Higher-Dimensional Timed Automata for Real-Time Concurrency

Uli Fahrenberg
with A.Amrane, H.Bazille, E.Clement, and P.Schlehuber

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

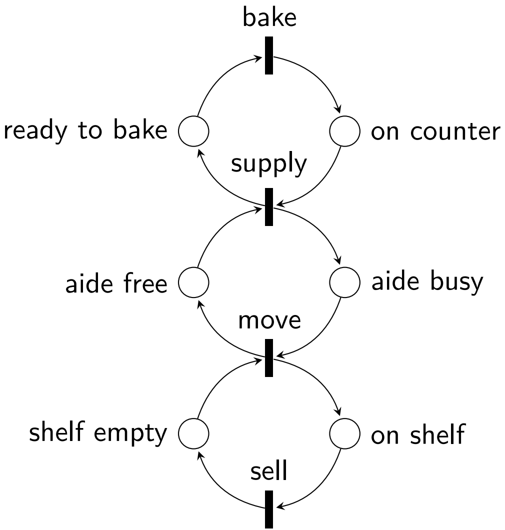
MF60, 28 February 2025, Oldenburg

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

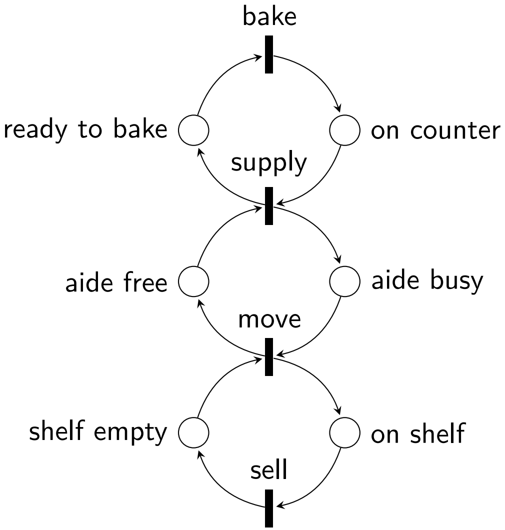
- very useful for modeling distributed or concurrent systems
- invented in 1962; ubiquitous in modeling



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

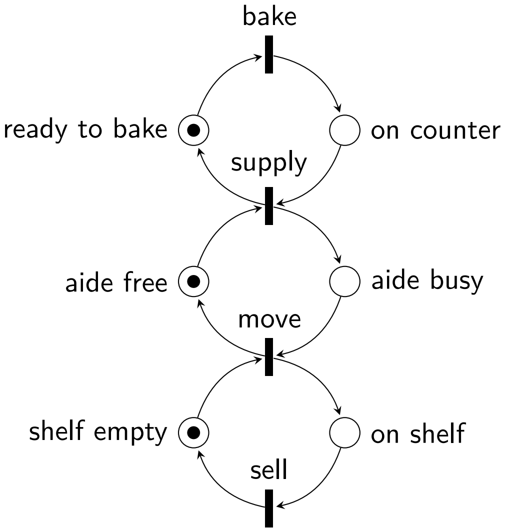


Petri nets

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

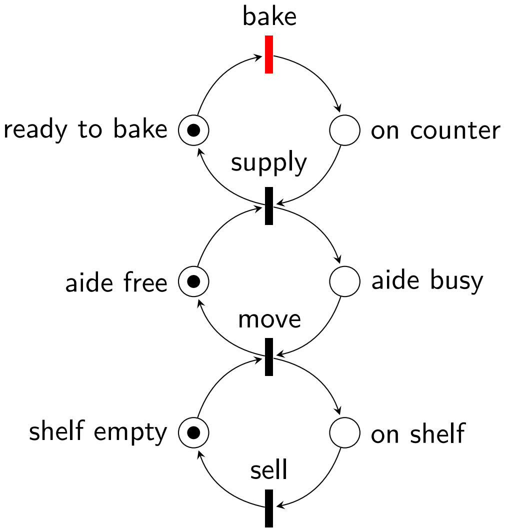
- marking: $S \to \mathbb{N}$:
  number of tokens per place
- preset of t: ${}^\bullet t = F(s, t)$
- postset of t: $t^\bullet(s) = F(t, s)$



Petri nets

A Petri net (S, T, F):

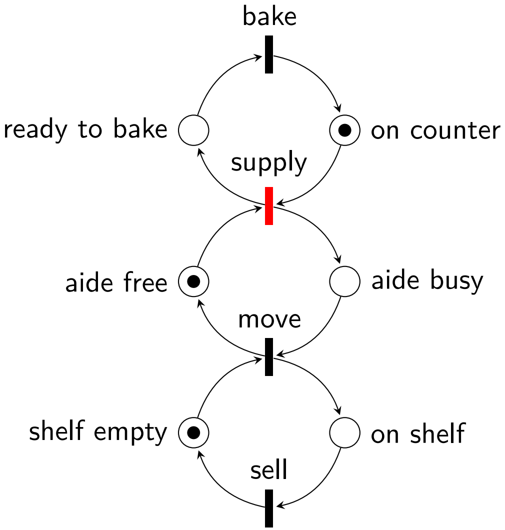
- S set of places
- T set of transitions, $S \cap T = \emptyset$
- $F : S \times T \cup T \times S \to \mathbb{N}$
  weighted flow relation
- marking: $S \to \mathbb{N}$:
  number of tokens per place
- preset of t: ${}^\bullet t = F(s, t)$
- postset of t: $t^\bullet(s) = F(t, s)$
- compute by transforming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



Petri nets

A Petri net (S, T, F):

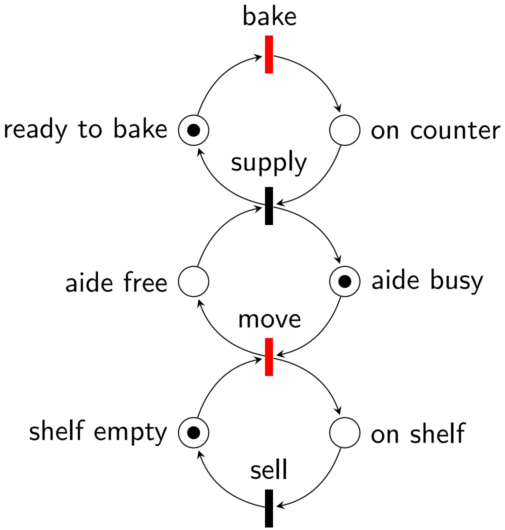
- S set of places
- T set of transitions, $S \cap T = \emptyset$
- $F : S \times T \cup T \times S \to \mathbb{N}$
  weighted flow relation
- marking: $S \to \mathbb{N}$:
  number of tokens per place
- preset of t: ${}^\bullet t = F(s, t)$
- postset of t: $t^\bullet(s) = F(t, s)$
- compute by transforming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

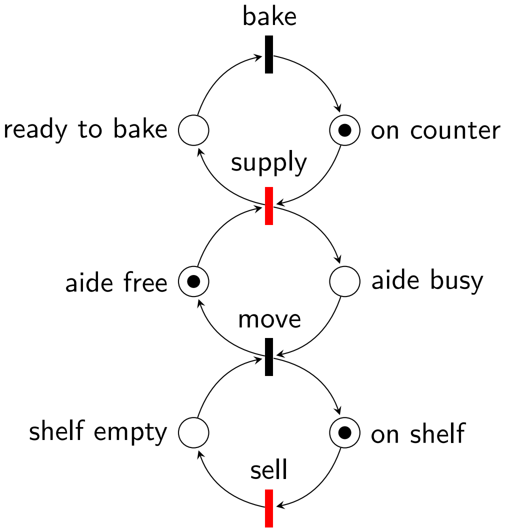
A Petri net (S, T, F):

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

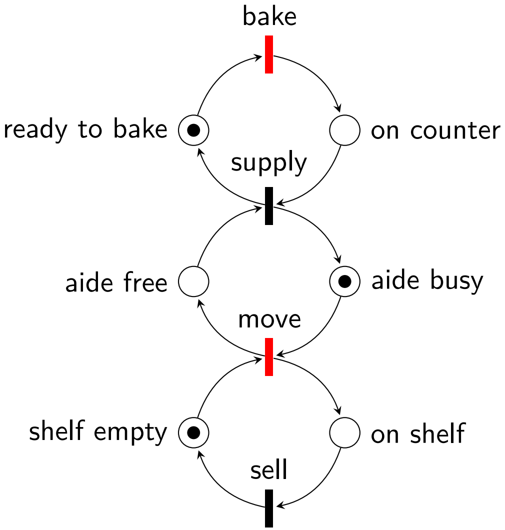
- only if ${}^\bullet t \leq m$



Petri nets

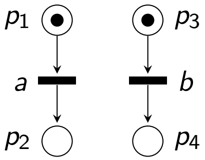
A Petri net (S, T, F):

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



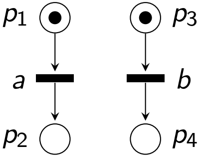
Semantics of Petri nets

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



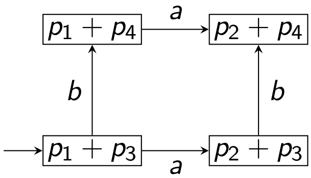
Semantics of Petri nets

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

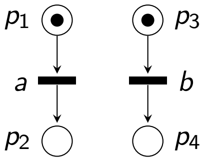


Interleaved semantics (reachability graph) (V, E):

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



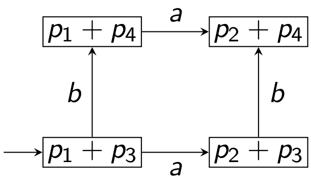
Semantics of Petri nets



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

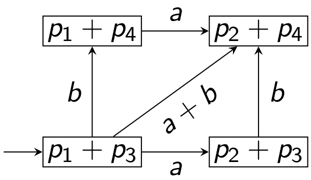
Interleaved semantics (reachability graph) (V, E):

- $V = \mathbb{N}^S$: 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 $\implies$ initial state; take reachable part

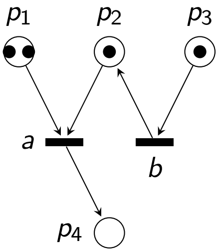


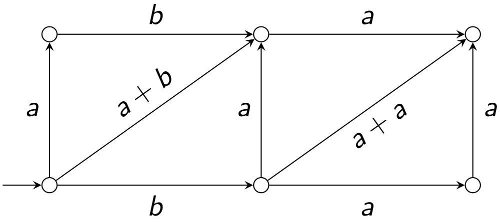
Concurrent step reachability graph (V, E'):

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



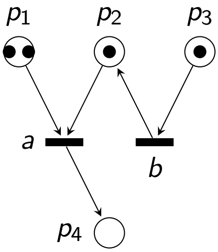
Another example

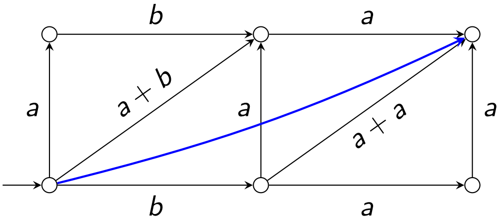




- after firing b, a is auto-concurrent

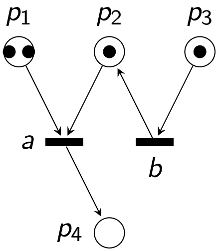
Another example

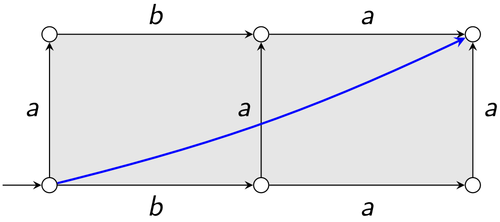




- after firing *b*, *a* is auto-concurrent
- semantics misses some behavior?
  - start *a* – start *b* – finish *b* – start another *a* – etc.

Another example





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

Higher-dimensional automata

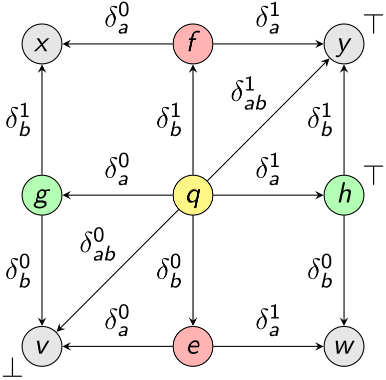
A conclist is a finite, totally ordered, Σ-labeled set. (a list of labeled events)

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



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

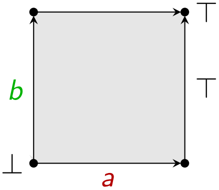
$$X[a] = \{e, f\}$$

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

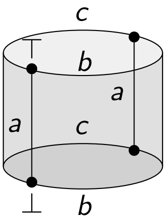
$$X[[\begin{smallmatrix} a \\ b \end{smallmatrix}]] = \{q\}$$

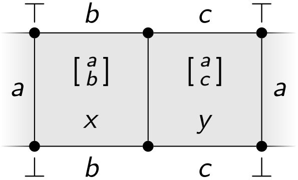
$$\bot_X = \{v\}$$

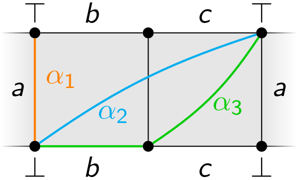
$$\top_X = \{h, y\}$$



Another one

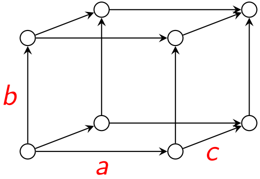




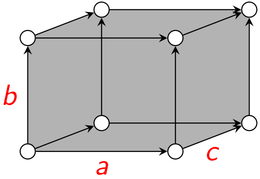


$$a \parallel (bc)^*$$

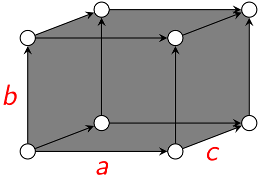
More examples



no concurrency



two out of three



full concurrency

Higher-dimensional automata & concurrency theory

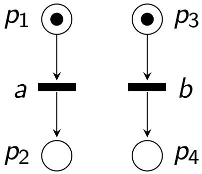
HDAs as a model for concurrency:

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

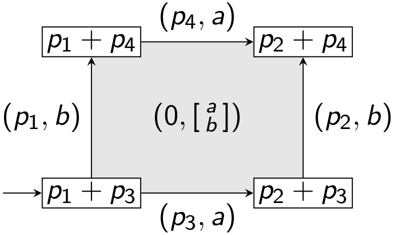
- $E = \{(m, t, m') \mid {}^\bullet t \le m, m' = m - {}^\bullet t + t^\bullet\}$

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

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

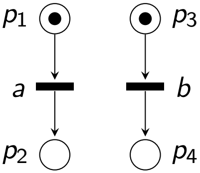
$$\delta^0_{t_i}(x) = (m + {}^\bullet t_i, (t_1, \dots, t_{i-1}, t_{i+1}, \dots, t_n))$$
$$\delta^1_{t_i}(x) = (m + t_i^\bullet, (t_1, \dots, t_{i-1}, t_{i+1}, \dots, t_n))$$

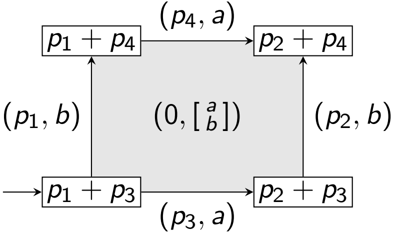
- initial marking $\implies$ initial cell; take reachable part
- (no accepting cells)



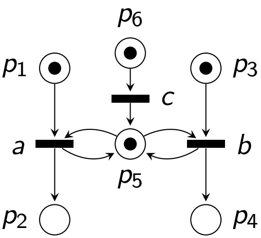
Event order

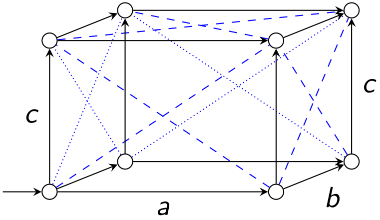
- trouble with symmetry:
  have a cell $(0, [\begin{smallmatrix} a \\ b \end{smallmatrix}])$, but also $(0, [\begin{smallmatrix} b \\ a \end{smallmatrix}])$ (not shown)
- solution: fix an arbitrary order $\preccurlyeq$ on T
- and use $\square = \left\{ \begin{bmatrix} t_1 \\ \vdots \\ t_n \end{bmatrix}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





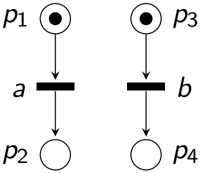
Another one

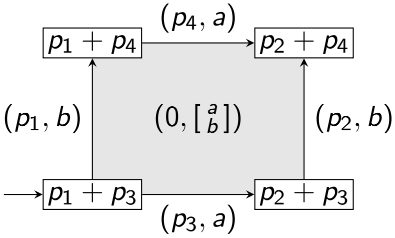




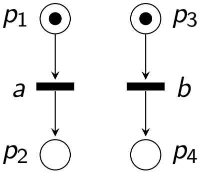
- initially, p_5 is a mutex place: it disables concurrency of a and b
- after c fires, p_5 holds two tokens, so a and b become concurrent
- semantically, a hollow cube without bottom face
- the five faces: front: $(p_3, [\,{}^{a}_{c}\,])$, back: $(p_4, [\,{}^{a}_{c}\,])$
  left: $(p_1, [\,{}^{b}_{c}\,])$, right: $(p_2, [\,{}^{b}_{c}\,])$
  top: $(0, [\,{}^{a}_{b}\,])$

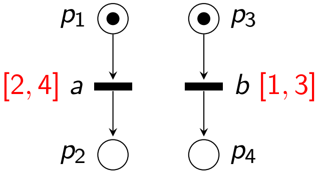
Enter time

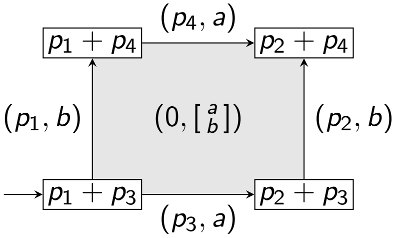




Enter time

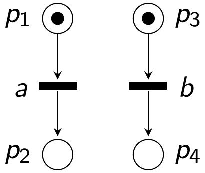

p_1
p_3
a
b
p_2
p_4

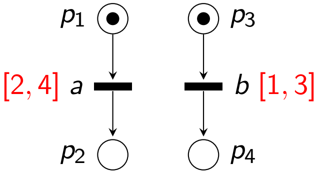

p_1
p_3
$[2,4]$ a
b $[1,3]$
p_2
p_4

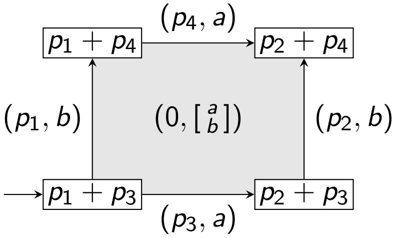

(p_4, a)
$p_1 + p_4$
$p_2 + p_4$
(p_1, b)
$(0, [\,{}^{a}_{b}\,])$
(p_2, b)
$p_1 + p_3$
$p_2 + p_3$
(p_3, a)

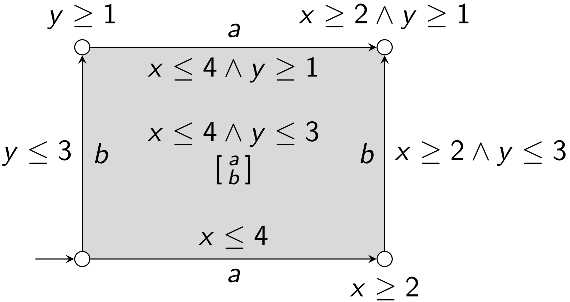
???

Enter 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)
$(0, [\,^a_b\,])$
(p_2, b)
$p_1 + p_3$
$p_2 + p_3$
(p_3, a)


$y \geq 1$
a
$x \geq 2 \wedge y \geq 1$
$x \leq 4 \wedge y \geq 1$
$y \leq 3$
b
$x \leq 4 \wedge y \leq 3$
$[\,^a_b\,]$
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 $[\![A]\!] = (S, \leadsto, S^\bot, S^\top, \rho)$, with $\leadsto \subseteq S \times (\mathsf{St} \cup \mathsf{Te} \cup \mathbb{R}_{\ge 0}) \times S$, given as follows:

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

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

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

Higher-dimensional timed automata

Definition (higher-dimensional timed automaton)

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

starters & terminators

may delay anywhere

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:

exit when leaving

start/terminate A

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

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

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

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

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

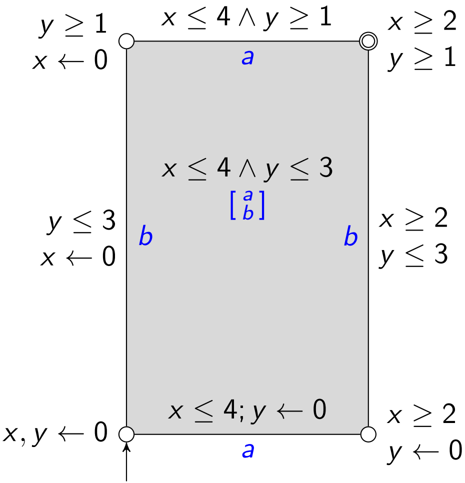
Actions take time?

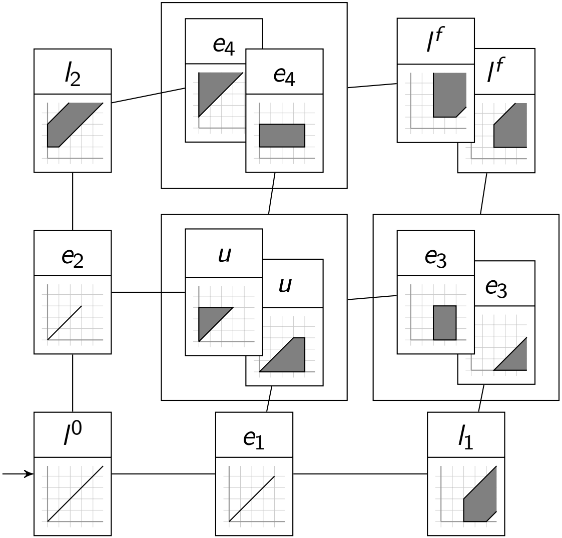
- Cardelli 1982 (ICALP): Actions take time.
  - 'We read $p \xrightarrow[t]{a} q$ as "p moves to q performing a for an interval t"'
- since Alur-Dill 1990 (even before?): Actions are immediate.
  - $(l, v) \overset{d}{\rightsquigarrow} (l, v+d) \overset{s}{\rightsquigarrow} (l', v+d)$
- Kim G. Larsen (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 ✓

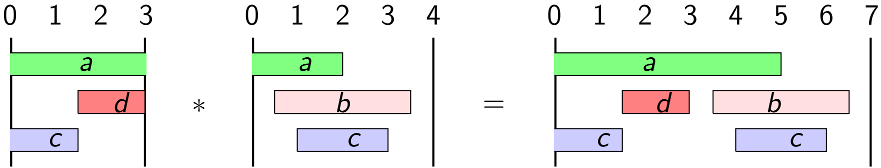
Zone-Based Reachability





Languages of HDTAs

- our "timed words" look (almost) like signals
- with interfaces and a gluing operation



Conclusion

Higher-dimensional automata have shown their versatility by now:

- Languages of Higher-Dimensional Automata [MSCS 2021]
- 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]

- Logic and Languages of Higher-Dimensional Automata [DLT 2024]
- Bisimulations and Logics for Higher-Dimensional Automata [ICTAC 2024]
- Petri Nets and Higher-Dimensional Automata [arxiv 2025]

Higher-dimensional timed automata: just the beginning, but looks good!

- Higher-Dimensional Timed and Hybrid Automata [LITES 2022]
- Languages of Higher-Dimensional Timed Automata [PETRI NETS 2024]