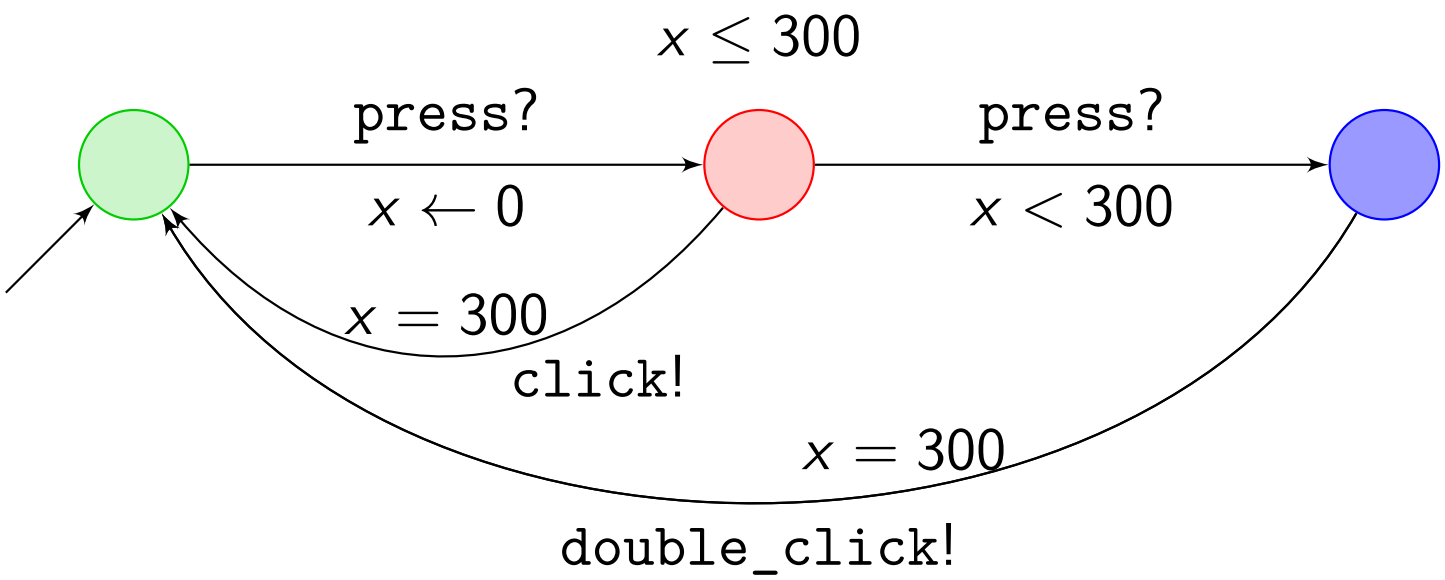
Higher-Dimensional Timed Automata

Uli Fahrenberg

École polytechnique, Palaiseau, France

GETCO September 2018

Recall Timed Automata



Recall Timed Automata

Definition

The set $\Phi(C)$ of clock constraints ϕ over a finite set C is defined by the grammar

$$\phi ::= x \bowtie k \mid \phi_1 \wedge \phi_2 \qquad (x, y \in C, k \in \mathbb{Z}, \bowtie \in \{\leq, <, \geq, >\}).$$

Definition

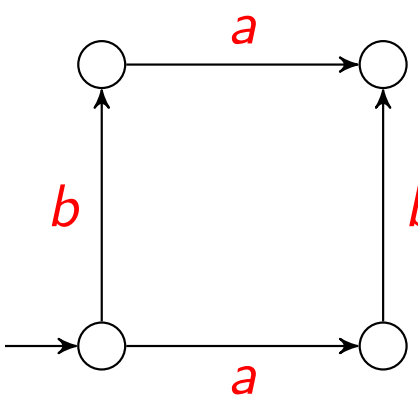
A timed automaton is a tuple $(L, \ell_0, C, \Sigma, I, E)$ consisting of a finite set L of locations, an initial location $\ell_0 \in L$, a finite set C of clocks, a finite set Σ of actions, a location invariants mapping $I : L \to \Phi(C)$, and a set $E \subseteq L \times \Phi(C) \times \Sigma \times 2^C \times L$ of edges.

Recall Timed Automata

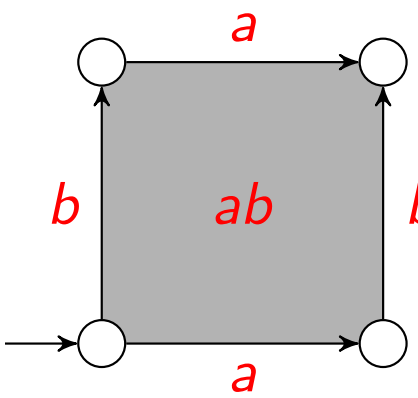
- Useful for modeling synchronous real-time systems
- Reachability, emptiness, LTL model checking PSPACE-complete
- Universality undecidable
- Fast on-the-fly algorithms, using zones, for reachability, liveness, and Timed CTL model checking
- UppAal
- Extensions to weighted timed automata, real-time games, etc.

Recall Higher-Dimensional Automata

$a|b$



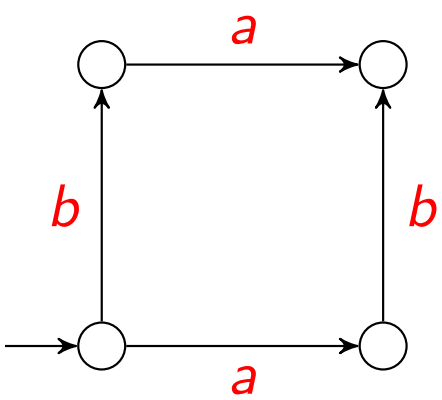
$a.b + b.a$



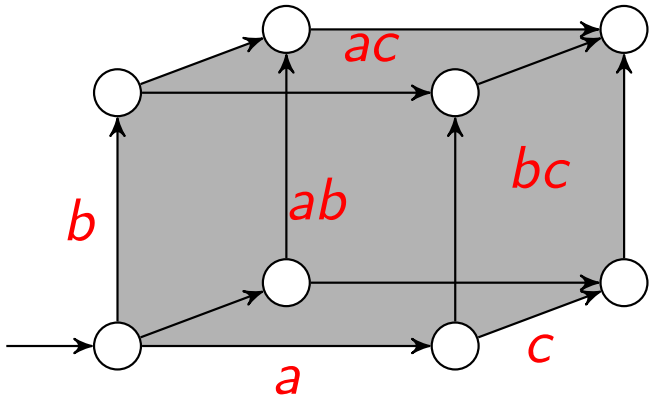
a and b are independent

Recall Higher-Dimensional Automata

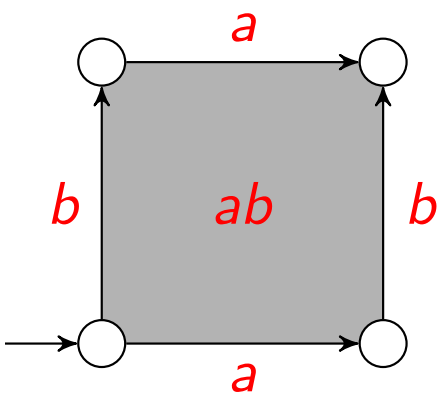
$a|b$

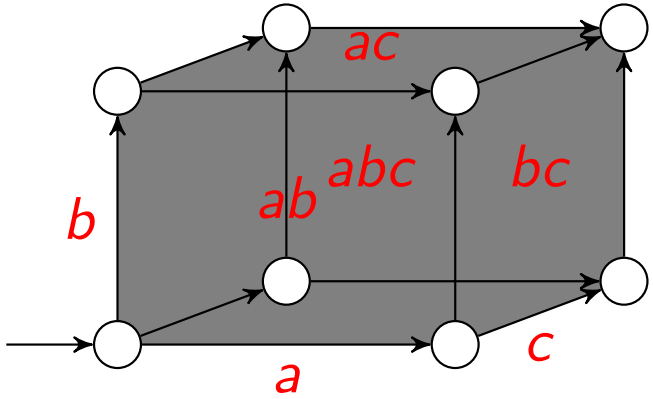


$a|b|c$



$a|b + a|c + b|c$



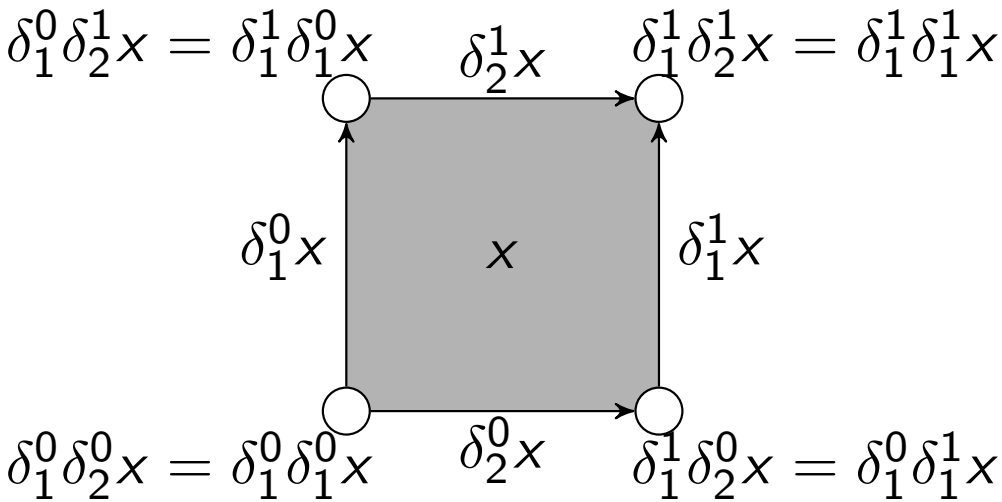


$\{a, b, c\}$ independent

Recall Higher-Dimensional Automata

A precubical set:

- a graded set $X = \{X_n\}_{n \in \mathbb{N}}$
- in each dimension n, $2n$ face maps $\delta_k^0, \delta_k^1 : X_n \to X_{n-1}$ $(k = 1, \ldots, n)$
- the precubical identity: $\delta_k^\nu \delta_\ell^\mu = \delta_{\ell-1}^\mu \delta_k^\nu$ for all $k < \ell$



A higher-dimensional automaton: a finite precubical set with initial state and accepting states

Recall Higher-Dimensional Automata

HDA as a model for concurrency:

- points $x \in X_0$: states
- edges $a \in X_1$: transitions (labeled with events)
- n-squares $\alpha \in X_n$ ($n \geq 2$): independency relations (concurrently executing events)

van Glabbeek (TCS 2006): Up to history-preserving bisimilarity, HDA generalize "the main models of concurrency proposed in the literature"

- (for example Petri nets)

The Marriage between Real Time and Concurrency

- In real-time formalisms, everything is synchronous
  - timed automata, timed Petri nets, hybrid automata, etc.
- and concurrency is interleaving
- In formalisms for (non-interleaving) concurrency, no real time
  - same for distributed computing theory
  - (Petri nets have a concurrent semantics; timed Petri nets don't)
- Our goal: formalisms for real-time concurrent systems
- Application: for example distributed cyber-physical systems
- Here: the marriage between timed and higher-dimensional automata

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 (many personal discussions): Actions are immediate because of technical resaons only. ("We know how to do.")
- 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!

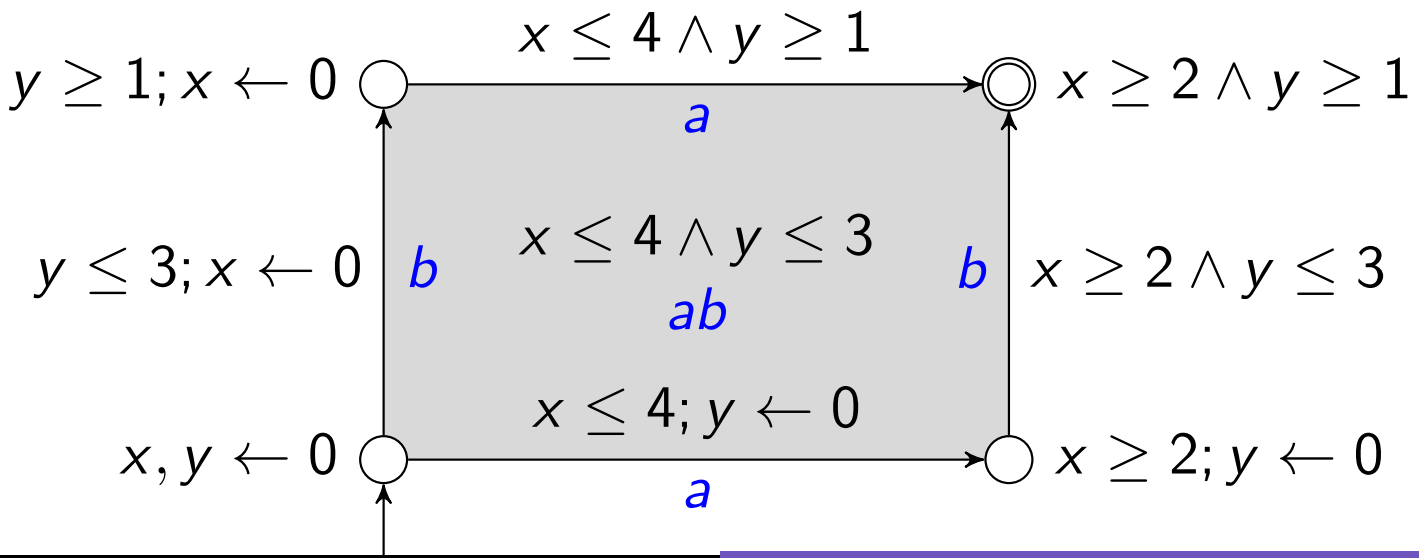
Higher-Dimensional Timed Automata

Definition

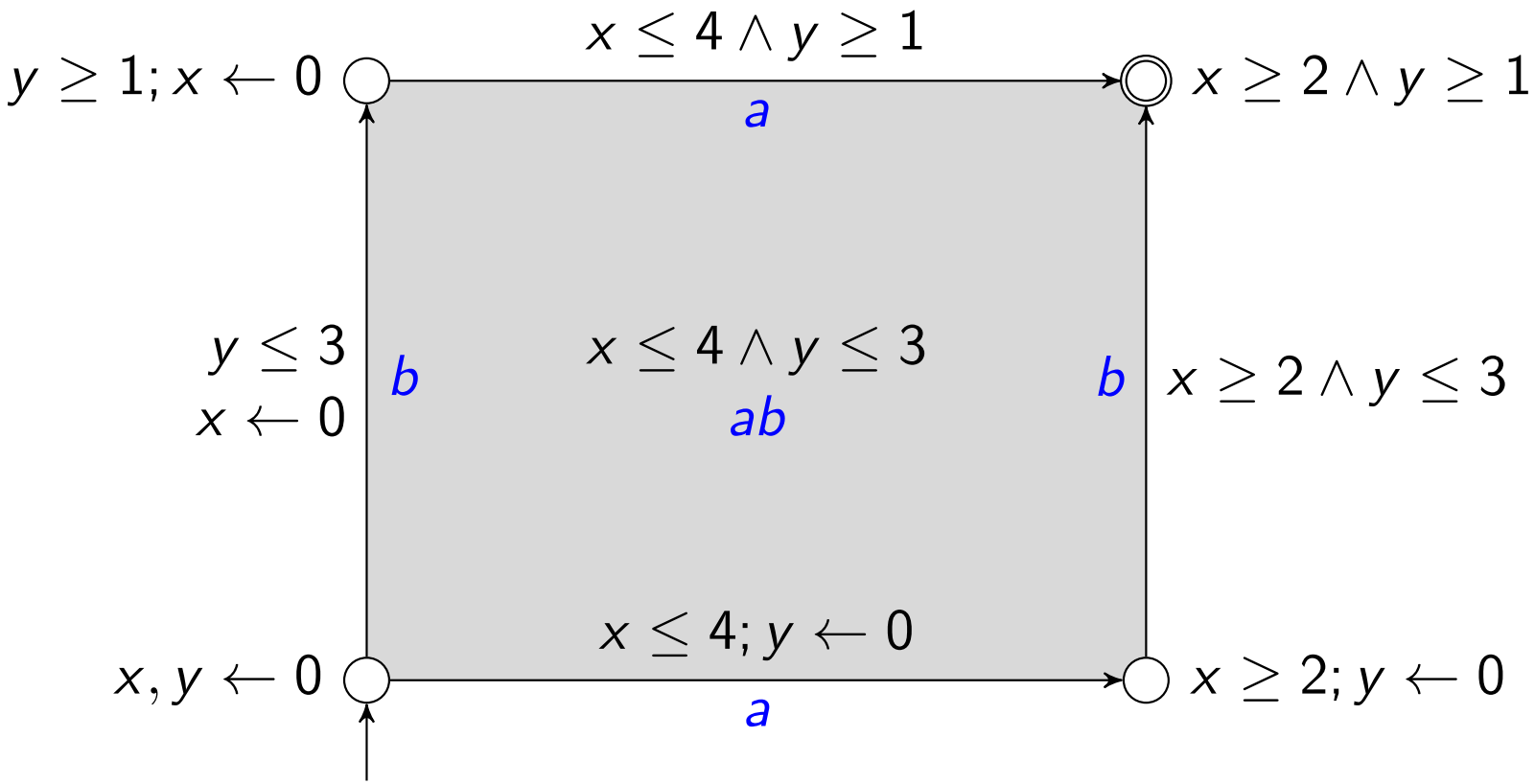
A HDTA is a structure $(L, l^0, L^f, \lambda, C, \mathrm{inv}, \mathrm{exit})$, where (L, l^0, L^f, λ) is a finite HDA, C is a finite set of clocks, and $\mathrm{inv} : L \to \Phi(C)$, $\mathrm{exit} : L \to 2^C$ give invariant and exit conditions for each n-cube.

Intuition:

- $\mathrm{inv}(l)$: conditions on the clock values while delaying in l
- $\mathrm{exit}(l)$: clocks to be reset to 0 when leaving l.

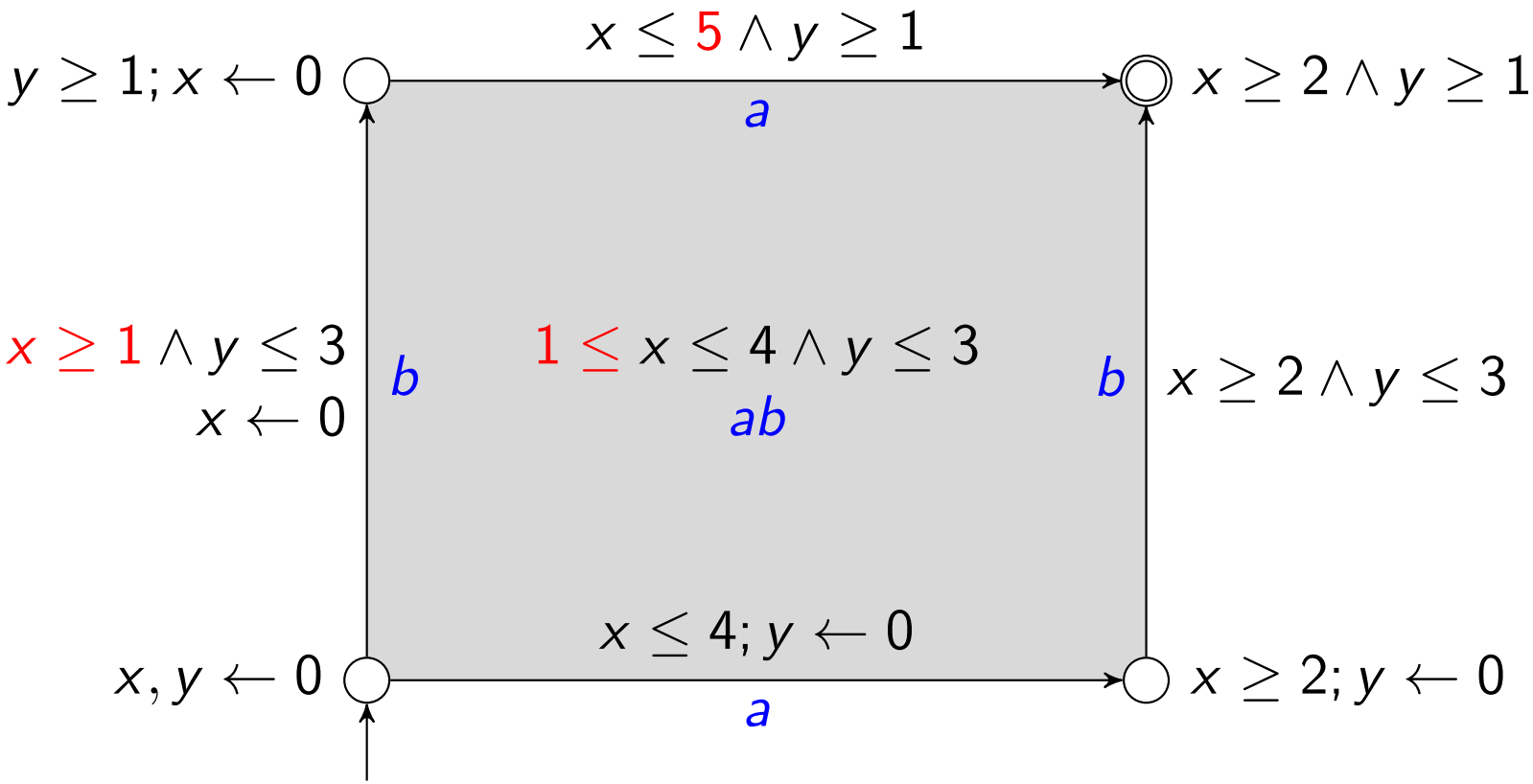


Examples



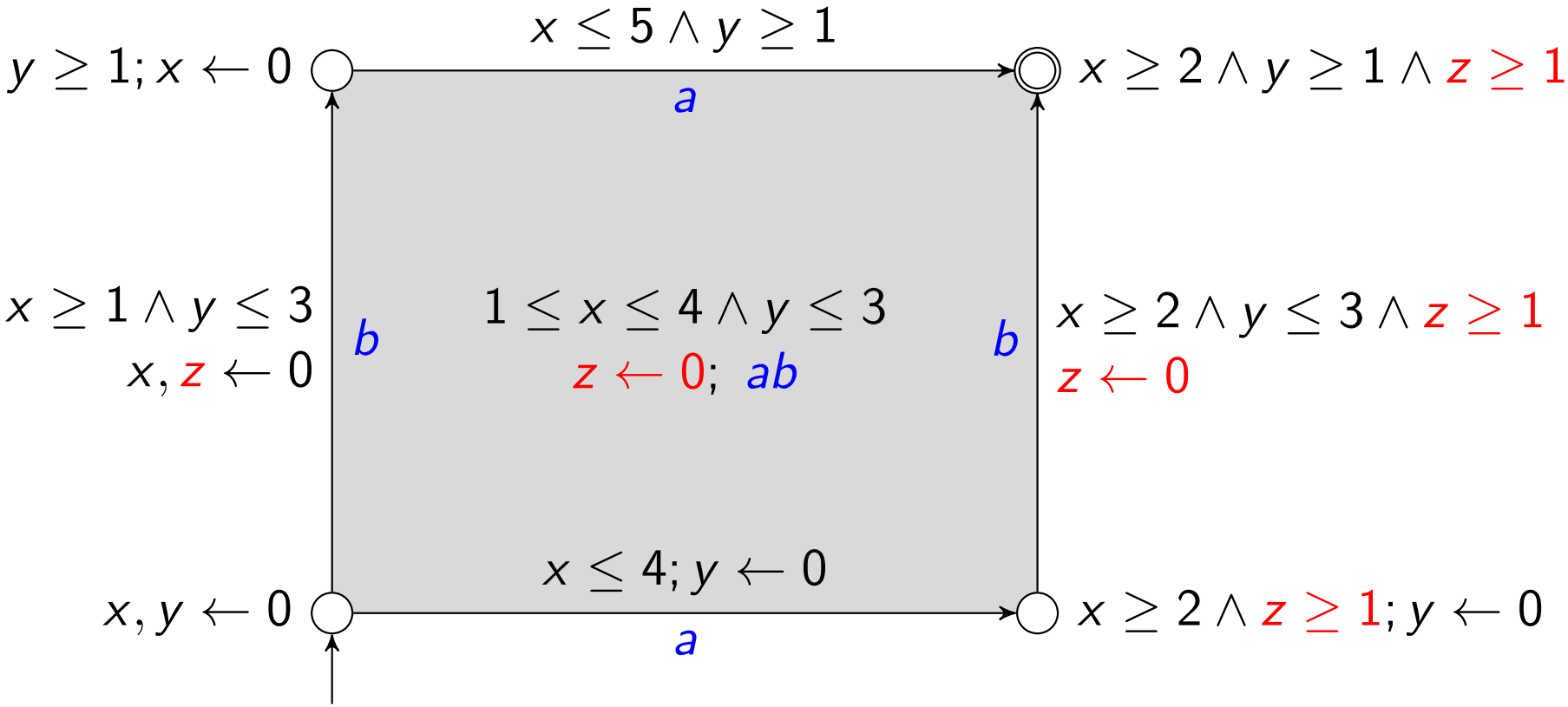
- a takes $[2, 4]$ time units, b takes $[1, 3]$ time units

Examples



- a takes $[2, 4]$ time units, b takes $[1, 3]$ time units
- unless b is done before a
- b can only start 1 time unit after a

Examples

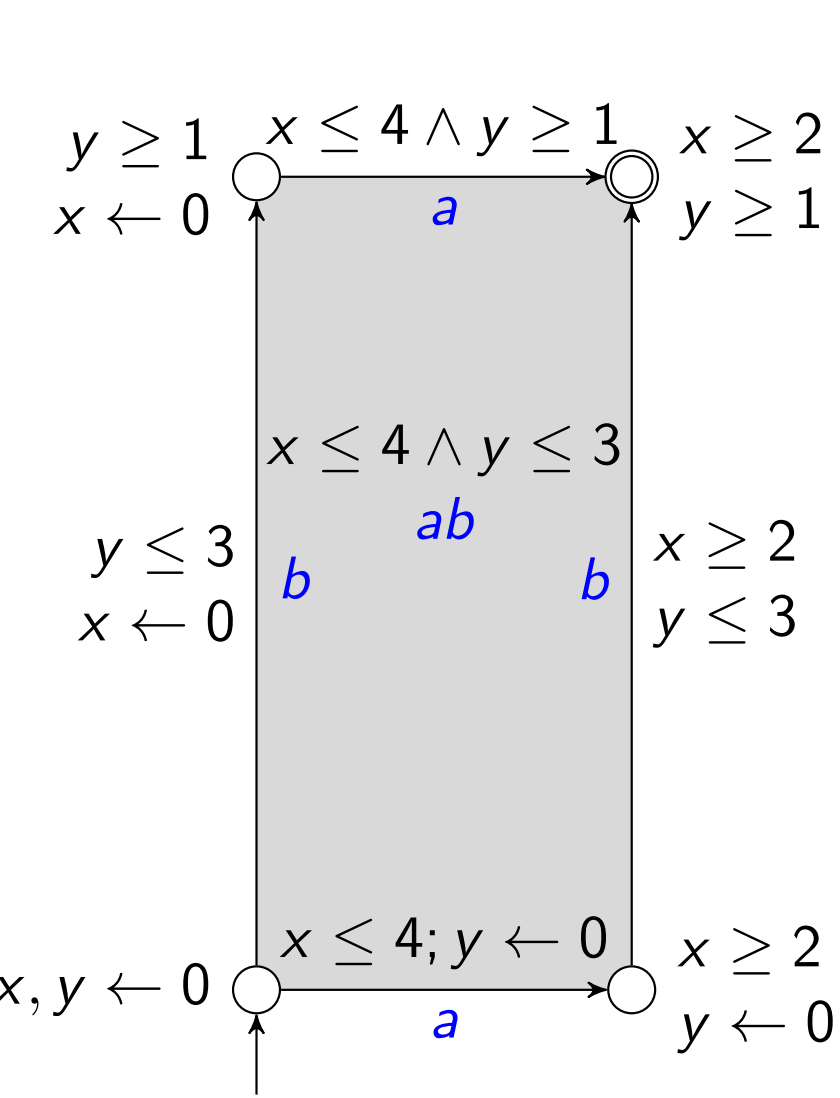


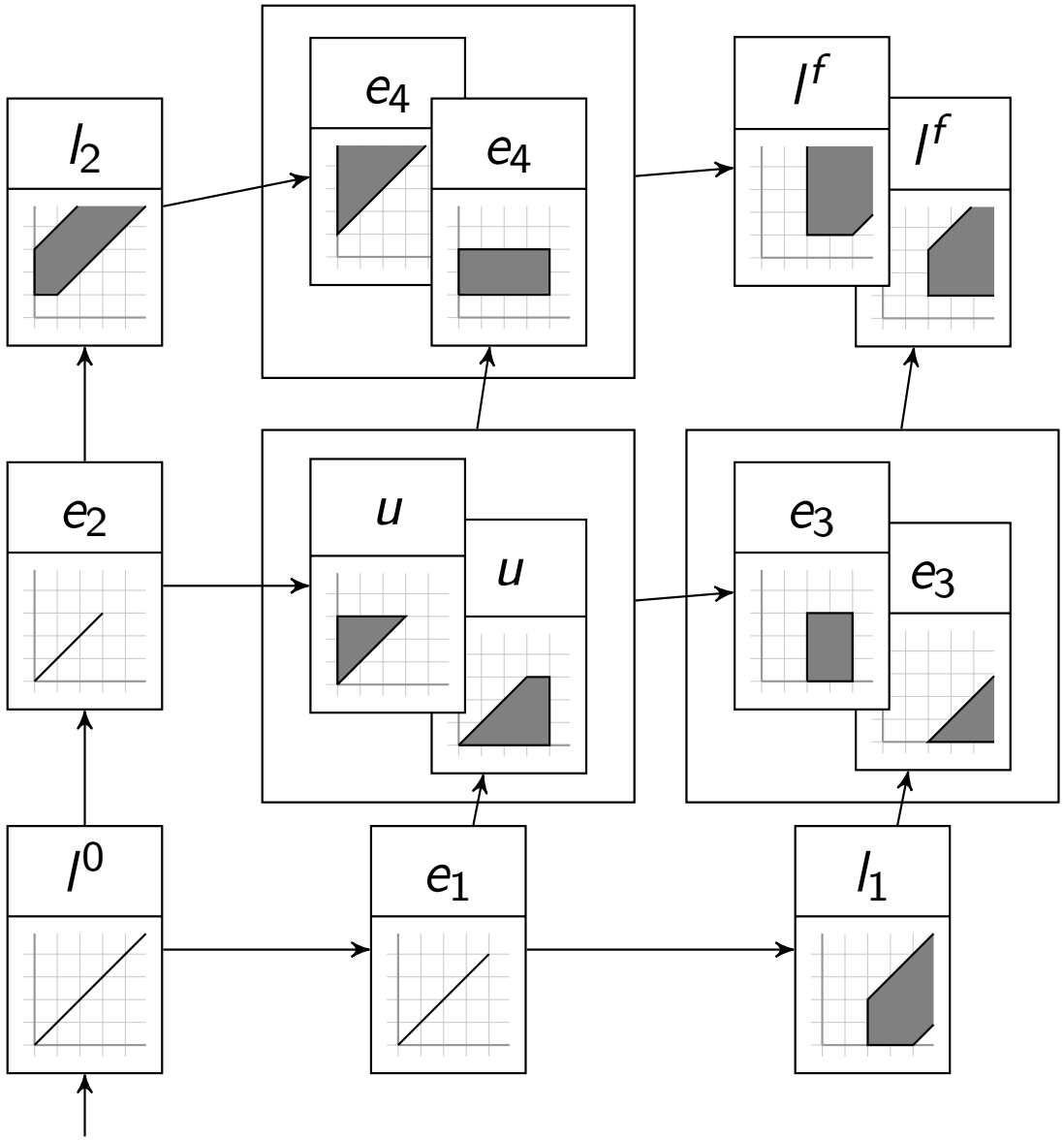
- a takes $[2, 4]$ time units, b takes $[1, 3]$ time units
- b can only start 1 time unit after a
- b has to finish 1 time unit before a

Good News

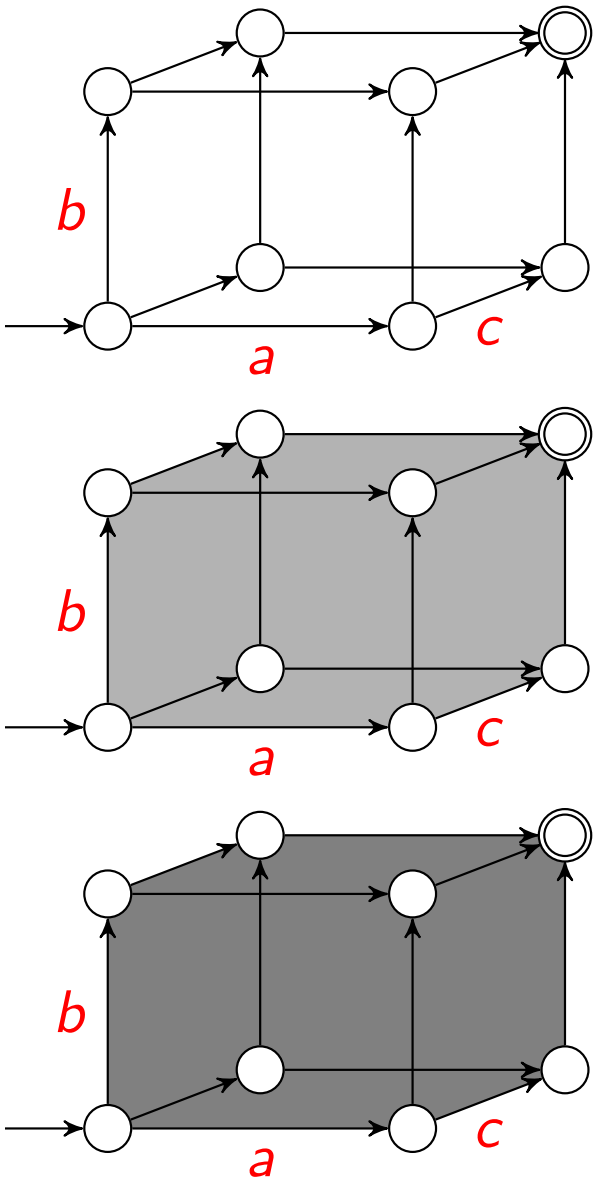
- Reachability for HDTA is PSPACE-complete
- and can be checked using zone-based algorithms
- (Everything works like for timed automata)
- Universality probably still undecidable

Zone-Based Reachability

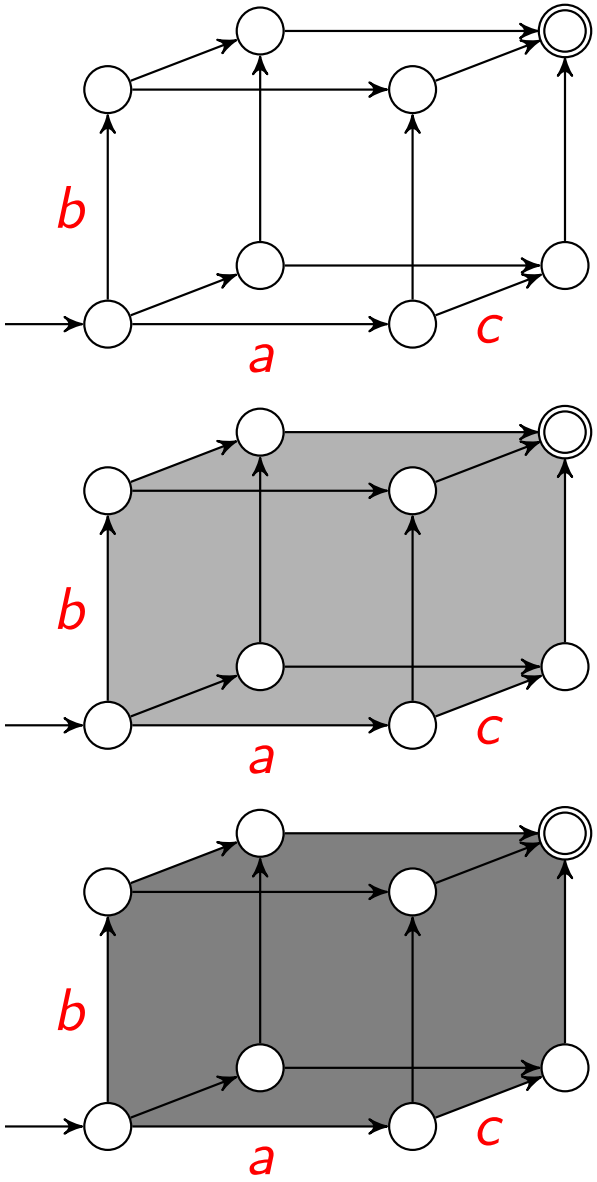




Languages of HDA

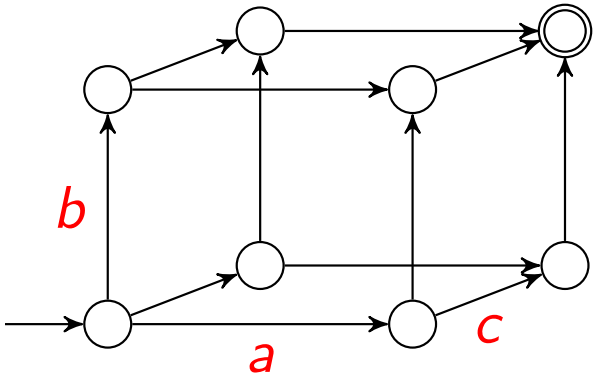


Languages of HDA

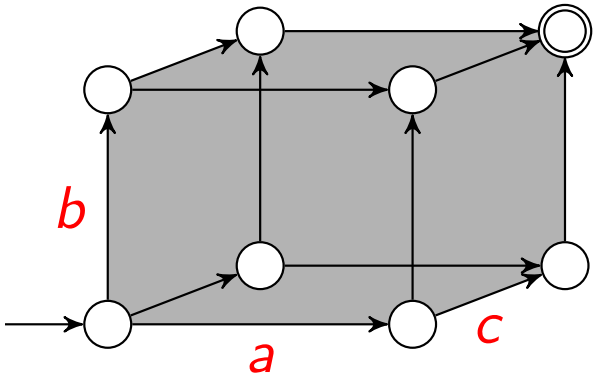


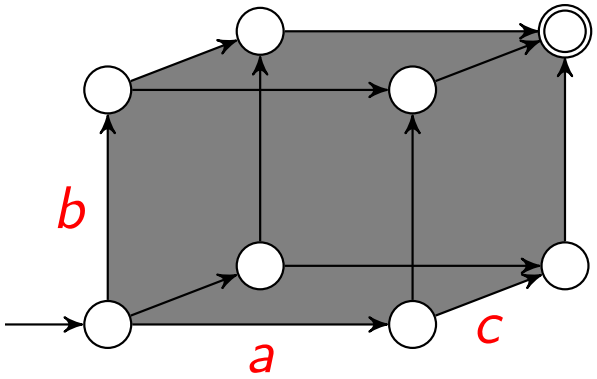
$$L_1 = \{abc, acb, bac, bca, cab, cba\}$$

Languages of HDA



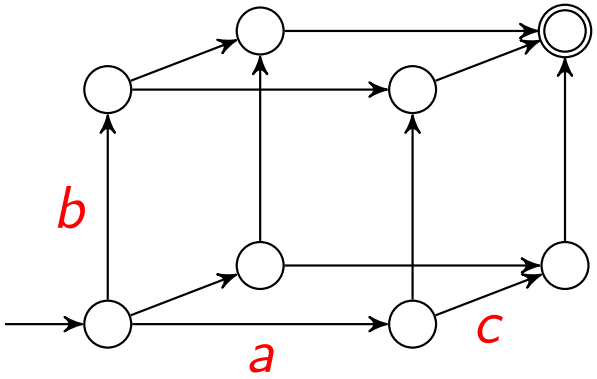
$$L_1 = \{abc, acb, bac, bca, cab, cba\}$$



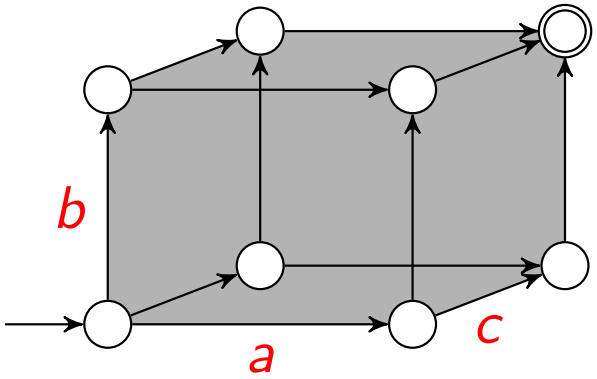


$$L_3 = \left\{ \begin{pmatrix} a \\ b \\ c \end{pmatrix}, \ldots \right\}$$

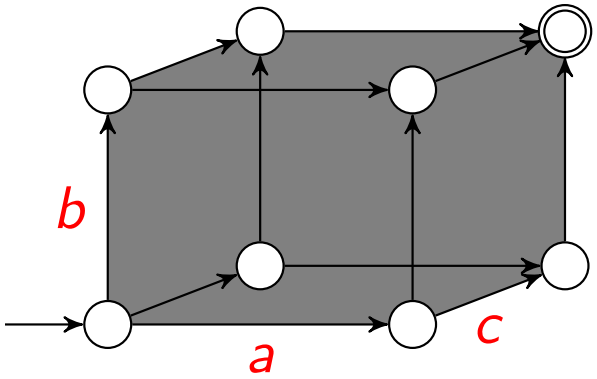
Languages of HDA



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

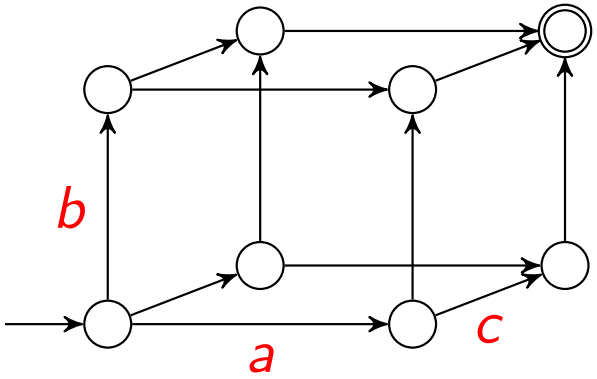


$$L_2 = \left\{ \begin{pmatrix} a \\ b \to c \end{pmatrix}, \begin{pmatrix} a \\ c \to b \end{pmatrix}, \begin{pmatrix} b \\ a \to c \end{pmatrix}, \begin{pmatrix} b \\ c \to a \end{pmatrix}, \begin{pmatrix} c \\ a \to b \end{pmatrix}, \begin{pmatrix} c \\ b \to a \end{pmatrix}, \dots \right\}$$

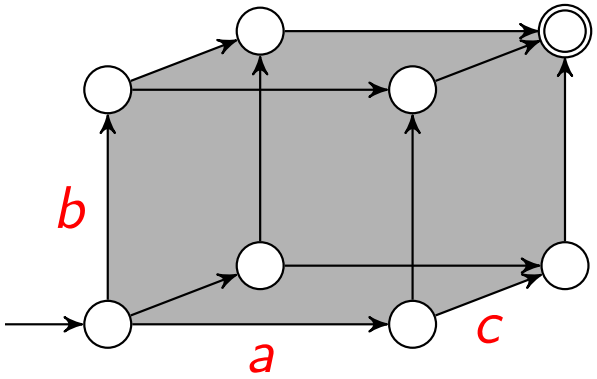


$$L_3 = \left\{ \begin{pmatrix} a \\ b \\ c \end{pmatrix}, \dots \right\}$$

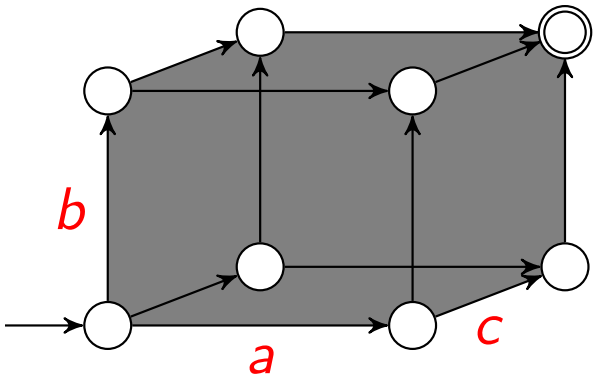
Languages of HDA



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



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



sets of pomsets

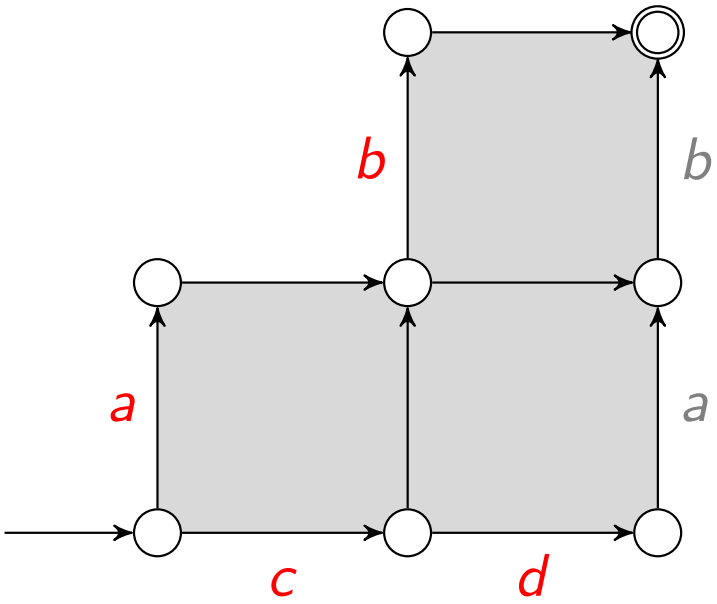
$$L_3 = \left\{ \begin{pmatrix} a \\ b \\ c \end{pmatrix} \right\} \cup L_2$$

Pomsets

A (finite) pomset ("partially ordered multiset") $(P, \leq, \ell)$:

- a finite partially ordered set $(P, \leq)$
- with labeling $\ell : P \to \Sigma$
- (AKA labeled partial order)
- [Lamport 1978]

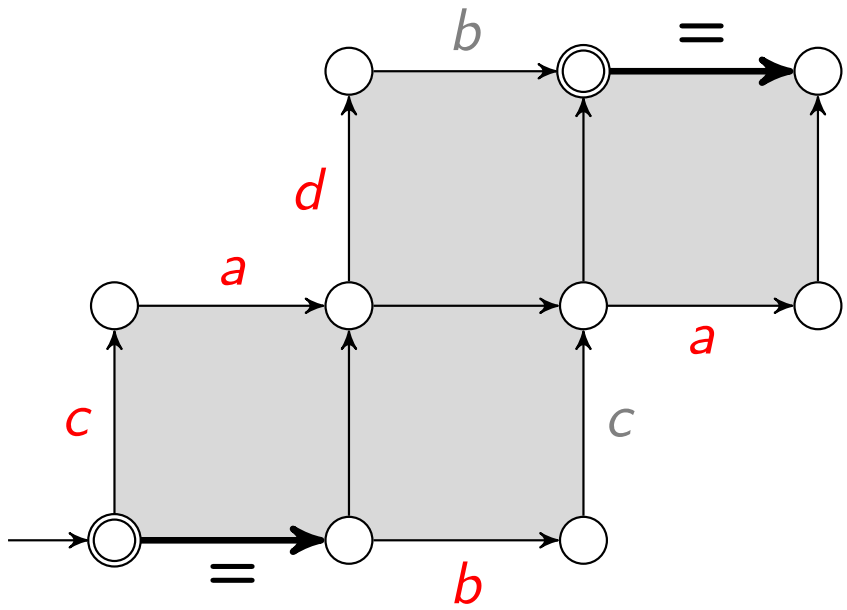
Example



$$\begin{pmatrix} a \longrightarrow b \\ c \longrightarrow d \end{pmatrix}$$

- not series-parallel!

A loop



$$\begin{pmatrix} a \to b \\ c \to d \end{pmatrix}$$

$$\begin{pmatrix} a \to b \to a \to b \\ c \to d \to c \to d \end{pmatrix}$$

$$\begin{pmatrix} a \to b \to a \to b \to a \to b \\ c \to d \to c \to d \to c \to d \end{pmatrix}$$

$\ldots$

Are all pomsets generated by HDA?

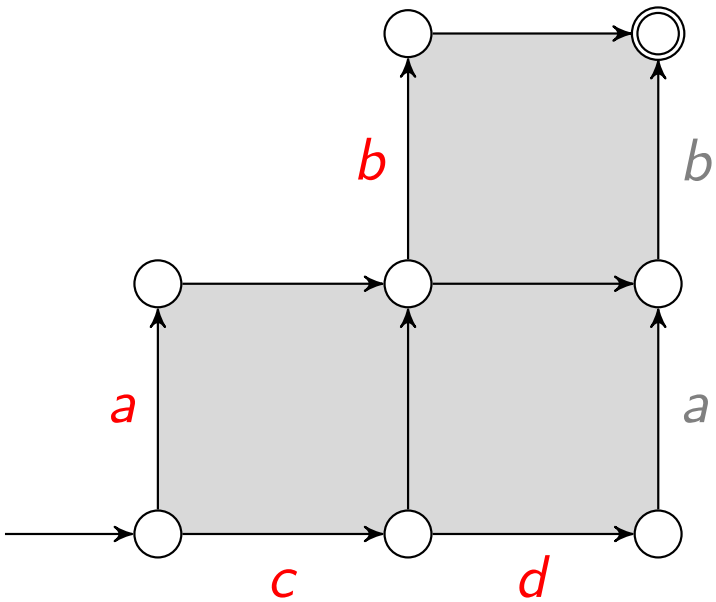
No, only (labeled) interval orders

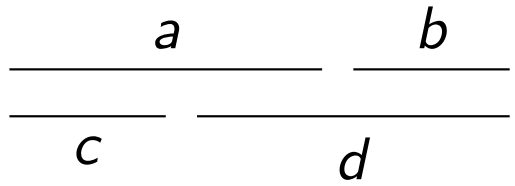
- Poset $(P, \leq)$ is an interval order iff it does not contain ($\rightrightarrows$)
  - (iff it is "2+2-free")
- iff it has an interval representation:
  - a set $I = \{[l_i, r_i]\}$ of real intervals
  - with order $[l_i, r_i] \preceq [l_j, r_j]$ iff $r_i \leq l_j$
  - and an order isomorphism $(P, \leq) \leftrightarrow (I, \preceq)$
- [Fishburn 1970]

Are all pomsets generated by HDA?

No, only (labeled) interval orders

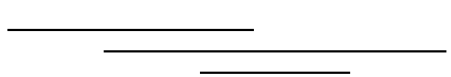
- Poset $(P, \leq)$ is an interval order iff it does not contain ($\rightrightarrows$)
  - (iff it is "2+2-free")
- iff it has an interval representation:
  - a set $I = \{[l_i, r_i]\}$ of real intervals
  - with order $[l_i, r_i] \preceq [l_j, r_j]$ iff $r_i \leq l_j$
  - and an order isomorphism $(P, \leq) \leftrightarrow (I, \preceq)$





$$\begin{pmatrix} a & \longrightarrow & b \\ c & \longrightarrow & d \end{pmatrix}$$

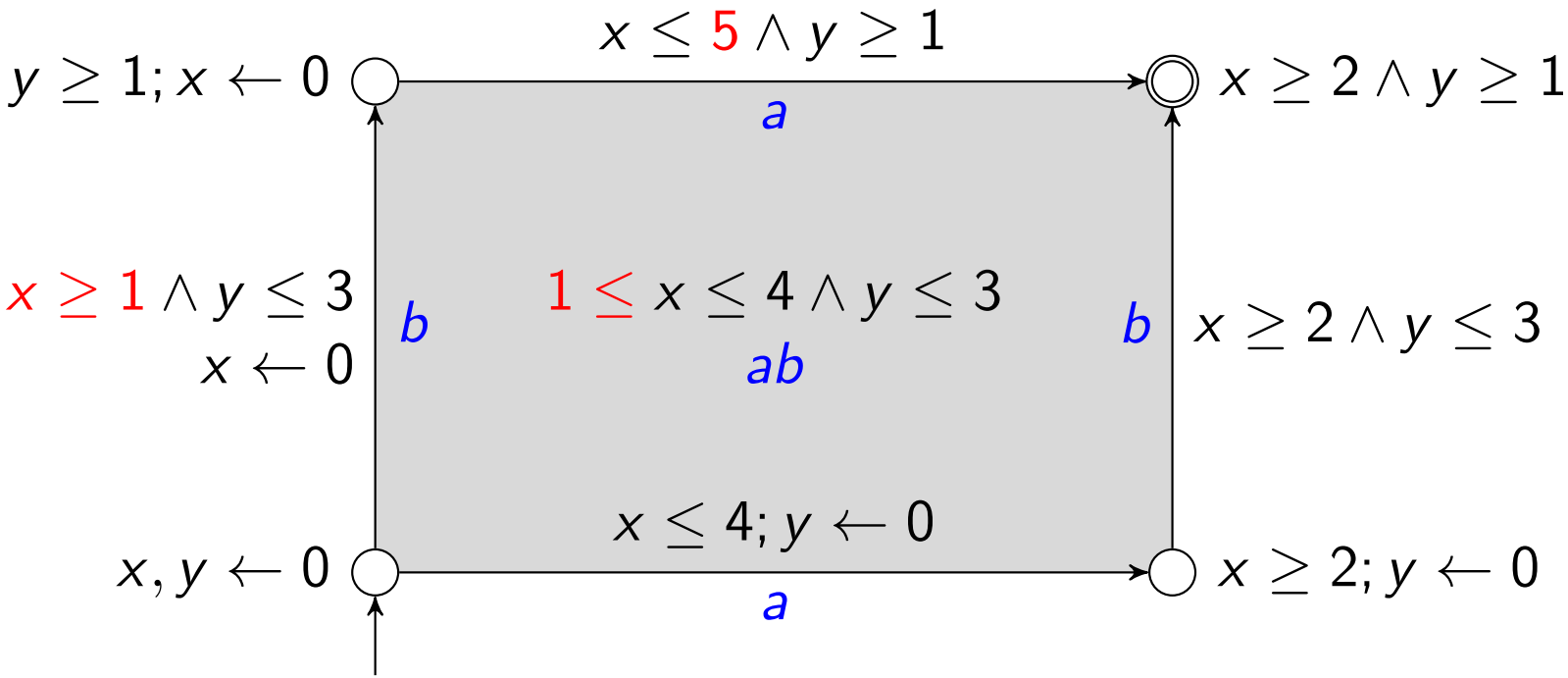
Interval orders vs ST-traces

- An ST-trace: $a^+ b^+ a^+ a^- a^- b^-$ [van Glabbeek 1990]
- as intervals: 
- Lemma: ST-traces up to the equivalence generated by $a^+b^+ \sim b^+a^+$ and $a^-b^- \sim b^-a^-$ are in bijection with interval orders.

Languages of HDTA

- Timed automata generate timed words (w, t):
  - $w = w_1 \dots w_n \in \Sigma^*$
  - $t = (t_1, \dots, t_n) \in \mathbb{R}^n_{\geq 0}$ increasing sequence of time stamps
  - example: $\begin{pmatrix} a & c & a & a \\ .7 & 1.1 & 1.1 & 1.7 \end{pmatrix}$
- Higher-dimensional automata generate labeled interval orders (I, ℓ):
  - $I = \{[l_i, r_i]\} \subseteq \mathbb{N} \times \mathbb{N}$ finite set of intervals $(l_i \leq r_i)$
  - $\ell : I \to \Sigma$
  - example: $\left(\begin{array}{cc} \multicolumn{1}{c}{\overline{}^{\,a}} & \overline{}^{\,b} \\ \underline{}_{\,c} & \multicolumn{1}{c}{\underline{}_{\,a}} \end{array} \right)$
- Proposal: HDTA generate timed interval orders (I, ℓ):
  - $I = \{\!\{[l_i, r_i]\}\!\} \subseteq \mathbb{R}_{\geq 0} \times \mathbb{R}_{\geq 0}$ finite multiset of real intervals $(l_i \leq r_i)$
  - $\ell : I \to \Sigma$

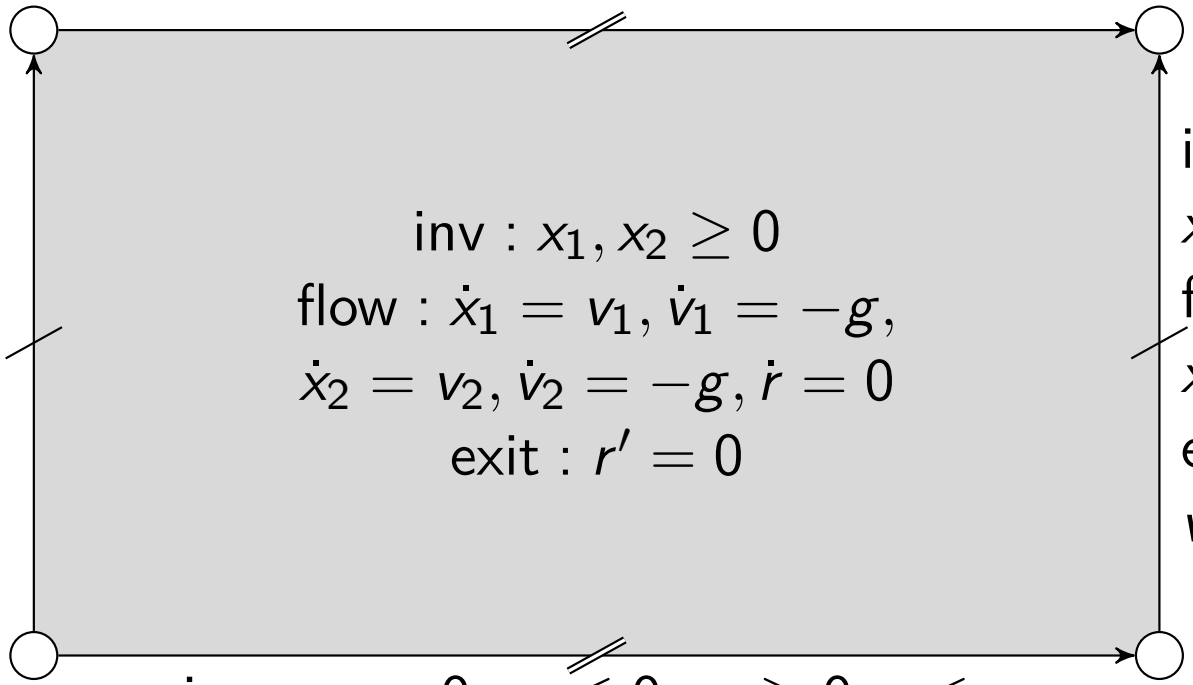
Example



$$L(A) = \left\{ \{\!\{ [l_1, r_1]^a, [l_2, r_2]^b \}\!\} \mid \begin{cases} 1 \leq r_2 - l_2 \leq 3 \\ 2 \leq r_1 - l_1 \leq \begin{cases} 4 & \text{if } r_1 < r_2 \\ 5 & \text{if } r_1 \geq r_2 \end{cases} \\ l_2 \geq l_1 + 1 \end{cases} \right\}$$

Higher-Dimensional Hybrid Automata

Two independently bouncing balls (with temporal regularization):



International Workshop on Methods and Tools for Distributed Hybrid Systems Aalborg, Denmark, 26 August 2017 Associated with MFCS 2017

About

The purpose of DHS is to connect researchers working in *real-time systems*, *hybrid systems*, *control theory*, *distributed computing*, and *concurrency*, in order to advance the subject of **distributed hybrid systems**.

Distributed hybrid systems, or distributed *cyber-physical systems*, are abundant. Many of them are safety-critical, but ensuring their correct functioning is very difficult. We believe that new techniques are needed for the analysis and validation of DHS. More precisely, we believe that convergence and interaction of methods and tools from different areas of computer science, engineering, and mathematics is needed in order to advance the subject.

This first edition of the DHS workshop aims at gathering researchers which work in the above areas in order to facilitate collaboration and discuss how the subject may advance.

Invited Talks

- Alessandro Abate, Oxford University, United Kingdom
  Hybrid models for heterogeneous populations of photovoltaic panels on the grid
- Martin Fränzle, Carl von Ossietzky Universität Oldenburg, Germany
  Indecision and delays are the parents of failures
- Kim G. Larsen, Aalborg Universitet, Denmark
  Synthesis and optimization for cyber-physical systems
- Martin Raussen, Aalborg Universitet, Denmark
  Topological models for spaces of executions in HDA
- Rafael Wisniewski, Aalborg Universitet, Denmark
  Safety verification of stochastic hybrid systems

Second International Workshop on Methods and Tools for Distributed Hybrid Systems Palaiseau, France, 4 July 2018

About

The purpose of DHS is to connect researchers working in *real-time systems*, *hybrid systems*, *control theory*, *distributed computing*, and *concurrency theory*, in order to advance the subject of **distributed hybrid systems**.

The first DHS workshop was held in Aalborg, Denmark, in August 2017, featuring invited talks by Alessandro Abate, Martin Fränzle, Kim G. Larsen, Martin Raussen, and Rafael Wisniewski. This second edition aims to continue the conversation.

Invited Talks

- Luc Jaulin, ENSTA Bretagne, Brest, France
  Distributed localization and control of underwater robots
- Thao Dang, Verimag, Grenoble, France
  Invariance and stability verification of hybrid systems
- Lisbeth Fajstrup, Aalborg University, Denmark
  Symmetries in the PV-model and of directed invariants
- Emmanuel Ledinot, Dassault Aviation, France
  Towards CPS certification reformation: call for effective foundations
- André Platzer, Carnegie Mellon University, United States
  Logic of distributed hybrid systems

Conclusion

- Higher-dimensional timed automata: a nice formalism for real-time concurrency?
- Also, higher-dimensional hybrid automata
- For HDTA verification, zones
- Tensor product for parallel composition
- "Partial-order reduction built in"
- Actions should take time!?