Discretizations of timed automata behaviors

CĂTĂLIN DIMA

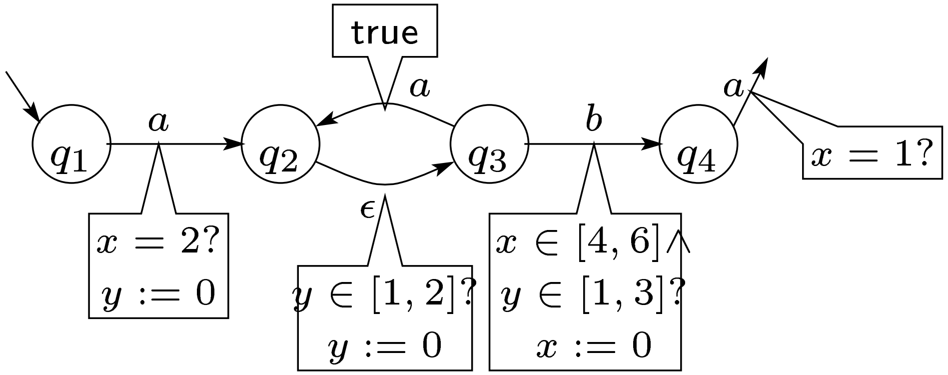
LABRI

Summary of the presentation

- Timed automata and clock relations
- Representing clock relations with n-automata
- Discretizations of timed languages
- Stack automata for discretizations of timed languages

Timed automata

Finite automata with clocks for measuring time passage

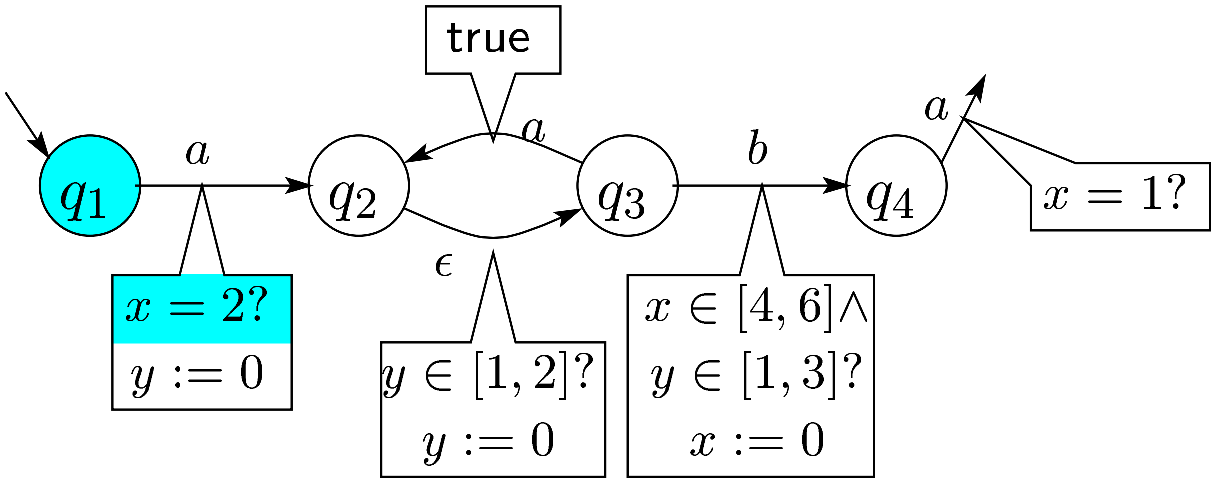


- Time passage in the states.
- Instantaneous transitions.
- Clocks evolve synchronously, at rate 1.

Reset time semantics for timed automata

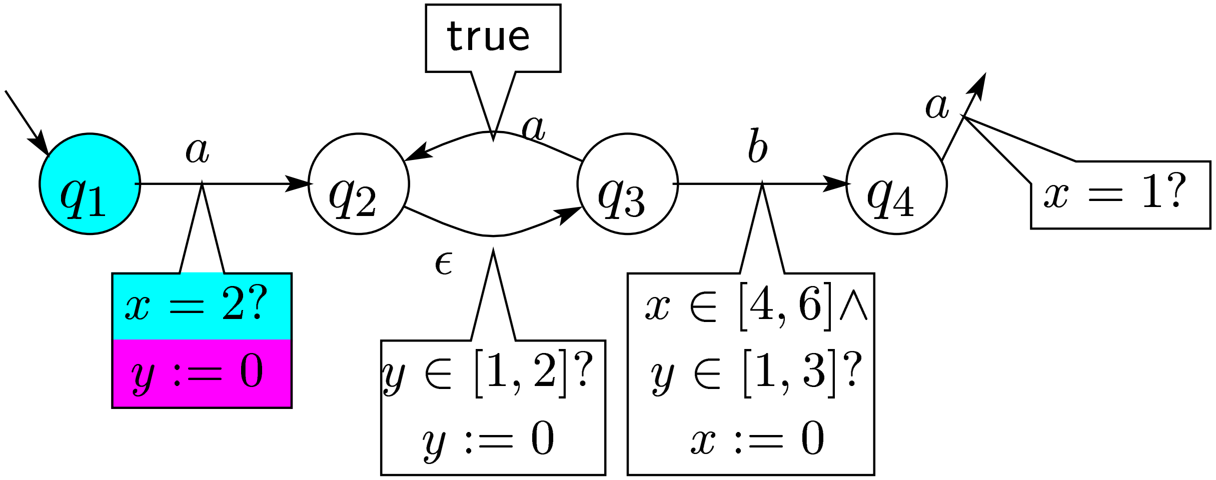
- *time* $=$ the current time.
- For each clock x, $r(x) \;= $ the latest moment before the current time when the clock x was reset.
- Time passage $=$ increment the *time* variable.
- Clock value for $x =$ *time* $- \, r(x)$.
- Reset clock $x \mapsto$ put $r(x) =$ *time*.

Reset time semantics



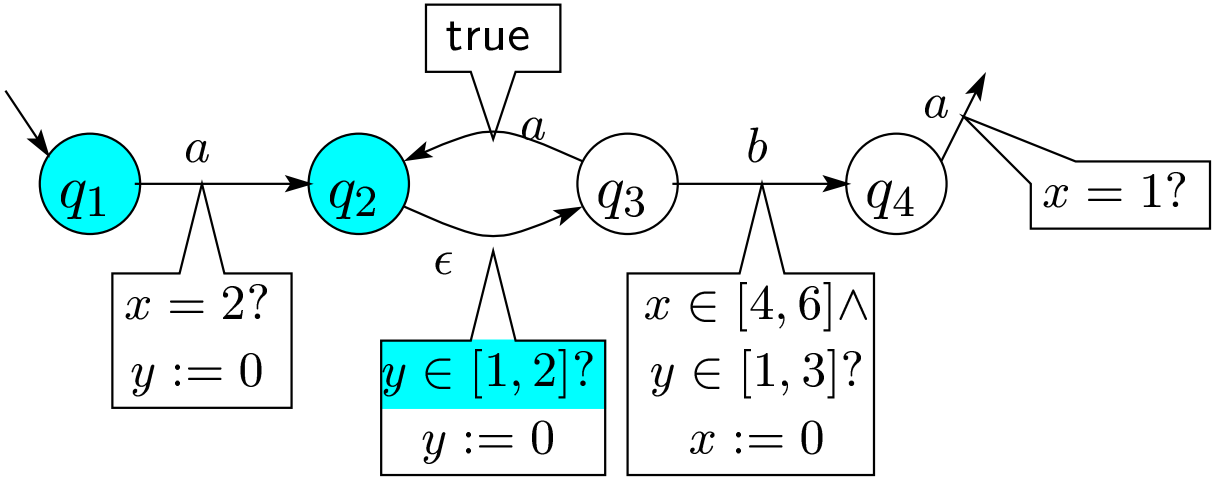
$r(x):$	0	0
$r(y):$	0	0
time :	0	2

Reset time semantics



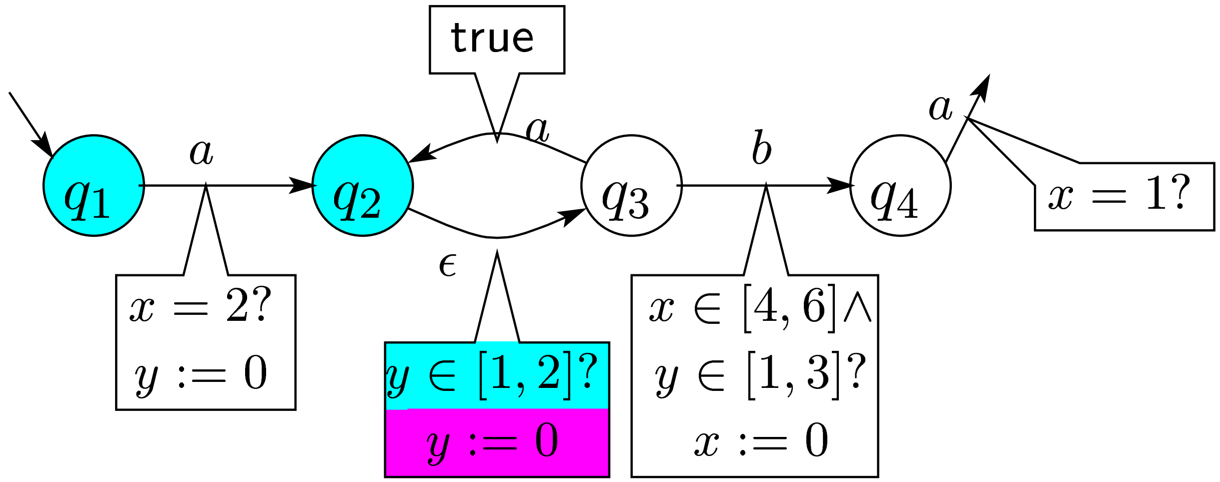
$r(x)$:	0	0
$r(y)$:	0	2
time :	0	2

Reset time semantics



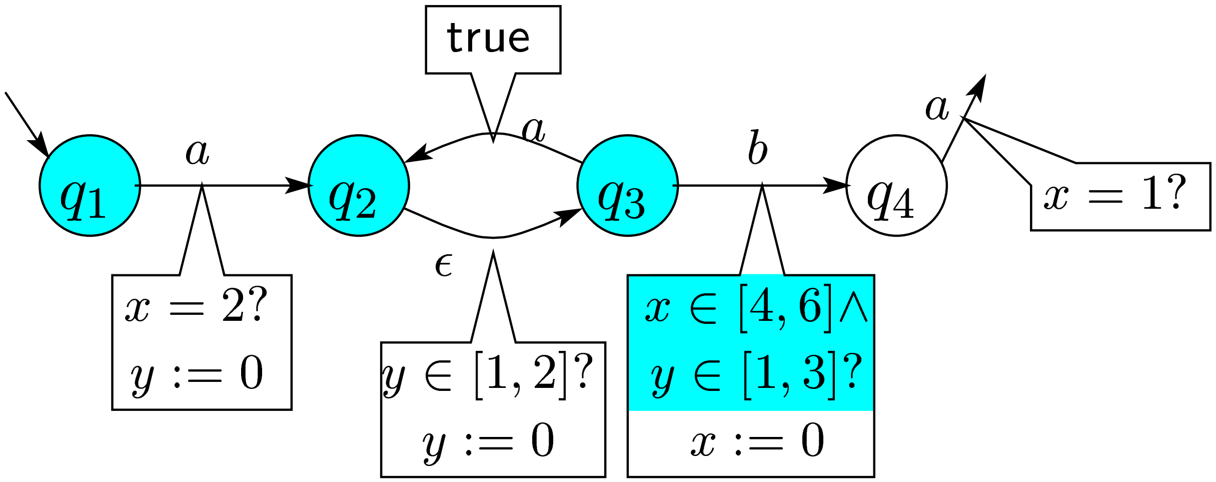
$r(x)$:	0	0	0
$r(y)$:	0	2	2
time :	0	2	3.5

Reset time semantics



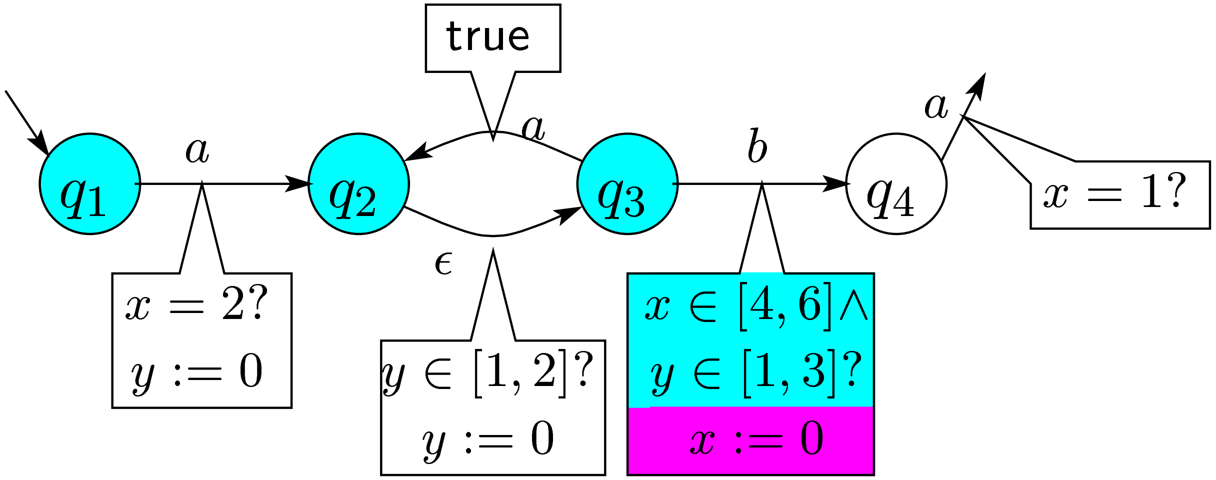
$r(x)$:	0	0	0
$r(y)$:	0	2	3.5
time :	0	2	3.5

Reset time semantics



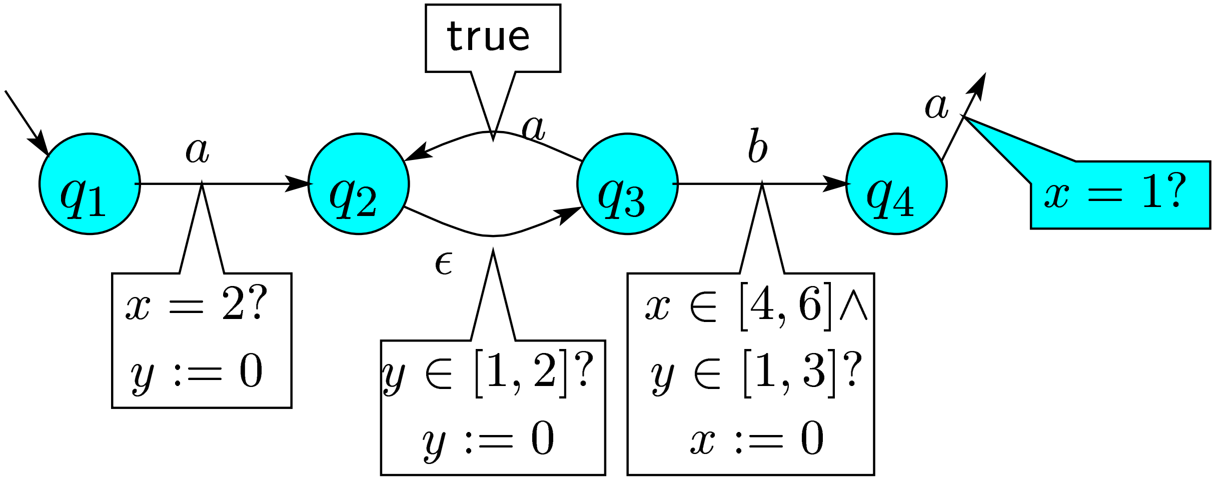
$r(x)$:	0	0	0	0
$r(y)$:	0	2	3.5	3.5
time :	0	2	3.5	5

Reset time semantics



$r(x)$:	0	0	0	5
$r(y)$:	0	2	3.5	3.5
time :	0	2	3.5	5

Reset time semantics



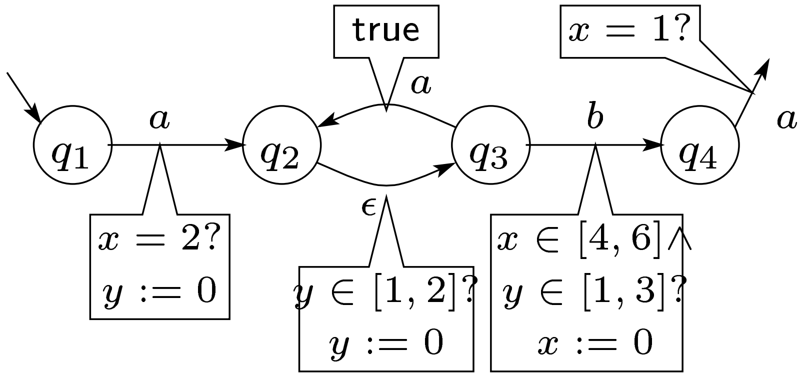
$r(x)$:	0	0	0	5	5
$r(y)$:	0	2	3.5	3.5	3.5
time :	0	2	3.5	5	6

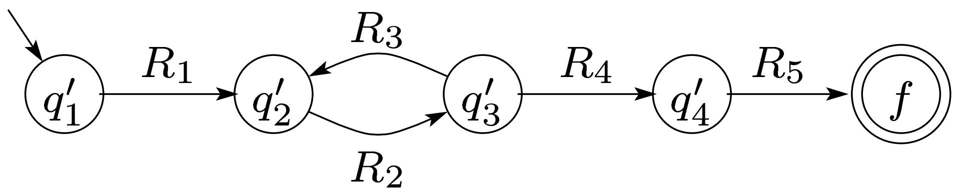
The reachability relation

- One relation associated to each transition.
- Chains of transitions = $\circ$.
- Choices in taking transitions = $\cup$.
- Circuits in the automaton = *.

The reachability relation

- One relation associated to each transition.
- Chains of transitions = $\circ$.
- Choices in taking transitions = $\cup$.
- Circuits in the automaton = *.





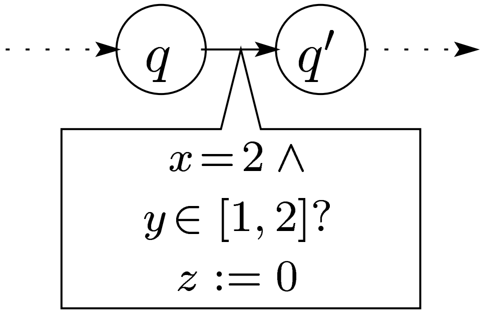
$$R_A = R_1 \circ (R_2 \circ R_3)^* \circ R_2 \circ R_4 \circ R_5$$

The reachability relation

$\overline{x}$ = variable representing the latest reset time for x

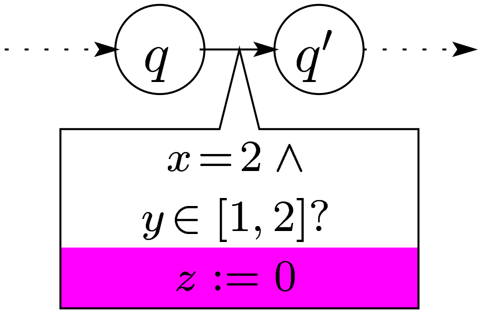
The reachability relation

$\overline{x}$ = variable representing the latest reset time for x



The reachability relation

$\overline{x}$ = variable representing the latest reset time for x



$\textit{time}, \overline{x}, \overline{y}, \overline{z} \leq \textit{time'} \wedge$

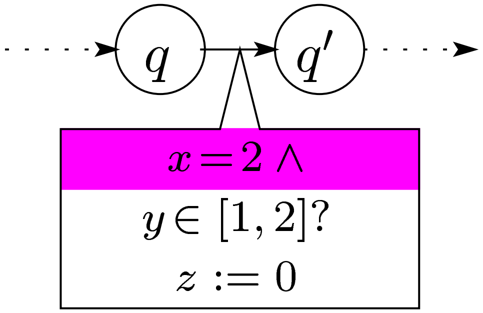
$\overline{x}' = \overline{x} \wedge$

$\overline{y}' = \overline{y} \wedge$

$\overline{z}' = \textit{time}' \wedge$

The reachability relation

$\overline{x}$ = variable representing the latest reset time for x



time, $\overline{x}, \overline{y}, \overline{z} \leq$ *time'* $\wedge$

$\overline{x}' = \overline{x} \wedge$

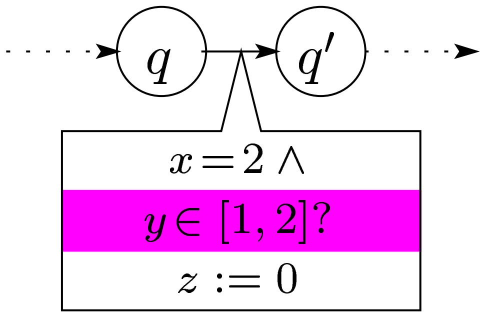
$\overline{y}' = \overline{y} \wedge$

$\overline{z}' = \mathit{time}' \wedge$

$\mathit{time}' - \overline{x} = 2 \wedge$

The reachability relation

$\overline{x}$ = variable representing the latest reset time for x



$\textit{time}, \overline{x}, \overline{y}, \overline{z} \leq \textit{time'} \wedge$

$\overline{x}' = \overline{x} \wedge$

$\overline{y}' = \overline{y} \wedge$

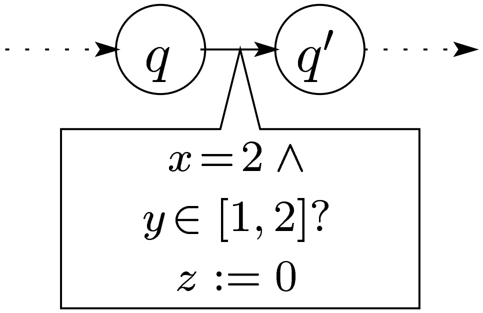
$\overline{z}' = \textit{time}' \wedge$

$\textit{time}' - \overline{x} = 2 \wedge$

$\textit{time}' - \overline{y} \in [1,2]$

The reachability relation

$\overline{x}$ = variable representing the latest reset time for x



$$\mathit{time}, \overline{x}, \overline{y}, \overline{z} \leq \mathit{time}' \wedge$$
$$\overline{x}' = \overline{x} \wedge$$
$$\overline{y}' = \overline{y} \wedge$$
$$\overline{z}' = \mathit{time}' \wedge$$
$$\mathit{time}' - \overline{x} = 2 \wedge$$
$$\mathit{time}' - \overline{y} \in [1,2]$$

By composition, difference constraints yield only difference constraints!

Modular construction of the reachability relation

- Unions = $\vee$.

Modular construction of the reachability relation

- Unions = $\vee$.
- Relational composition:

Modular construction of the reachability relation

- Unions = $\vee$.
- Relational composition:

$$R_1 \mapsto f_1(x, y, z, x', y', z') \ , \qquad R_2 \mapsto f_2(x, y, z, x', y', z')$$

$$R_1 \circ R_2 \mapsto \exists x', y', z'.\ f_1(x, y, z, x', y', z') \wedge f_2(x', y', z', x'', y'', z'')$$

- Reflexive-transitive closure:
  Second-order quantification – verification is a problem.

Modular construction of the reachability relation

- Unions = $\vee$.
- Relational composition:

$$R_1 \mapsto f_1(x, y, z, x', y', z')\ , \qquad R_2 \mapsto f_2(x, y, z, x', y', z')$$

$$R_1 \circ R_2 \mapsto \exists x', y', z'.\ f_1(x, y, z, x', y', z') \wedge f_2(x', y', z', x'', y'', z'')$$

- Reflexive-transitive closure:
Second-order quantification – verification is a problem.

Required:

- Discretizations of clock relations;
- Automata accepting such discretizations.

Summary of the presentation

- Timed automata and clock relations
- Representing clock relations with n-automata
- Discretizations of timed languages
- Stack automata for discretizations of timed languages

Representing clock relations

(Discrete time case – resets occur at integer time points)

$$R : x' - x \in [2,3] \wedge x' - y \in]-1,2[\wedge y' - x' \in [0,\infty[$$

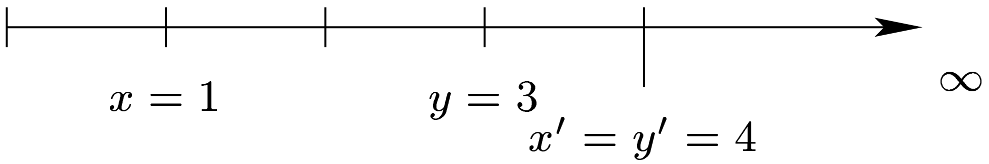
$$R \ni (x = 1, y = 3, x' = 4, y' = 4)$$

Representing clock relations

(Discrete time case – resets occur at integer time points)

$$R : x' - x \in [2,3] \wedge x' - y \in]-1,2[\wedge y' - x' \in [0,\infty[$$

$$R \ni (x = 1, y = 3, x' = 4, y' = 4)$$

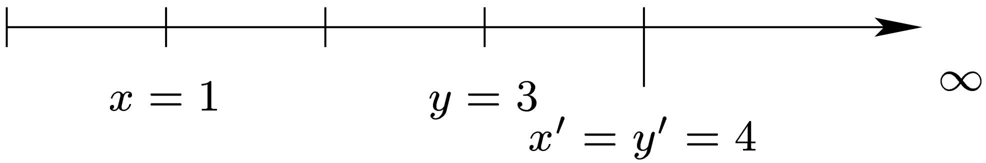


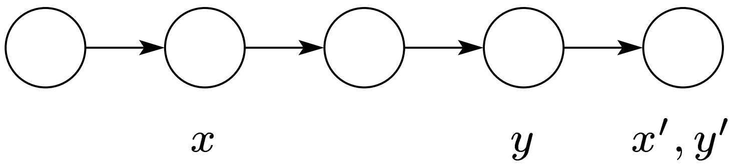
Representing clock relations

(Discrete time case – resets occur at integer time points)

$$R : x' - x \in [2,3] \wedge x' - y \in \,]-1,2[\, \wedge y' - x' \in [0,\infty[$$

$$R \ni (x = 1, y = 3, x' = 4, y' = 4)$$

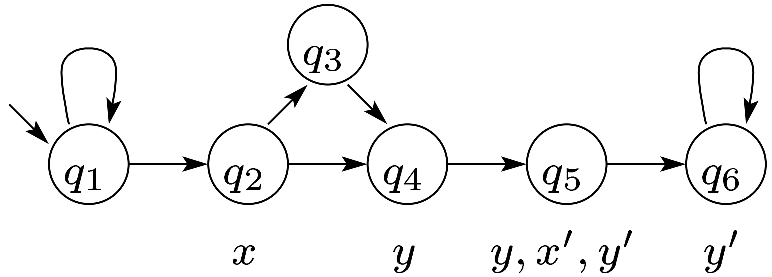




$2n$-automata

$$R : x'-x \in [2,3] \wedge x'-y \in \,]-1,2[\, \wedge y'-x' \in [0,\infty[$$

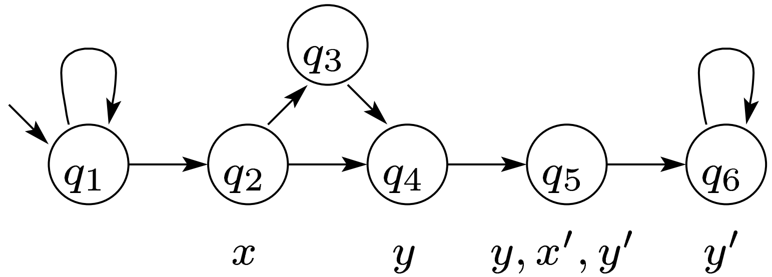
4-automaton for all the tuples in R:



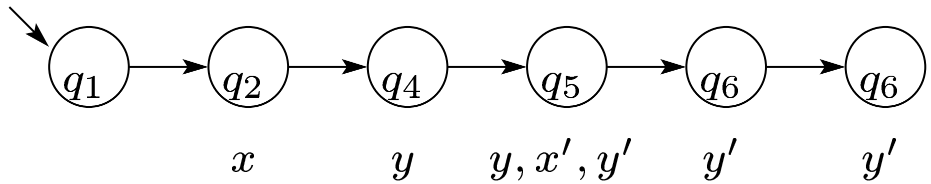
$2n$-automata

$$R : x' - x \in [2,3] \wedge x' - y \in \,]-1,2[\, \wedge y' - x' \in [0,\infty[$$

4-automaton for all the tuples in R:



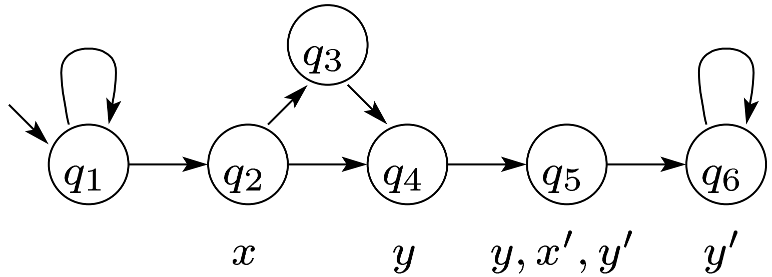
Accepting run:



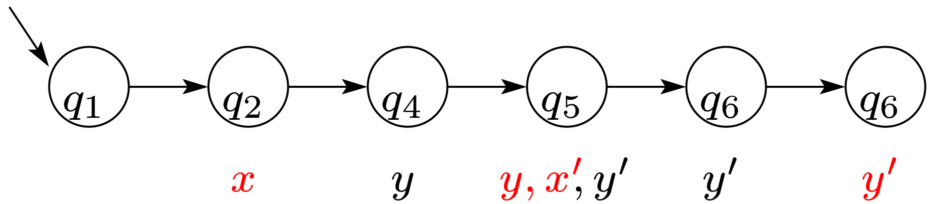
$2n$-automata

$$R : x' - x \in [2,3] \wedge x' - y \in]-1,2[\wedge y' - x' \in [0,\infty[$$

4-automaton for all the tuples in R:



Accepting run:



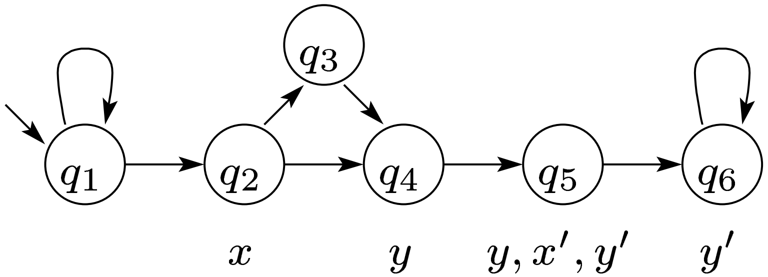
Tuples associated to this accepting run:

$$(1, 3, 3, 5),$$

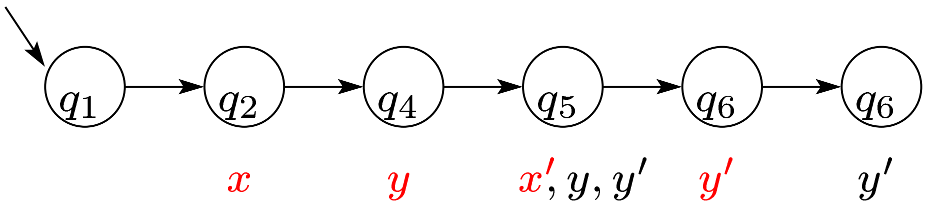
$2n$-automata

$$R : x' - x \in [2,3] \wedge x' - y \in]-1,2[\wedge y' - x' \in [0,\infty[$$

4-automaton for all the tuples in R:



Accepting run:



Tuples associated to this accepting run:

$$(1,3,3,5), (1,2,3,4) \ldots$$

Properties of $2n$-automata

- Emptiness is decidable (NP-complete).

Properties of $2n$-automata

- Emptiness is decidable (NP-complete).
  - Algorithm with time exponential in n but linear in the number of states.

n-automata and relational composition

Relational composition:

$$\{(x = 1, y = 2, x' = 4, y' = 3)\} \circ$$
$$\{x = 4, y = 3, x' = 6, y' = 4)\} =$$
$$\{(x = 1, y = 2, x' = 6, y' = 4)\}$$

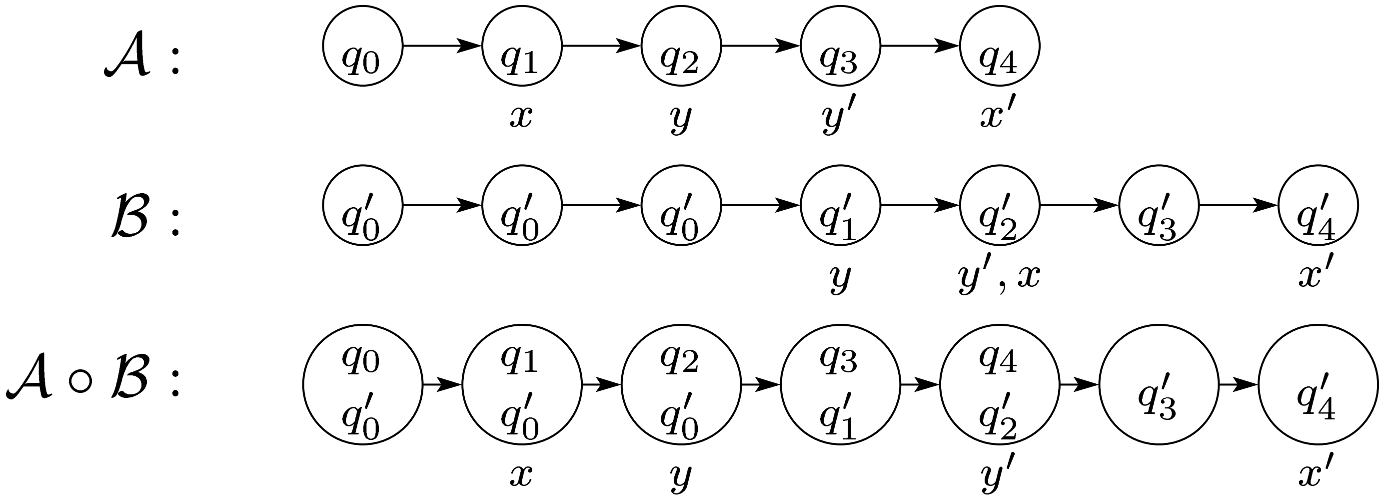
n-automata and relational composition

Relational composition:

$$\{(x = 1, y = 2, x' = 4, y' = 3)\}\circ$$

$$\{x = 4, y = 3, x' = 6, y' = 4)\} =$$

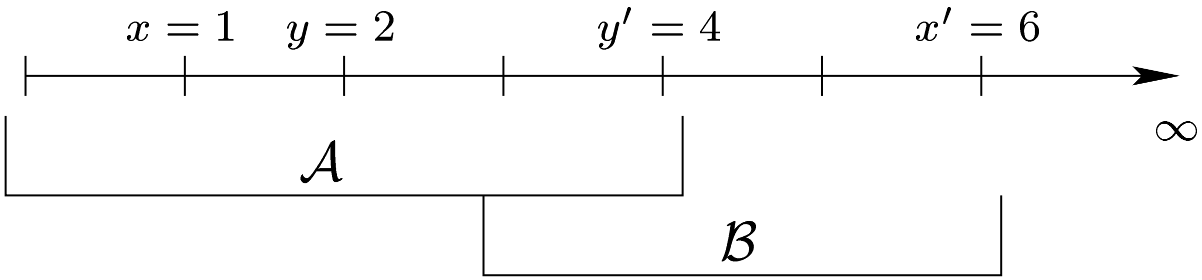
$$\{(x = 1, y = 2, x' = 6, y' = 4)\}$$



n-automata and relational composition

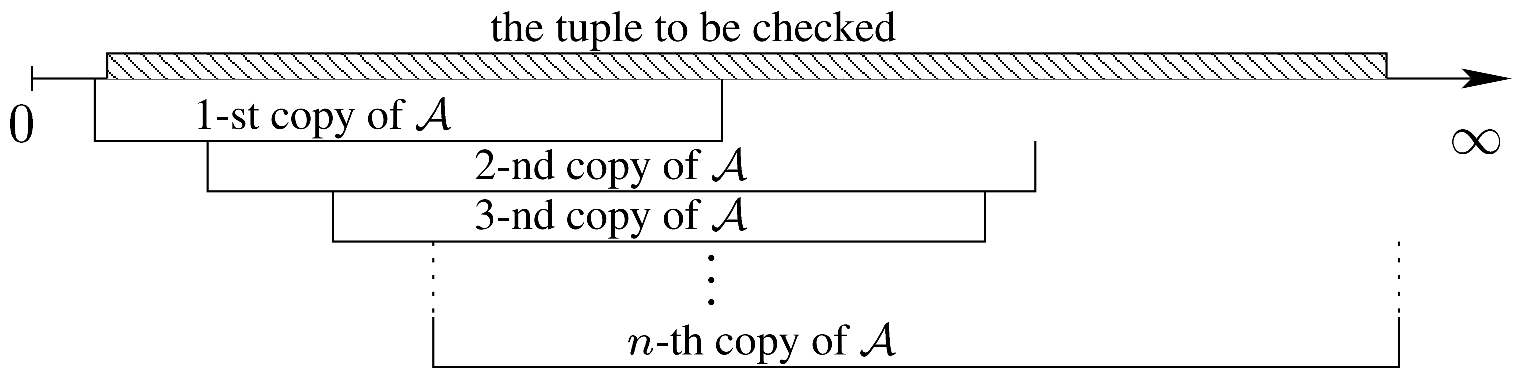
Relational composition:

$$\{(x = 1, y = 2, x' = 4, y' = 3)\} \circ$$
$$\{x = 4, y = 3, x' = 6, y' = 4)\} =$$
$$\{(x = 1, y = 2, x' = 6, y' = 4)\}$$



The star closure problem

In general, $2n$-automata are not closed under star.



“Non-elastic” relations

Relation $R \subseteq \mathbb{R}^{2n}$ which contains only tuples $(x_1, \ldots, x_n, x'_1, \ldots, x'_n)$ for which:

$$\text{if } x'_i \neq x_i \text{ and } x'_j \neq x_j \text{ then } x'_i - x_j \geq 0$$

If, along a chain of transitions, two clocks x_i and x_j are both reset, then their new reset points are both greater than their former reset points.

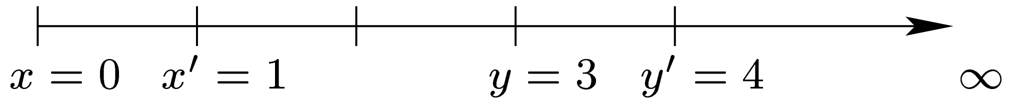
“Non-elastic” relations

Relation $R \subseteq \mathbb{R}^{2n}$ which contains only tuples $(x_1, \ldots, x_n, x'_1, \ldots, x'_n)$ for which:

$$\text{if } x'_i \neq x_i \text{ and } x'_j \neq x_j \text{ then } x'_i - x_j \geq 0$$

If, along a chain of transitions, two clocks x_i and x_j are both reset, then their new reset points are both greater than their former reset points.

“Elastic” tuple:



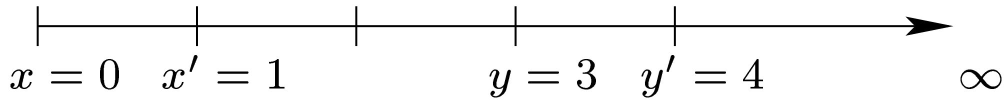
“Non-elastic” relations

Relation $R \subseteq \mathbb{R}^{2n}$ which contains only tuples $(x_1, \ldots, x_n, x'_1, \ldots, x'_n)$ for which:

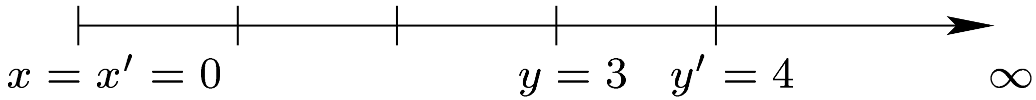
$$\text{if } x'_i \neq x_i \text{ and } x'_j \neq x_j \text{ then } x'_i - x_j \geq 0$$

If, along a chain of transitions, two clocks x_i and x_j are both reset, then their new reset points are both greater than their former reset points.

“Elastic” tuple:



“Non-elastic” tuple:



“Non-elastic” relations (2)

Theorem. Given a $2n$-automaton $\mathcal{A}$, if $L(\mathcal{A})^k$ contains only of non-elastic tuples for all $k \in \mathbb{N}$ then $L(\mathcal{A})^*$ is recognizable by a $2n$-automaton.

Complexity = $O(2^{card(Q_\mathcal{A}) \cdot 4^n})$

“Non-elastic” relations (2)

Theorem. Given a $2n$-automaton $\mathcal{A}$, if $L(\mathcal{A})^k$ contains only of non-elastic tuples for all $k \in \mathbb{N}$ then $L(\mathcal{A})^*$ is recognizable by a $2n$-automaton.

Complexity = $O(2^{card(Q_{\mathcal{A}})\cdot 4^n})$

Theorem. The union of all reachability relations for all timed automata is a non-elastic relation.

Representing clock relations – dense time case

Each boolean combination over “difference constraints” is a disjunction of “basic blocks”:

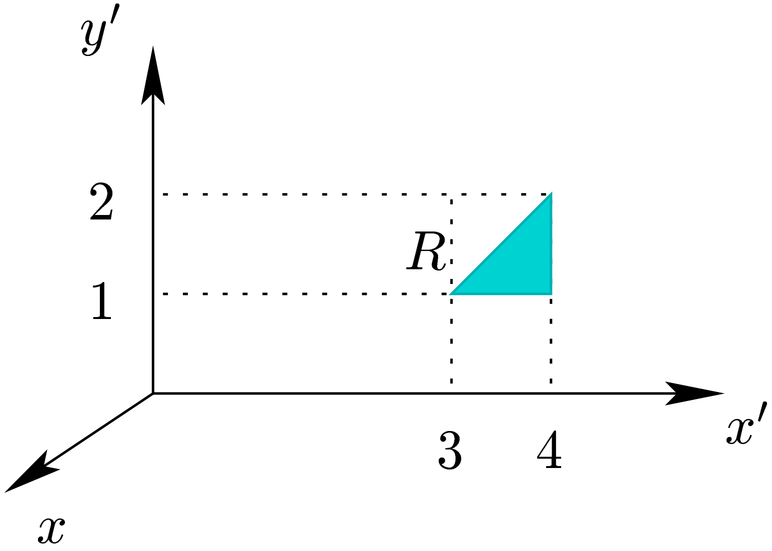
Region: $\bigwedge_{1 \le i < j \le n} x_i - x_j \in I_{ij}$ → point interval / open unit interval

How to use $2n$-automata for regions?

Regions

$$R : x = y \wedge x' - y' \ \in]2,3[\wedge y' - x \ \in]1,2[\wedge x' - x \ \in]3,4[$$

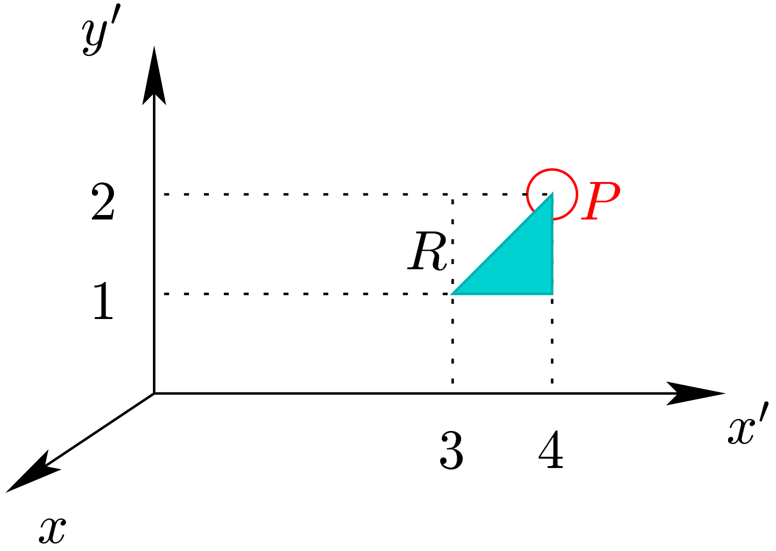
If we set $y = 0$ then we can represent the respective “projection” as:



Regions

$$R : x = y \wedge x' - y' \;\in\,]2,3[\wedge y' - x \;\in\,]1,2[\wedge x' - x \;\in\,]3,4[$$

If we set $y = 0$ then we can represent the respective “projection” as:



Idea: instead of working with R, work with one of its three vertices.

Vertex P: tuple of integers $(x = 0, y = 0, x' = 4, y' = 2)$.

But P is also a vertex in other 15 regions!

Representing regions

Relative positioning of P w.r.t. R:

$$P : (0, 0, 4, 2) \mapsto \begin{array}{cc} & \begin{array}{cccc} x & y & x' & y' \end{array} \\ \begin{array}{c} x \\ y \\ x' \\ y' \end{array} & \begin{pmatrix} 0 & 0 & -4 & -2 \\ 0 & 0 & -4 & -2 \\ 4 & 4 & 0 & 2 \\ 2 & 2 & -2 & 0 \end{pmatrix} \end{array}$$

Representing regions

Relative positioning of P w.r.t. R:

$$P : (0,0,4,2) \mapsto \begin{array}{cc} & \begin{array}{cccc} x & y & x' & y' \end{array} \\ \begin{array}{c} x \\ y \\ x' \\ y' \end{array} & \begin{pmatrix} 0 & 0 & -4 & -2 \\ 0 & 0 & -4 & -2 \\ 4 & 4 & 0 & 2 \\ 2 & 2 & -2 & 0 \end{pmatrix} \end{array}$$

$$\begin{aligned} R : \quad & x = y \wedge x'-y' \in \,]2,3[\, \wedge \\ & y'-x \in \,]1,2[\, \wedge x'-x \in \,]3,4[\end{aligned} \qquad \begin{pmatrix} \{0\} & \{0\} &]-4,-3[&]-2,-1[\\ \{0\} & \{0\} &]-4,-3[&]-2,-1[\\ ]3,4[&]3,4[& \{0\} &]2,3[\\ ]1,2[&]1,2[&]-3,-2[& \{0\} \end{pmatrix}$$

Representing regions

$$P : \quad \begin{pmatrix} 0 & 0 & -4 & -2 \\ 0 & 0 & -4 & -2 \\ 4 & 4 & 0 & 2 \\ 2 & 2 & -2 & 0 \end{pmatrix} \quad M = \begin{pmatrix} `=` & `=` & `>` & `>` \\ `=` & `=` & `>` & `>` \\ `<` & `<` & `=` & `>` \\ `<` & `<` & `<` & `=` \end{pmatrix}$$

$$R : \begin{pmatrix} \{0\} & \{0\} &]-4,-3[&]-2,-1[\\ \{0\} & \{0\} &]-4,-3[&]-2,-1[\\ ]3,4[&]3,4[& \{0\} &]2,3[\\ ]1,2[&]1,2[&]-3,-2[& \{0\} \end{pmatrix}$$

Representing regions

$$P : \begin{pmatrix} 0 & 0 & -4 & -2 \\ 0 & 0 & -4 & -2 \\ 4 & 4 & 0 & 2 \\ 2 & 2 & -2 & 0 \end{pmatrix} \quad M = \begin{pmatrix} `=` & `=` & `>` & `>` \\ `=` & `=` & `>` & `>` \\ `<` & `<` & `=` & `>` \\ `<` & `<` & `<` & `=` \end{pmatrix}$$

$$R : \begin{pmatrix} \{0\} & \{0\} &]-4,-3[&]-2,-1[\\ \{0\} & \{0\} &]-4,-3[&]-2,-1[\\ ]3,4[&]3,4[& \{0\} &]2,3[\\ ]1,2[&]1,2[&]-3,-2[& \{0\} \end{pmatrix}$$

Endow $2n$-automata with such matrix labels.

$2n$-automata and timed automata

The reachability relation R_A defined by a timed automaton A can be computed using $2n$-automata.

The result is a $2n$-automaton whose accepting runs represent the regions in R_A.

Summary of the presentation

- Timed automata and clock relations
- Representing clock relations with n-automata
- Discretizations of timed languages
- Stack automata for discretizations of timed languages

Timed words

- Sequences of actions and durations:

$$2\,a\,1.5\,1.5\,b\,1\,a$$

Timed words

- Sequences of actions and durations:

$$2\,a\,1.5\,1.5\,b\,1\,a = 2\,a\,3\,b\,1\,a$$

- The monoid of timed words : $TW(\Sigma) = \Sigma^* \oplus \mathbb{R}_{\geq 0}$.

Timed words

- Sequences of actions and durations:

$$2\,a\,1.5\,1.5\,b\,1\,a = 2\,a\,3\,b\,1\,a$$

- The monoid of timed words : $TW(\Sigma) = \Sigma^* \oplus \mathbb{R}_{\geq 0}$.
- Direct sum $\simeq$ shuffle:

$$2\,a\,3\,b\,1\,a \in aba \sqcup\!\sqcup 6$$

Timed words

- Sequences of actions and durations:

$$2\,a\,1.5\,1.5\,b\,1\,a = 2\,a\,3\,b\,1\,a$$

- The monoid of timed words : $TW(\Sigma) = \Sigma^* \oplus \mathbb{R}_{\geq 0}$.
- Direct sum $\simeq$ shuffle:

$$2\,a\,3\,b\,1\,a \in aba \sqcup\!\sqcup 6$$

- Actions serve for communication/synchronization.

Timed words

- Sequences of actions and durations:

$$2\,a\,1.5\,1.5\,b\,1\,a = 2\,a\,3\,b\,1\,a$$

- The monoid of timed words : $TW(\Sigma) = \Sigma^* \oplus \mathbb{R}_{\geq 0}$.
- Direct sum $\simeq$ shuffle:

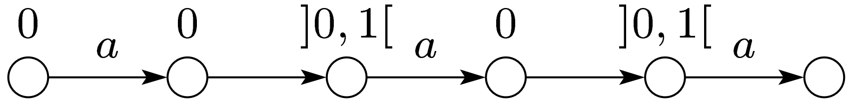
$$2\,a\,3\,b\,1\,a \in aba \sqcup\!\sqcup 6$$

- Actions serve for communication/synchronization.
- Synchronization $=$ intersection of timed languages.

The “region discretization”

$$L_1 = \{at_1at_2a \mid t_1, t_2 \ \in\ ]0,1[\} = \left\| \langle a\underline{\mathbf{t}}\rangle_{]0,1[} \langle a\underline{\mathbf{t}}a\rangle_{]0,1[} \right\|$$

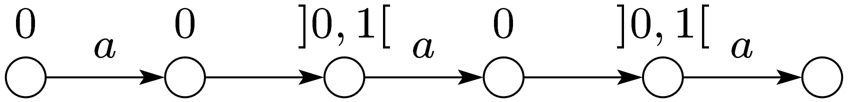
$\mathcal{A}_1$:



The “region discretization”

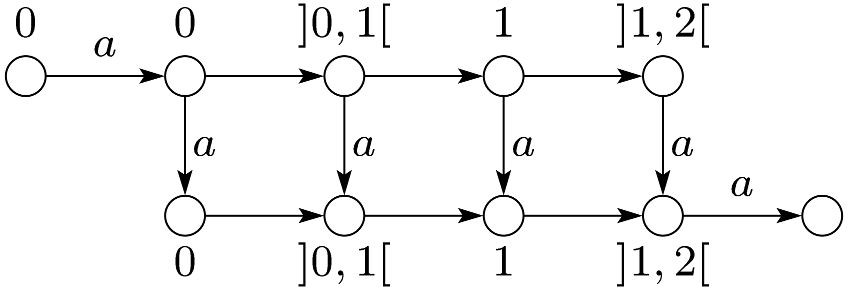
$$L_1 = \{at_1at_2a \mid t_1, t_2 \in]0,1[\} = \left\| \langle a\underline{\mathbf{t}}\rangle_{]0,1[} \langle a\underline{\mathbf{t}}a\rangle_{]0,1[} \right\|$$

$\mathcal{A}_1$:



$$L_2 = \{at_1at_2a \mid t_1 + t_2 \in]1,2[\} = \left\| \langle a\underline{\mathbf{t}}a\underline{\mathbf{t}}a\rangle_{]1,2[} \right\|$$

$\mathcal{A}_2$:



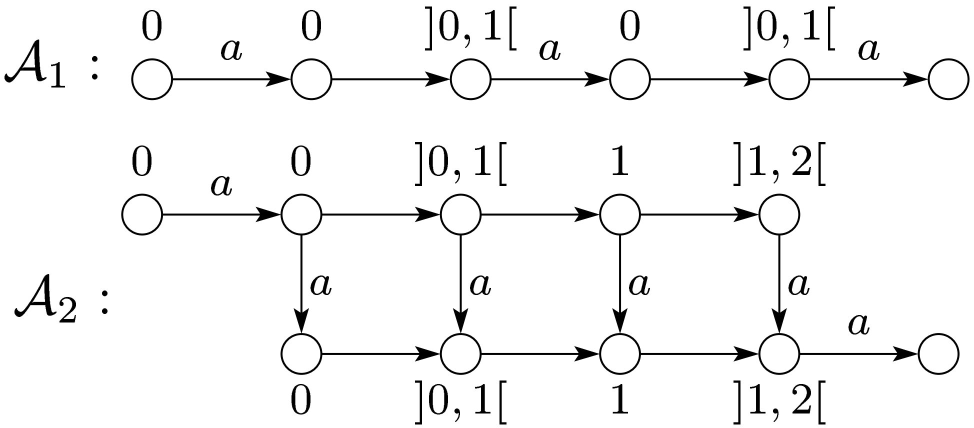
The region discretization and intersection

$$L_1 \cap L_2 = \left\| \left[\langle a\underline{\mathbf{t}} \rangle_{]0,1[} \langle a\underline{\mathbf{t}}a \rangle_{]0,1[} \; \right]_{]1,2[} \right\|$$

$$\|\mathcal{A}_1 \cap \mathcal{A}_2\| = L_1 \cap L_2?$$

The region discretization and intersection

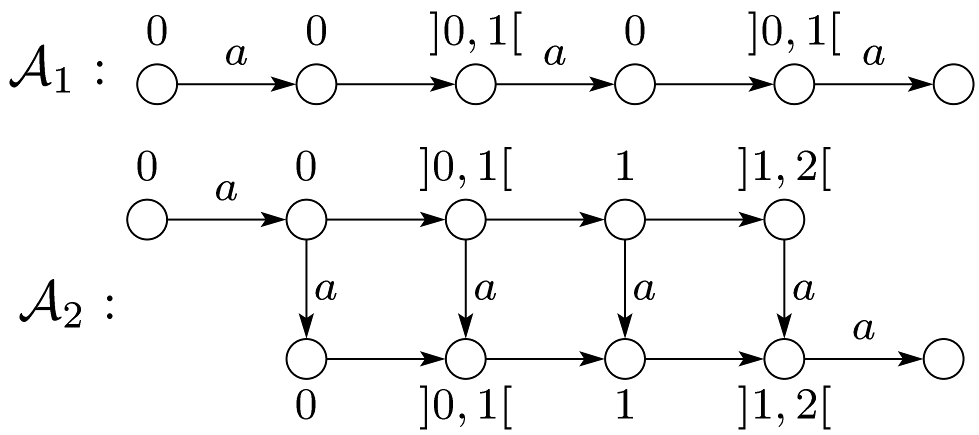
$$L_1 \cap L_2 = \left\| \left[\langle a\underline{\mathbf{t}}\rangle_{]0,1[} \langle a\underline{\mathbf{t}}a\rangle_{]0,1[} \right]_{]1,2[} \right\|$$



$$\|\mathcal{A}_1 \cap \mathcal{A}_2\| = L_1 \cap L_2?$$

The region discretization and intersection

$$L_1 \cap L_2 = \left\| \left[\langle a\underline{\mathbf{t}} \rangle_{]0,1[} \langle a\underline{\mathbf{t}}a \rangle_{]0,1[} \right]_{]1,2[} \right\|$$



$$\|\mathcal{A}_1 \cap \mathcal{A}_2\| = L_1 \cap L_2?$$

Obviously not !

A discretization with nonarchimedian delays

- δ^+ – a positive delay, smaller than 1.

A discretization with nonarchimedian delays

- δ^+ – a positive delay, smaller than 1.
- δ^- – a negative delay, grater than -1.

A discretization with nonarchimedian delays

- δ^+ – a positive delay, smaller than 1.
- δ^- – a negative delay, grater than -1.
- $\|\delta^+ \cdot \delta^-\| = 0$.

A discretization with nonarchimedian delays

- δ^+ – a positive delay, smaller than 1.
- δ^- – a negative delay, grater than -1.
- $\|\delta^+ \cdot \delta^-\| = 0$.
- both nonarchimedian: $\forall n \in \mathbb{N}, \left\|(\delta^+)^n\right\| < 1$ and $\left\|(\delta^-)^n\right\| > -1$.

A discretization with nonarchimedian delays

- δ^+ – a positive delay, smaller than 1.
- δ^- – a negative delay, grater than -1.
- $\|\delta^+ \cdot \delta^-\| = 0$.
- both nonarchimedian: $\forall n \in \mathbb{N}, \left\|(\delta^+)^n\right\| < 1$ and $\left\|(\delta^-)^n\right\| > -1$.

$$L_1 \cap L_2 = \left\| \left[\langle a\underline{\mathbf{t}}\rangle_{]0,1[} \langle a\underline{\mathbf{t}}a\rangle_{]0,1[} \right]_{]1,2[} \right\|$$

A discretization with nonarchimedian delays

- δ^+ – a positive delay, smaller than 1.
- δ^- – a negative delay, grater than -1.
- $\|\delta^+ \cdot \delta^-\| = 0$.
- both nonarchimedian: $\forall n \in \mathbb{N}, \left\|(\delta^+)^n\right\| < 1$ and $\left\|(\delta^-)^n\right\| > -1$.

$$L_1 \cap L_2 = \left\| \left[\langle a\underline{\mathbf{t}} \rangle_{]0,1[} \langle a\underline{\mathbf{t}}a \rangle_{]0,1[} \right]_{]1,2[} \right\| = \left\| a\delta^+\delta^+ a 1 \delta^- a \right\|$$

A discretization with nonarchimedian delays

- δ^+ – a positive delay, smaller than 1.
- δ^- – a negative delay, grater than -1.
- $\|\delta^+ \cdot \delta^-\| = 0$.
- both nonarchimedian: $\forall n \in \mathbb{N}, \left\|(\delta^+)^n\right\| < 1$ and $\left\|(\delta^-)^n\right\| > -1$.

$$L_1 \cap L_2 = \left\| \left[\langle a\underline{\mathbf{t}} \rangle_{]0,1[} \langle a\underline{\mathbf{t}}a \rangle_{]0,1[} \right]_{]1,2[} \right\| = \left\| a\delta^+\delta^+ a1\delta^- a \right\|$$

$$\left\| aa1a \right\| = \left\{ at_1at_2a \mid t_1 = 0, t_2 = 1, t_1 + t_2 = 1 \right\}$$

A discretization with nonarchimedian delays

- δ^+ – a positive delay, smaller than 1.
- δ^- – a negative delay, grater than -1.
- $\|\delta^+ \cdot \delta^-\| = 0$.
- both nonarchimedian: $\forall n \in \mathbb{N}, \left\|(\delta^+)^n\right\| < 1$ and $\left\|(\delta^-)^n\right\| > -1$.

$$L_1 \cap L_2 = \left\| \left[\langle a\underline{\mathbf{t}} \rangle_{]0,1[} \langle a\underline{\mathbf{t}}a \rangle_{]0,1[} \right]_{]1,2[} \right\| = \left\| a\delta^+\delta^+a1\delta^-a \right\|$$

$$\left\| aa1a \right\| = \left\{ at_1at_2a \mid t_1 = 0, t_2 = 1, t_1 + t_2 = 1 \right\}$$

$$\left\| \overbrace{\underbrace{a\delta^+\delta^+}_{\#_{\delta^+} > \#_{\delta^-}} \; \underbrace{a1\delta^-a}_{\#_{\delta^+} < \#_{\delta^-}}}^{\#_{\delta^+} > \#_{\delta^-}} \right\| = \left\{ at_1at_2a \mid t_1 \in]0,1[, t_2 \in]0,1[, t_1 + t_2 \in]1,2[\right\}$$

Arguments in favor of nonarchimedian delays

- Each symbol in Σ acts like a “time hook”.

Arguments in favor of nonarchimedian delays

- Each symbol in Σ acts like a “time hook”.
- Words over $\Sigma^* \sqcup\!\sqcup 1^*$ are timed words over a discrete time domain.

Arguments in favor of nonarchimedian delays

- Each symbol in Σ acts like a "time hook".
- Words over $\Sigma^* \sqcup\!\sqcup 1^*$ are timed words over a discrete time domain.
- All is needed then is, for each word, a "pattern" for the fractional parts between all pairs of time hooks.

Arguments in favor of nonarchimedian delays

- Each symbol in Σ acts like a “time hook”.
- Words over $\Sigma^* \sqcup\!\sqcup\, 1^*$ are timed words over a discrete time domain.
- All is needed then is, for each word, a “pattern” for the fractional parts between all pairs of time hooks.
- Practically, what we need is the shuffle $\Sigma^* \sqcup\!\sqcup\, 1^* \sqcup\!\sqcup\, \mathcal{G}_3$, where $\mathcal{G}_3$ is the three-element group.

Properties of nonarchimedian delay discretizations

- Nonarchimedian delay discretizations induce a partition on TW(Σ): for all $w_1, w_2 \in \Sigma^* \sqcup\!\sqcup 1^* \sqcup\!\sqcup (\delta^+ + \delta^-)^*$,

  if $\|w_1\| \cap \|w_2\| \neq \emptyset$ then $\|w_1\| = \|w_2\|$.

Properties of nonarchimedian delay discretizations

- Nonarchimedian delay discretizations induce a partition on TW(Σ): for all $w_1, w_2 \in \Sigma^* ⧢ 1^* ⧢ (\delta^+ + \delta^-)^*$,

$$\text{if } \|w_1\| \cap \|w_2\| \neq \emptyset \text{ then } \|w_1\| = \|w_2\|.$$

- Each class can be represented by several (nonarchimedian) delay discretizations:

$$\|a\delta^+\delta^+a1\delta^-a\| = \|a1\delta^-a\delta^+\delta^+a\| = \|a1\delta^-a1\delta^-a\|$$

Nonarchimedian delay discretizations for timed languages

- **Aim:** obtain nonarchimedian delay discretizations for all timed languages by $+$, $\cap$, $\cdot$, $*$ and action renaming.

Nonarchimedian delay discretizations for timed languages

- **Aim:** obtain nonarchimedian delay discretizations for all timed languages by $+$, $\cap$, $\cdot$, $*$ and action renaming.
- **Basic blocks:** shuffles of a regular language $L \subseteq \Sigma^*$, a regular language *Ticks* $\subseteq 1^*$ and a (?) language $D \subseteq \left(\delta^+ + \delta^-\right)^*$.

Nonarchimedian delay discretizations for timed languages

- **Aim:** obtain nonarchimedian delay discretizations for all timed languages by $+$, $\cap$, $\cdot$, $*$ and action renaming.
- **Basic blocks:** shuffles of a regular language $L \subseteq \Sigma^*$, a regular language *Ticks* $\subseteq 1^*$ and a (?) language $D \subseteq \left(\delta^+ + \delta^-\right)^*$.
- The number of positive/negative delays in between two “hook” points may be in general unbounded:

We cannot bind the number of δ^+ in the discretization of

$$\left[a\left(\langle b\underline{\mathbf{t}}\rangle_{]0,1[}\right)^* a\left(\langle b\underline{\mathbf{t}}\rangle_{]0,1[}\right)^*\right]_{]1,2[}$$

Languages of nonarhimedian delay discretizations

$$L_1 \cap L_2 = \Big\|\Big\{a1w_1a1w_2a \mid w_1, w_2 \in (\delta^+ + \delta^-)^*, \#_{\delta^+}(w_1) < \#_{\delta^-}(w_1)$$
$$\#_{\delta^+}(w_2) < \#_{\delta^-}(w_2),\ \#_{\delta^+}(w_1w_2) < \#_{\delta^-}(w_1w_2)\Big\}\Big\|$$

A non-regular language ! (in this case – context free)

Languages of nonarhimedian delay discretizations

$$L_1 \cap L_2 = \left\| \{ a1w_1a1w_2a \mid w_1, w_2 \in (\delta^+ + \delta^-)^*, \#_{\delta^+}(w_1) < \#_{\delta^-}(w_1) \right.$$
$$\left. \#_{\delta^+}(w_2) < \#_{\delta^-}(w_2),\ \#_{\delta^+}(w_1w_2) < \#_{\delta^-}(w_1w_2) \} \right\|$$

A non-regular language ! (in this case – context free)

$$L = \{ aw_1aw_2aw_3a \mid w_1, w_2, w_3 \in (\delta^+ + \delta^-)^*,$$
$$\#_{\delta^+}(w_1w_2) < \#_{\delta^-}(w_1w_2), \#_{\delta^+}(w_2w_3) > \#_{\delta^-}(w_2w_3) \}$$
$$L \cap\ a(\delta^+)^*a(\delta^+)^*a(\delta^+)^*a =$$
$$= \left\| \{ a(\delta^+)^m a(\delta^+)^n a(\delta^+)^p a \mid m < n, n > p \} \right.$$

Context sensitive language !

Summary of the presentation

- Timed automata and clock relations
- Representing clock relations with n-automata
- Discretizations of timed languages
- Stack automata for discretizations of timed languages

Stack automata

A (one-way) **stack automaton** = Turing machine with two tapes – the input tape and the working tape – and n sets of distinguished states Q_i $(1 \leq i \leq n)$.

- The input tape is parsed in one pass.
- The working tape is unbounded in both directions.
- A given word is accepted iff, when its parse is finished, the machine arrives in a final state.

(Not exactly like Ginsburg's stack automata...)

Example

$$L = \left\{aw_1aw_2aw_3a \mid w_1, w_2, w_3 \in (1 + \delta^+ + \delta^-)^*, \right.$$
$$\left. \#_{\delta^+}(w_1w_2) < \#_{\delta^-}(w_1w_2), \#_{\delta^+}(w_2w_3) > \#_{\delta^-}(w_2w_3)\right\}$$

- Symbols on the working tape: $\perp, \delta^+, \delta^-$ plus four markers, 1, 2, 3 and 4.
- When meeting the i-th a we put an i on the working tape ($i = 1..4$).
- 3 must be to the right of 1 and 4 to the left of 2.

Example

$$L = \{aw_1aw_2aw_3a \mid w_1, w_2, w_3 \in (1 + \delta^+ + \delta^-)^*,$$
$$\#_{\delta^+}(w_1w_2) < \#_{\delta^-}(w_1w_2), \#_{\delta^+}(w_2w_3) > \#_{\delta^-}(w_2w_3)\}$$

- Symbols on the working tape: $\perp, \delta^+, \delta^-$ plus four markers, 1, 2, 3 and 4.
- When meeting the i-th a we put an i on the working tape ($i = 1..4$).
- 3 must be to the right of 1 and 4 to the left of 2.

Conjecture: The class of stack automata that are used in the nonarhimedian discretization of timed automata have a decidable emptiness problem.

Intersection

Theorem. The stack automata that represent timed automata are closed under intersection.

Intersection

Theorem. The stack automata that represent timed automata are closed under intersection.

Lemma. Given two stack automata, suppose that in each jointly reachable configuration both automata move their head in the same direction and/or write the same number of symbols and/or erase the same number of symbols.

Then the intersection of their languages can be accepted by a stack automaton too.

Conclusions

New discretization techniques for timed systems.

- Compositionality – concatenation, intersection.
- Classes of automata.
- Connections to trace theory ?