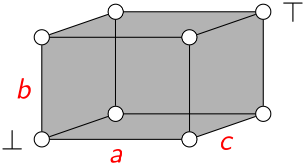
- initially, p_5 is a mutex place: it disables concurrency of a and b
- after c fires, p_5 holds two tokens, so a and b become concurrent

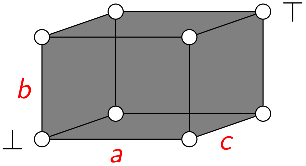
- semantically, a hollow cube without bottom face ("Fahrenberg's matchbox")
- the five faces: front: $(p_3, [\,^a_c\,])$, back: $(p_4, [\,^a_c\,])$
  left: $(p_1, [\,^b_c\,])$, right: $(p_2, [\,^b_c\,])$
  top: $(0, [\,^a_b\,])$

Languages of HDAs: Examples



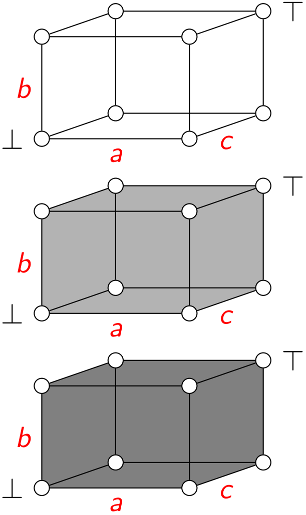
$$L_1 = \{abc, acb, bac, bca, cab, cba\}$$





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

Languages of HDAs: Examples



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

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

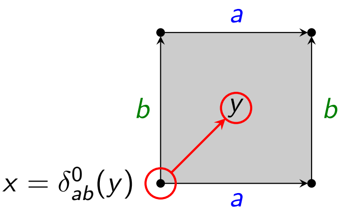
sets of pomsets

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

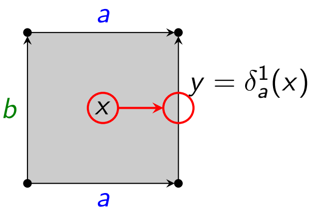
Computations of HDAs

An HDA computes by starting and terminating events in sequence:

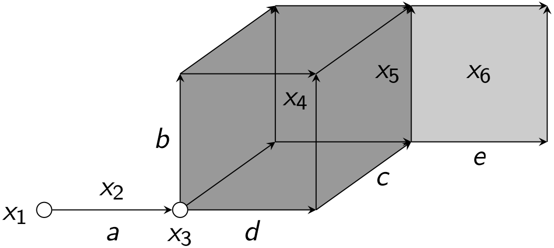
upstep $x \nearrow y$, starting $[\begin{smallmatrix} a \\ b \end{smallmatrix}]$:



downstep $x \searrow y$, terminating a:

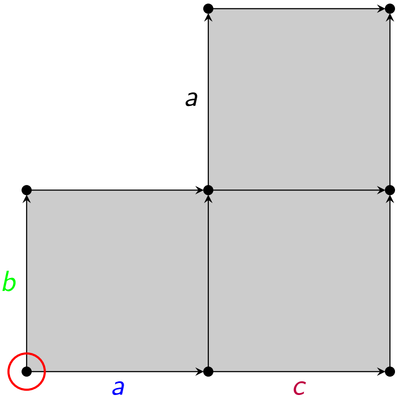


Example

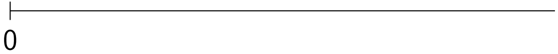


$$\left(x_1 \nearrow^{a} x_2 \searrow_{a} x_3 \nearrow^{\{b,c,d\}} x_4 \searrow_{\{c,d\}} x_5 \nearrow^{e} x_6\right)$$

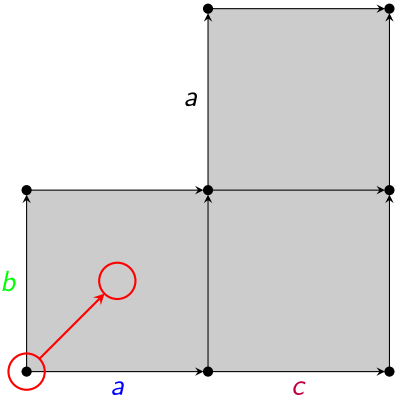
Event Ipomset of a Path



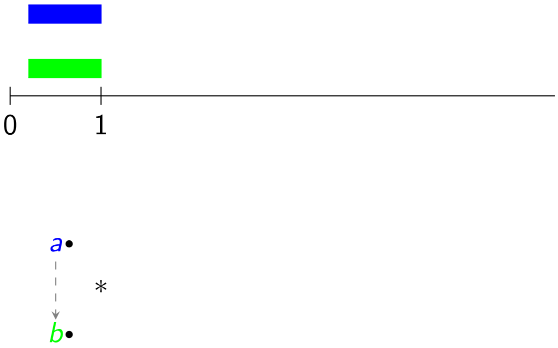
Lifetimes of events



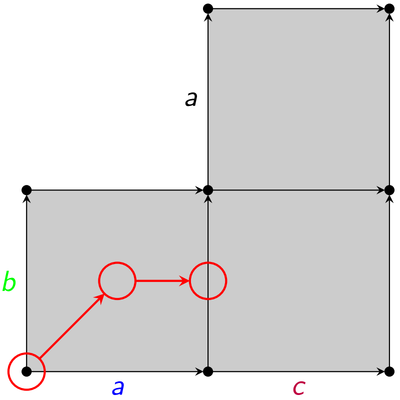
Event Ipomset of a Path



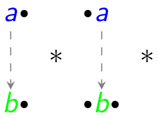
Lifetimes of events



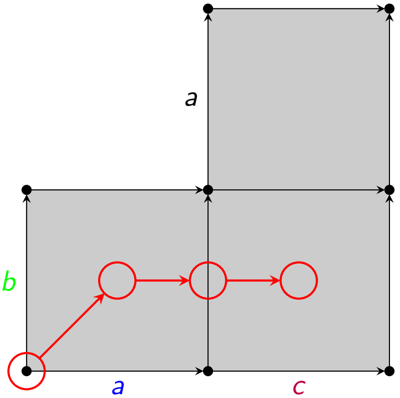
Event Ipomset of a Path



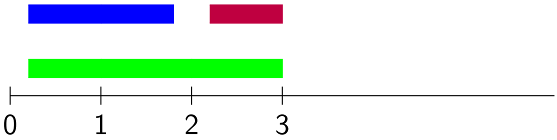
Lifetimes of events

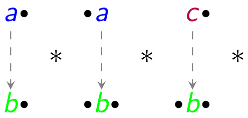


Event Ipomset of a Path

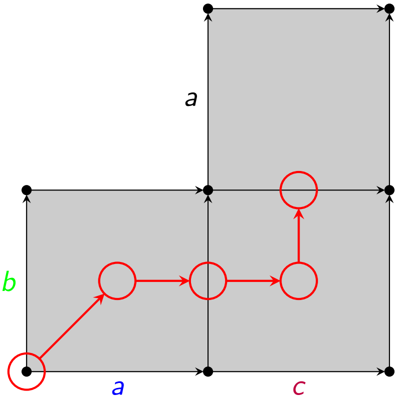


Lifetimes of events

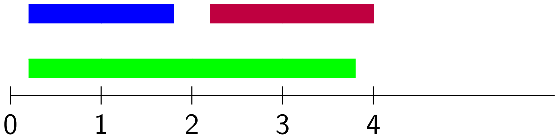


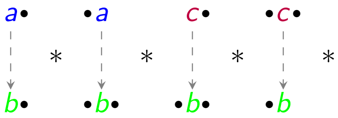


Event Ipomset of a Path

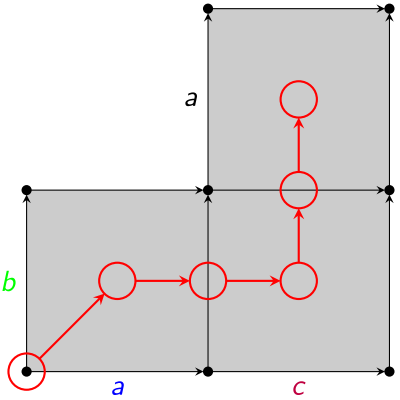


Lifetimes of events

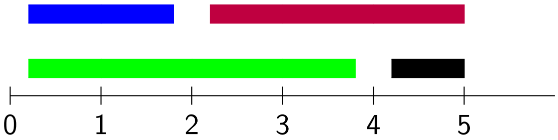


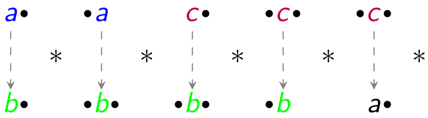


Event Ipomset of a Path

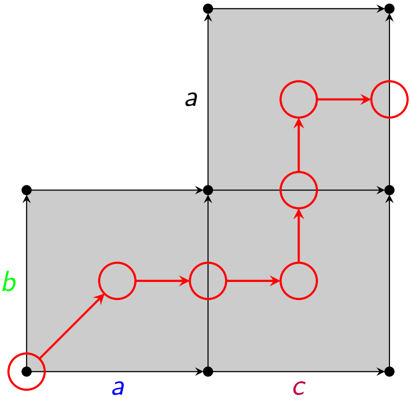


Lifetimes of events

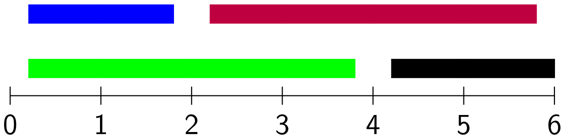


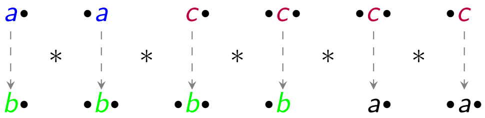


Event Ipomset of a Path

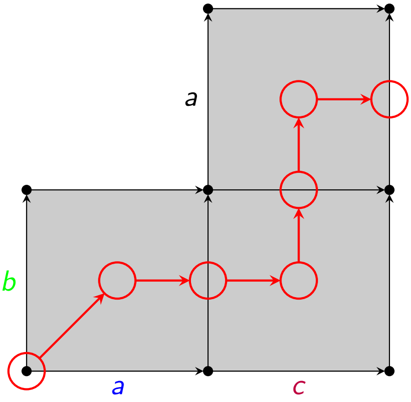


Lifetimes of events

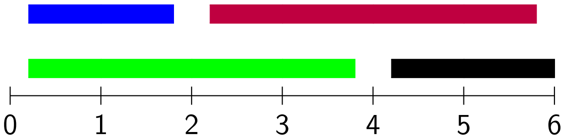




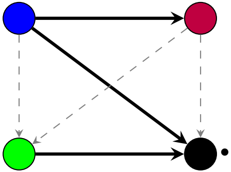
Event Ipomset of a Path



Lifetimes of events



Event ipomset



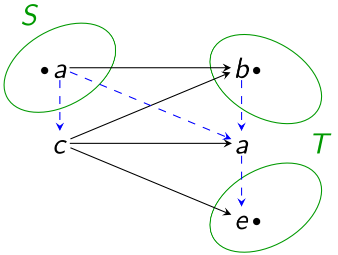
(not series-parallel!)

Pomsets with Interfaces

Definition

A pomset with interfaces (ipomset): $(P, <, \dashrightarrow, S, T, \lambda)$:

- P finite set of events, $\lambda : P \to \Sigma$
- two partial orders $<$ (precedence order), $\dashrightarrow$ (event order)
  - s.t. $< \cup \dashrightarrow$ is a total relation;
- $S, T \subseteq P$ source and target interfaces
  - s.t. S is $<$-minimal and T is $<$-maximal.

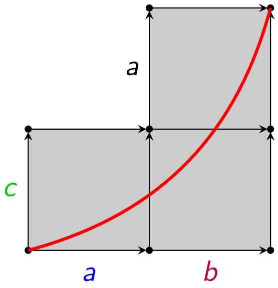


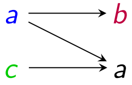
Interval Orders

Definition

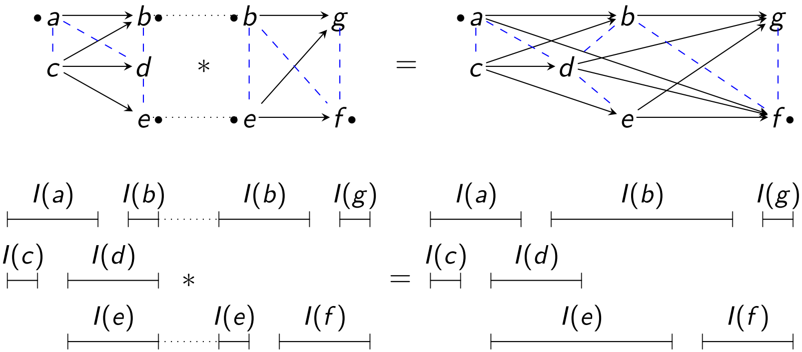
An ipomset $(P, <_P, \dashrightarrow, S, T, \lambda)$ is interval if $(P, <_P)$ has an interval representation: functions $b, e : P \to \mathbb{R}$ s.t.

- $\forall x \in P : b(x) \le_\mathbb{R} e(x)$;
- $\forall x, y \in P : e(x) <_\mathbb{R} b(y) \iff x <_P y$



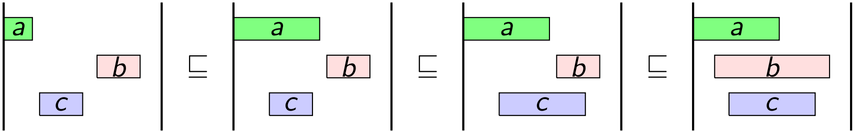


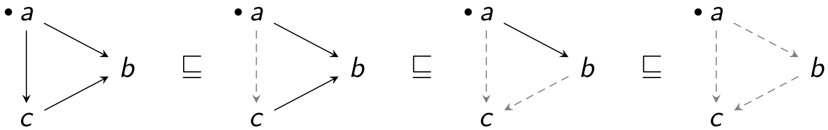
Gluing Composition



- Gluing $P * Q$: P before Q, except for interfaces (which are identified)
- (also have parallel composition $P \parallel Q$: disjoint union)

Subsumption





P refines Q / Q subsumes P / $P \sqsubseteq Q$ iff

- P and Q have same interfaces
- P has more $<$ than Q
- Q has more $\dashrightarrow$ than P

Languages of HDAs

Definition

The language of an HDA X is the set of event ipomsets of all accepting paths:

$$L(X) = \{\mathrm{ev}(\pi) \mid \pi \in \mathrm{Paths}(X), \mathrm{src}(\pi) \in \bot_X, \mathrm{tgt}(\pi) \in \top_X\}$$

- $L(X)$ contains only interval ipomsets,
- is closed under subsumption,
- and has finite width

Definition

A language $L \subseteq \mathrm{iiPoms}$ is regular if there is an HDA X with $L = L(X)$.

Theorems

Definition (Rational Languages over Σ)

- Generated by $\emptyset$, $\{\epsilon\}$, and all $\{[a]\}$, $\{[\bullet\, a]\}$, $\{[a\,\bullet]\}$, $\{[\bullet\, a\,\bullet]\}$ for $a \in \Sigma$
- under operations $\cup$, $*$, $\|$ and (Kleene plus) $^+$
- (these need to take subsumption closure into account)

Definition (Monadic Second-Order Logics over Ipomsets)

$$\psi ::= a(x) \mid \mathrm{s}(x) \mid \mathrm{t}(x) \mid x < y \mid x \dashrightarrow y \mid x \in X \mid$$
$$\exists x.\, \psi \mid \forall x.\, \psi \mid \exists X.\, \psi \mid \forall X.\, \psi \mid \psi_1 \wedge \psi_2 \mid \psi_1 \vee \psi_2 \mid \neg\psi$$

Theorem (à la Kleene): regular $\iff$ rational

Theorem (à la Myhill-Nerode): regular $\iff$ finite prefix quotient

Theorem (à la Büchi-Elgot-Trakhtenbrot): [DLT 2024]

regular $\iff$ MSO-definable, of finite width, and subsumption-closed

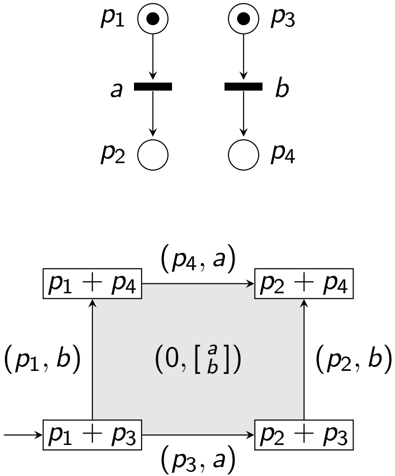
Languages of Petri Nets

Petri net $\rightsquigarrow$ HDA semantics $\rightsquigarrow$ language semantics

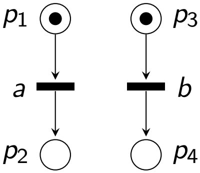
$\Longrightarrow$ languages of Petri nets are sets of interval orders

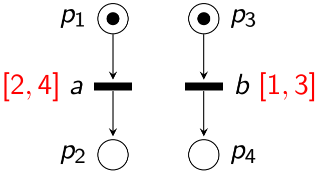
- related to work by Janicki-Kleijn-Koutny-Mikulski
- relation to trace theory?

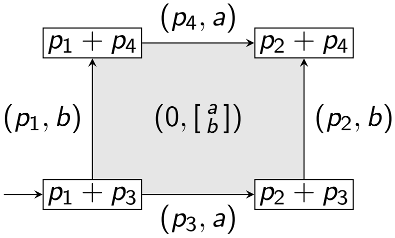
It Is Time!


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

It Is Time!

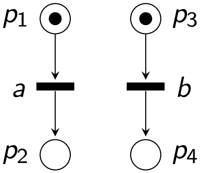

p_1
p_3
a
b
p_2
p_4

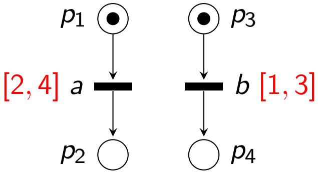

p_1
p_3
$[2,4]$ a
b $[1,3]$
p_2
p_4

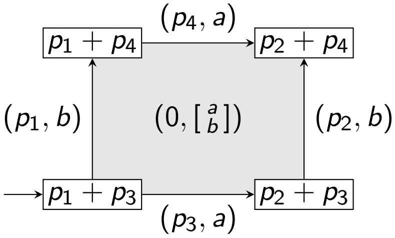

(p_4, a)
$p_1 + p_4$
$p_2 + p_4$
(p_1, b)
$(0, [\begin{smallmatrix} a \\ b \end{smallmatrix}])$
(p_2, b)
$p_1 + p_3$
$p_2 + p_3$
(p_3, a)

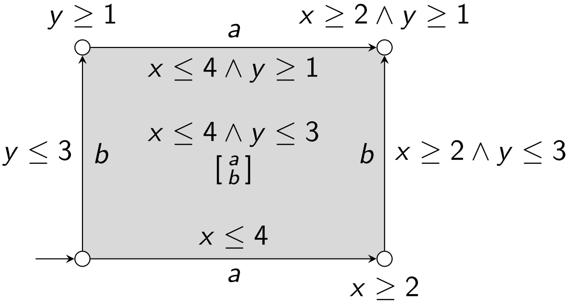
???

It Is Time!


p_1
p_3
a
b
p_2
p_4


p_1
p_3
$[2,4]$ a
b $[1,3]$
p_2
p_4


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


$y \geq 1$
a
$x \geq 2 \wedge y \geq 1$
$x \leq 4 \wedge y \geq 1$
$x \leq 4 \wedge y \leq 3$
$y \leq 3$
b
$[\begin{smallmatrix} a \\ b \end{smallmatrix}]$
b
$x \geq 2 \wedge y \leq 3$
$x \leq 4$
a
$x \geq 2$

Higher-dimensional timed automata

Definition (higher-dimensional timed automaton)

An HDTA is a structure $(\Sigma, C, Q, \bot, \top, \mathsf{inv}, \mathsf{exit})$, where $(\Sigma, Q, \bot, \top)$ is a finite HDA and $\mathsf{inv} : Q \to \Phi(C)$, $\mathsf{exit} : Q \to 2^C$ assign invariant and exit conditions to each cell.

Definition (operational semantics)

The op.sem. of an HDTA $A = (Q, \bot, \top, \mathsf{inv}, \mathsf{exit})$ is the state-labeled automaton $[\![A]\!] = (S, \rightsquigarrow, S^\bot, S^\top, \rho)$, with $\rightsquigarrow \subseteq S \times (\mathsf{St} \cup \mathsf{Te} \cup \mathbb{R}_{\geq 0}) \times S$, given as follows:

$$S = \{(q, v) \in Q \times \mathbb{R}_{\geq 0}^C \mid v \models \mathsf{inv}(q)\} \qquad \rho((q, v)) = \mathsf{ev}(q)$$

$$S^\bot = \{(q, v^0) \mid q \in \bot\} \qquad S^\top = S \cap \top \times \mathbb{R}_{\geq 0}^C$$

$$\begin{aligned}\rightsquigarrow = {} & \{((q, v), d, (q, v + d)) \mid \forall 0 \leq d' \leq d : v + d' \models \mathsf{inv}(q)\} \\ & \cup \{((\delta_A^0(q), v), {}_A{\uparrow}\mathsf{ev}(q), (q, v')) \mid A \subseteq \mathsf{ev}(q), v' = v[\mathsf{exit}(\delta_A^0(q)) \leftarrow 0]\} \\ & \cup \{((q, v), \mathsf{ev}(q){\downarrow}_A, (\delta_A^1(q), v')) \mid A \subseteq \mathsf{ev}(q), v' = v[\mathsf{exit}(q) \leftarrow 0]\}\end{aligned}$$

Higher-dimensional timed automata

Definition (higher-dimensional timed automaton)

An HDTA is a structure $(\Sigma, C, Q, \bot, \top, \mathrm{inv}, \mathrm{exit})$, where $(\Sigma, Q, \bot, \top)$ is a finite HDA and $\mathrm{inv} : Q \to \Phi(C)$, $\mathrm{exit} : Q \to 2^C$ assign …ons to each cell.

starters & terminators

may delay anywhere

Definition (operational semantics)

The op.sem. of an HDTA $A = (Q, \bot, \top, \mathrm{inv}, \mathrm{exit})$ is the state-labeled automaton $[\![A]\!] = (S, \rightsquigarrow, S^\bot, S^\top, \rho)$, with $\rightsquigarrow \subseteq S \times (\mathrm{St} \cup \mathrm{Te} \cup \mathbb{R}_{\geq 0}) \times S$, given as follows:

exit when leaving

start/terminate A

$$S = \{(q, v) \in Q \times \mathbb{R}^C_{\geq 0} \mid \ldots ((q, v)) = \mathrm{ev}(q)\}$$

$$S^\bot = \{(q, v^0) \mid q \in \bot\} \qquad S^\top = \ldots \times \mathbb{R}^C_{\geq 0}$$

$$\rightsquigarrow = \{((q, v), d, (q, v + d)) \mid \forall 0 \leq d' \leq d : v + d' \models \mathrm{inv}(q)\}$$

$$\cup \{((\delta^0_A(q), v), {}_A{\uparrow}\mathrm{ev}(q), (q, v')) \mid A \subseteq \mathrm{ev}(q), v' = v[\mathrm{exit}(\delta^0_A(q)) \leftarrow 0]\}$$

$$\cup \{((q, v), \mathrm{ev}(q){\downarrow}_A, (\delta^1_A(q), v')) \mid A \subseteq \mathrm{ev}(q), v' = v[\mathrm{exit}(q) \leftarrow 0]\}$$

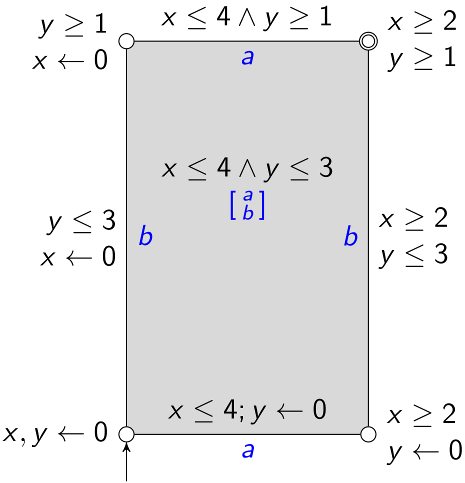
Actions take time?

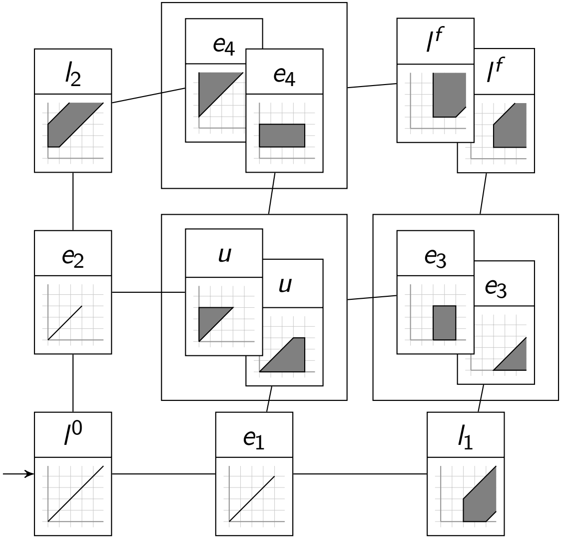
- Cardelli 1982 (ICALP): Actions take time.
  - 'We read $p \xrightarrow[t]{a} q$ as "p moves to q performing a for an interval t"'
- since Alur-Dill 1990 (even before?): Actions are immediate.
  - $(l, v) \overset{d}{\leadsto} (l, v+d) \overset{s}{\leadsto} (l', v+d)$
- Kim G. Larsen (personal discussions): Actions are immediate mostly because of technical reasons. ("We know how to do; it's nice; and it's sufficient")
- Henzinger-Manna-Pnueli 1990: same
- Chatain-Jard 2013: In the concurrent semantics for time Petri nets, time has to (locally) be allowed to run backwards??
- U.F. 2018: In real-time concurrency, actions cannot be immediate.
  - and it appears that the "technical reasons" argument is quite weak!

Good news

- regions ✓
- zones ✓
- zone-based reachability ✓
  - reachability is PSPACE-complete
- language inclusion undecidable
- untimings of languages are regular $\Longrightarrow$ untimed language inclusion decidable ✓

Zone-Based Reachability



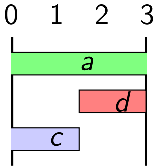


Languages of HDTAs

Definition

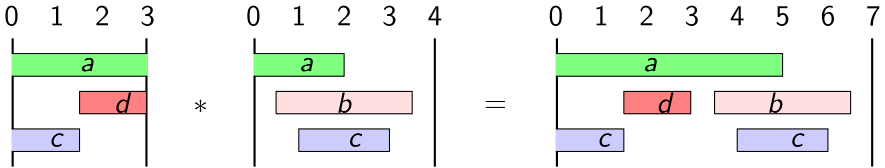
A timed ipomset is $P = (P, <_P, \dashrightarrow, S, T, \lambda, \sigma^-, \sigma^+, d)$:

- P set of events, $\lambda : P \to \Sigma$
- $<$ precedence, $\dashrightarrow$ event order s.t. $< \cup \dashrightarrow$ is a total relation
- $S, T \subseteq P$ sources & targets s.t. S is $<$-minimal, T is $<$-maximal
- $\sigma^-, \sigma^+ : P \to \mathbb{R}_{\geq 0}$ interval timestamps, $d \in \mathbb{R}_{\geq 0}$ duration
- $\forall x \in P$, $0 \leq \sigma^-(x) \leq \sigma^+(x) \leq d$
- $x \in S \implies \sigma^-(x) = 0$; $x \in T \implies \sigma^+(x) = d$
- $\sigma^+(x) < \sigma^-(y) \implies x <_P y \implies \sigma^+(x) \leq \sigma^-(y)$



Languages of HDTAs

- timed ipomsets look (almost) like signals
- with interfaces and a gluing operation



Conclusion

Higher-dimensional automata:

- An automaton-like model for non-interleaving concurrency
- ... with a nice language theory!
- The trifecta Kleene–Myhill-Nerode–Büchi-Elgot-Trakhtenbrot is now complete for HDAs [LMCS 2024]–[FI 2024]–[DLT 2024]
- Next step: recognizability by finite categories?
- Geometry & topology provide plenty of intuition

... and Petri nets:

- HDAs provide concise concurrent semantics for Petri nets
- ... and extensions! (needs partial HDAs)

... and real time?

- Real-time concurrent semantics of timed extensions of Petri nets
- HD timed automata seem rather well-behaved
- relation to signals (and STL etc.)