Automata for Real-time Systems

B. Srivathsan

Chennai Mathematical Institute

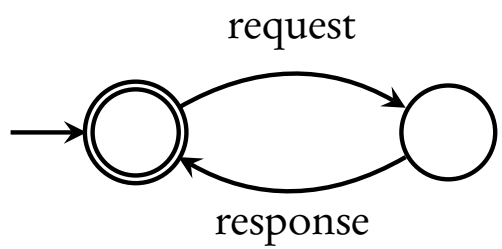
http://www.cmi.ac.in/~sri/Courses/TIME2013

Why do this course?

Automata (*Finite State Machines*) are **good abstractions** of many real systems

hardware circuits, communication protocols, biological processes, . . .

Automata can model many **properties** of systems



every request is followed by a response


System
Property
Automaton $\mathcal{A}$
Automaton $\mathcal{B}$

System

↓

Automaton $\mathcal{A}$

Property

↓

Automaton $\mathcal{B}$

Does system **satisfy** property?

System

↓

Automaton $\mathcal{A}$

Property

↓

Automaton $\mathcal{B}$

$$\mathcal{L}(\mathcal{A}) \subseteq \mathcal{L}(\mathcal{B})?$$

Does system **satisfy** property?

Model-checking

System → Automaton $\mathcal{A}$

Property → Automaton $\mathcal{B}$

$$\mathcal{L}(\mathcal{A}) \subseteq \mathcal{L}(\mathcal{B})?$$

Does system **satisfy** property?

In practice...

Huge system

Property

In practice...

Huge system

↓

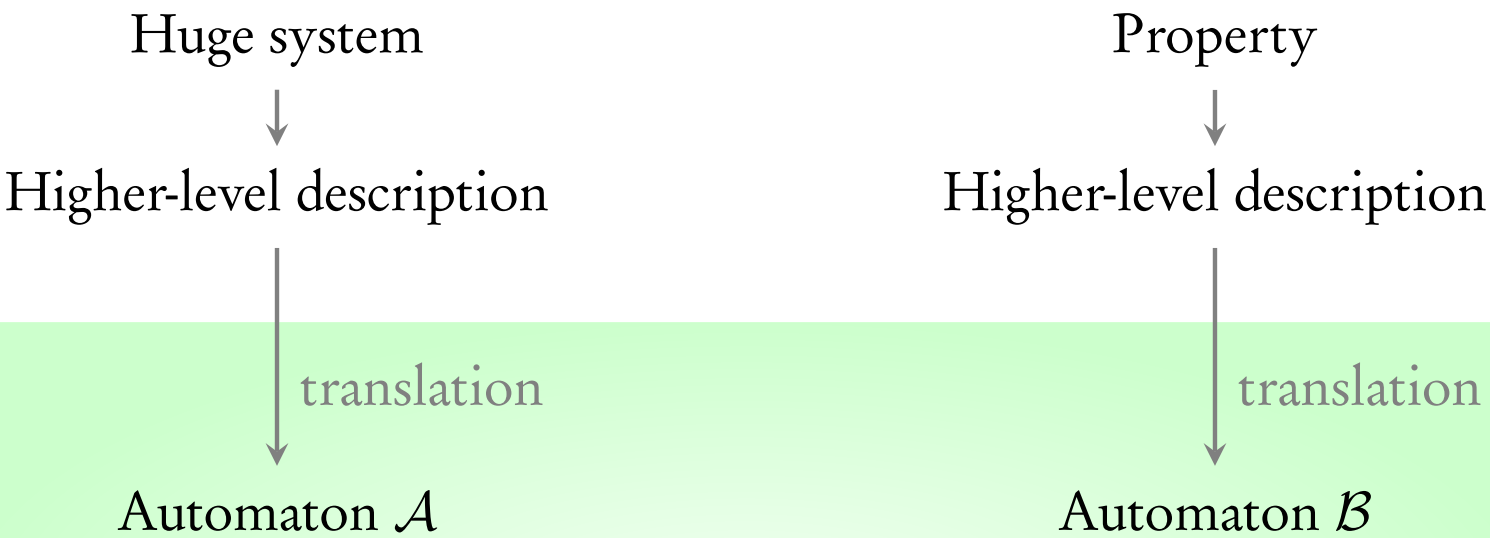
Higher-level description

Property

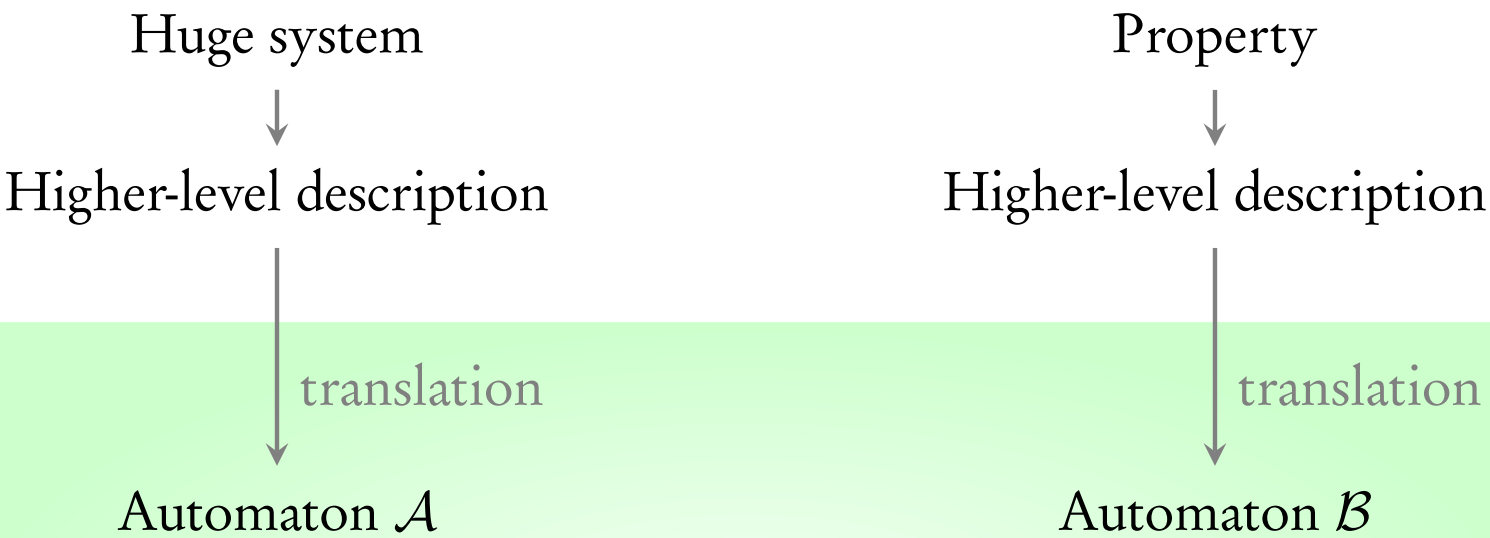
↓

Higher-level description

In practice...

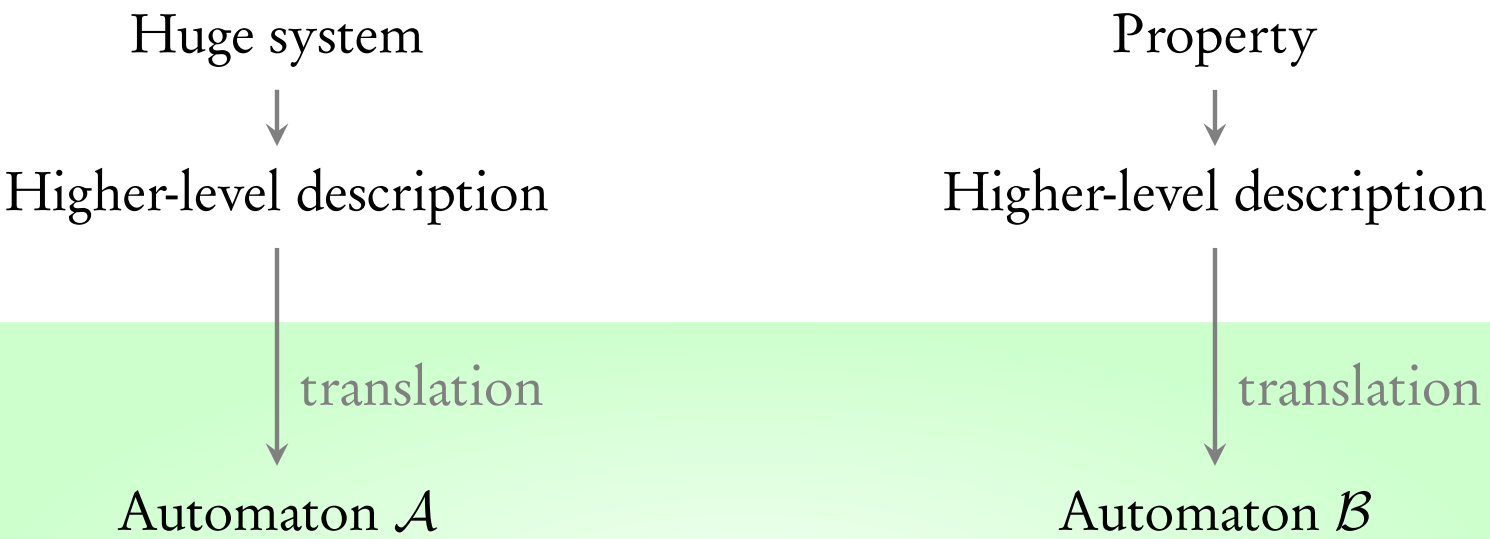

Huge system
Property
Higher-level description
Higher-level description
translation
translation
Automaton $\mathcal{A}$
Automaton $\mathcal{B}$
Model-Checker
$\mathcal{L}(\mathcal{A}) \subseteq \mathcal{L}(\mathcal{B})$?

In practice...



Some model-checkers: SMV, NuSMV, SPIN, ...

In practice...



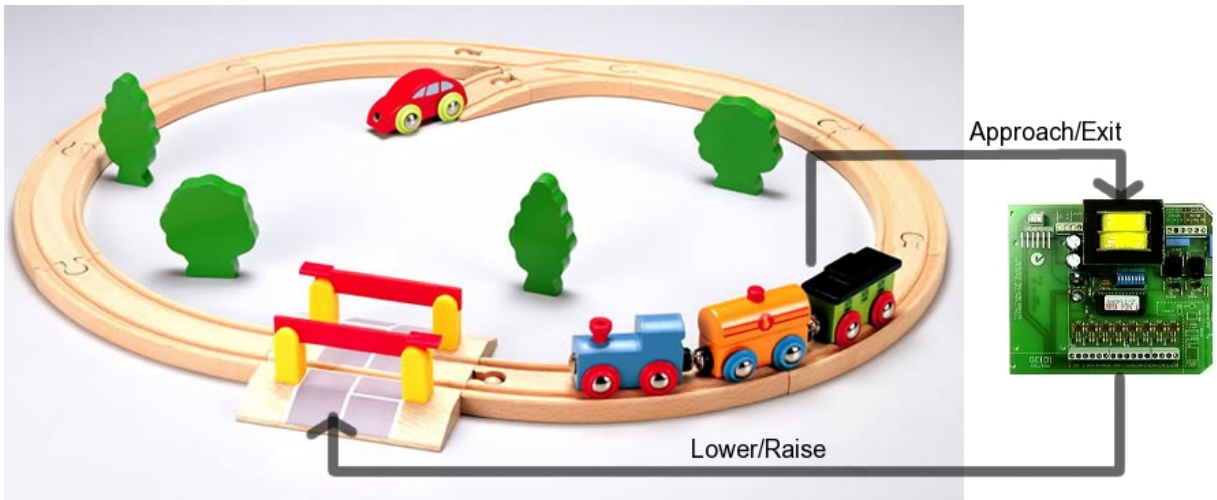
Some model-checkers: SMV, NuSMV, SPIN, ...

Turing Awards: Clarke, Emerson, Sifakis and Pnueli

Automata are **good abstractions** of many real systems

Automata are **good abstractions** of many real systems

Our course: Automata for **real-time** systems



Picture credits: F. Herbreteau

pacemaker, vehicle control systems, air traffic controllers, . . .

Timed Automata

R. Alur and D. Dill in early 90s

Timed Automata

R. Alur and D. Dill in early 90s

Some model-checkers: UPPAAL, KRONOS, RED, . . .

Goals of our course

- Understand **language theoretic** properties of timed automata
- Study **algorithms** used in model-checkers
- Examine selected **case-studies**

Model-checking caters to **both theory** enthusiasts and **practice** enthusiasts

Model-checking caters to **both theory** enthusiasts and **practice** enthusiasts

this course is a good starting point for model-checking real-time systems

Lecture 1:

Timed languages and timed automata

Σ : alphabet $\quad \{a, b\}$

Σ^* : words $\quad \{\varepsilon, a, b, aa, ab, ba, bb, aab, \dots\}$

$L \subseteq \Sigma^*$: language $\longrightarrow$ *property over words*

$L_1 :=$ {set of words starting with an " a "}

$\{a, aa, ab, aaa, aab, \dots\}$

$L_2 :=$ {set of words with a non-zero even length }

$\{aa, bb, ab, ba, abab, aaaa, \dots\}$

Σ : alphabet $\quad \{a, b\}$

Σ^* : words $\quad \{\varepsilon, a, b, aa, ab, ba, bb, aab, \dots\}$

$L \subseteq \Sigma^*$: language $\longrightarrow$ *property over words*

$L_1 :=$ {set of words starting with an " a "}

$\{a, aa, ab, aaa, aab, \dots\}$

$L_2 :=$ {set of words with a non-zero even length }

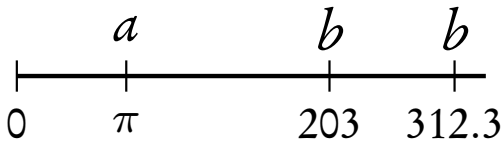
$\{aa, bb, ab, ba, abab, aaaa, \dots\}$

Finite automata, pushdown automata, Turing machines, ...

Σ : alphabet $\{a, b\}$

$T\Sigma^*$: timed words

$(aa;\ 0.8, 2.5)$

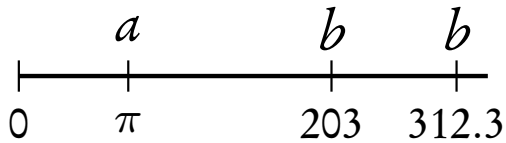


$(abb;\ \pi, 203, 312.3)$

Σ : alphabet $\{a, b\}$

$T\Sigma^*$: timed words

$(aa;\ 0.8, 2.5)$



$(abb;\ \pi, 203, 312.3)$

(w, τ)

Word

Time sequence

$w = a_1 \ldots a_n$

$a_i \in \Sigma$

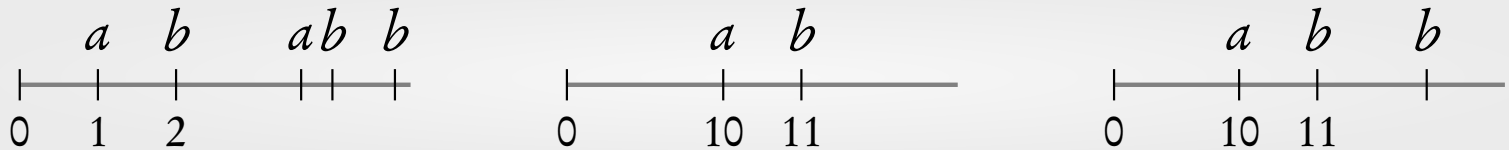
$\tau = \tau_1 \ldots \tau_n$

$\tau_i \in \mathbb{R}_{\geq 0}$

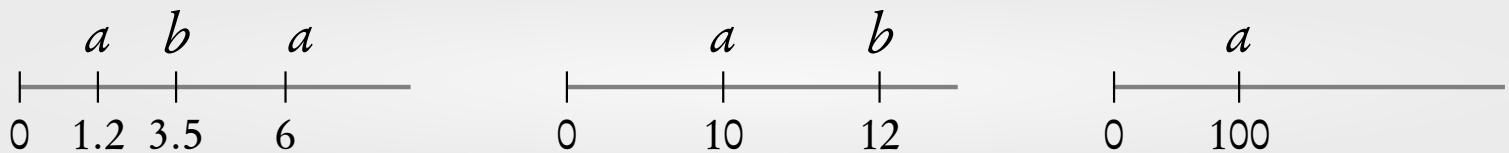
$\tau_1 \leq \cdots \leq \tau_n$

$L \subseteq T\Sigma^*$: Timed language $\longrightarrow$ *property over timed words*

$$L_1 := \{(\, ab(a+b)^*, \tau\,) \mid \tau_2 - \tau_1 = 1\}$$

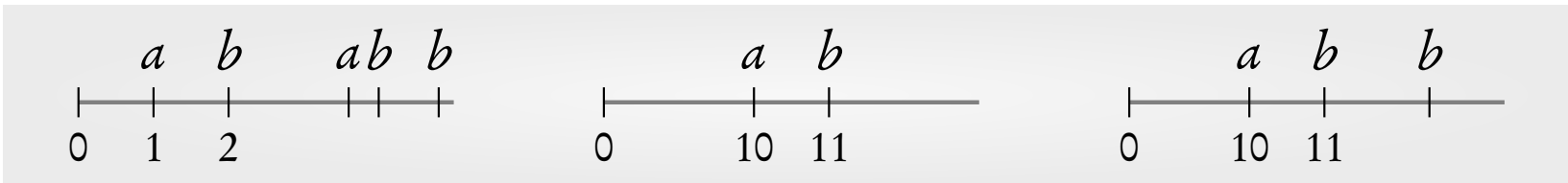


$$L_2 := \{\, (w, \tau) \mid \tau_{i+1} - \tau_i \geq 2 \text{ for all } i < |w|\}$$

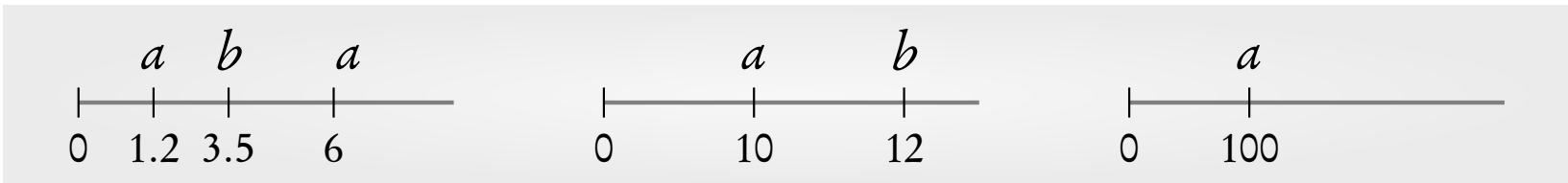


$L \subseteq T\Sigma^*$: Timed language $\longrightarrow$ *property over timed words*

$$L_1 := \{(\, ab(a+b)^*, \tau\,) \mid \tau_2 - \tau_1 = 1\}$$

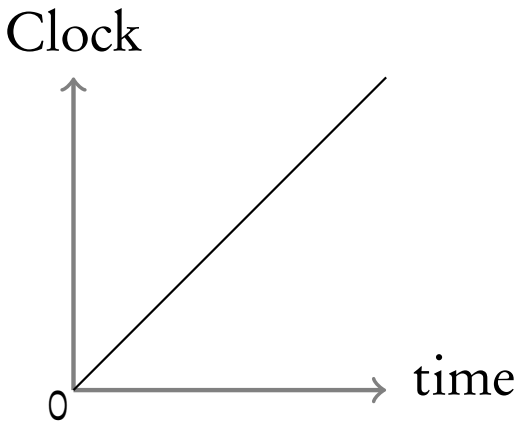


$$L_2 := \{\, (w, \tau) \mid \tau_{i+1} - \tau_i \geq 2 \text{ for all } i < |w|\}$$

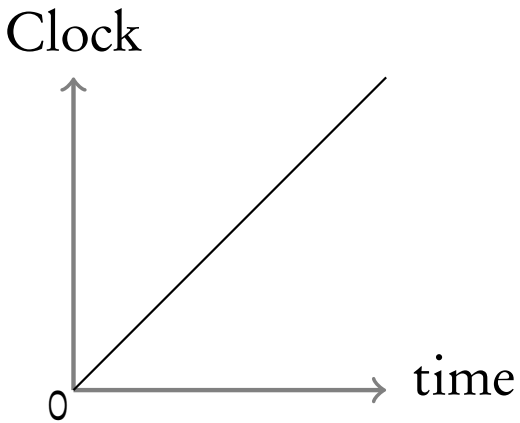


Timed automata

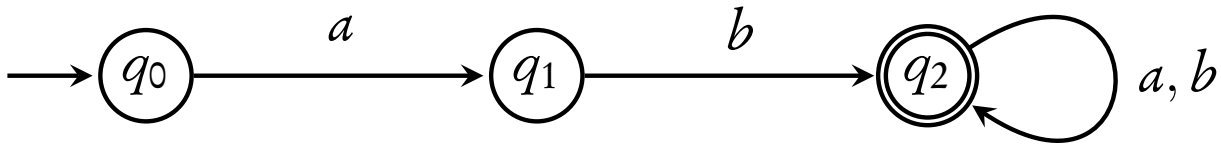
Timed automaton: Finite automaton + Finite no. of *Clocks*



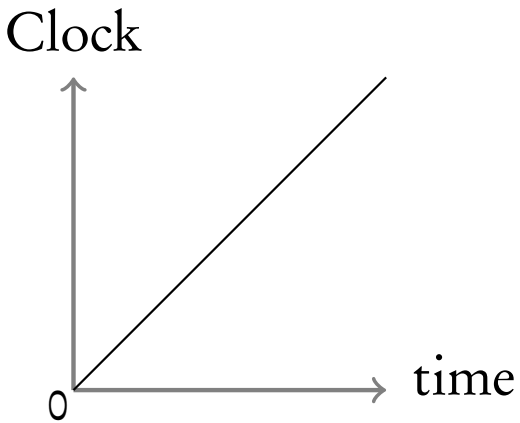
Timed automaton: Finite automaton + Finite no. of *Clocks*



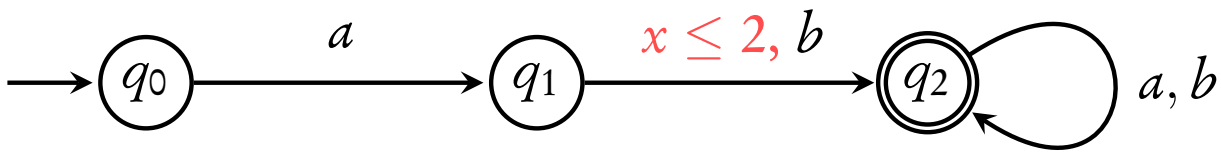
$$\{(\, ab(a+b)^*, \tau) \mid \tau_2 \leq 2\}$$



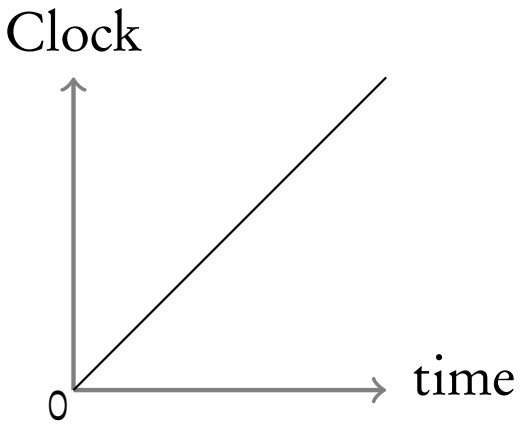
Timed automaton: Finite automaton + Finite no. of *Clocks*



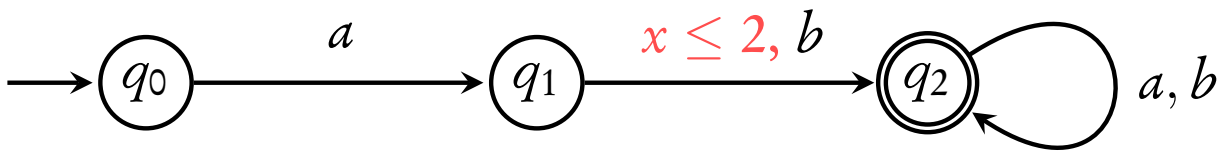
$$\{(\, ab(a+b)^*, \tau) \mid \tau_2 \leq 2\}$$

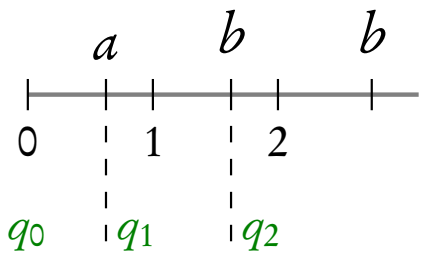


Timed automaton: Finite automaton + Finite no. of *Clocks*

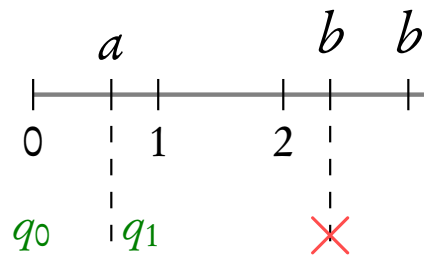


$$\{(\, ab(a+b)^*, \tau) \mid \tau_2 \leq 2\}$$



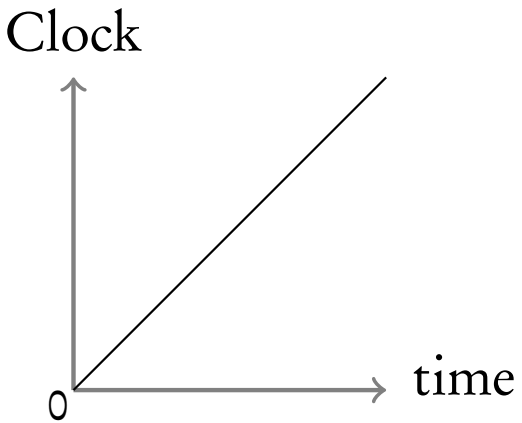


accept



reject

Timed automaton: Finite automaton + Finite no. of *Clocks*

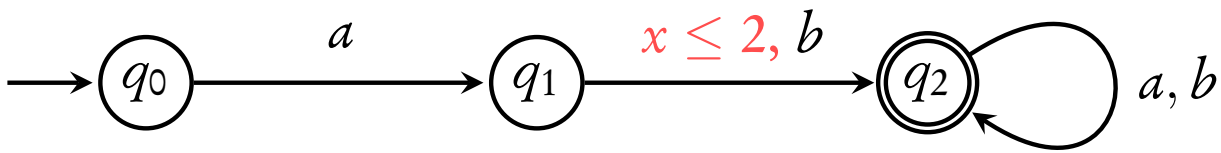


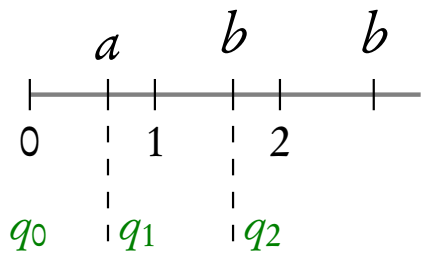
Guards

$$\phi := x \leq c \mid x \geq c \mid \neg\phi \mid \phi \wedge \phi$$

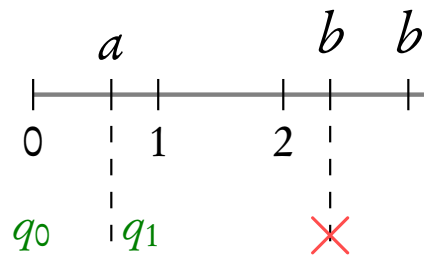
$$x \in \mathit{Clocks}\,,\ c \in \mathbb{Q}_{\geq 0}$$

$$\{(\, ab(a+b)^*, \tau) \mid \tau_2 \leq 2\}$$



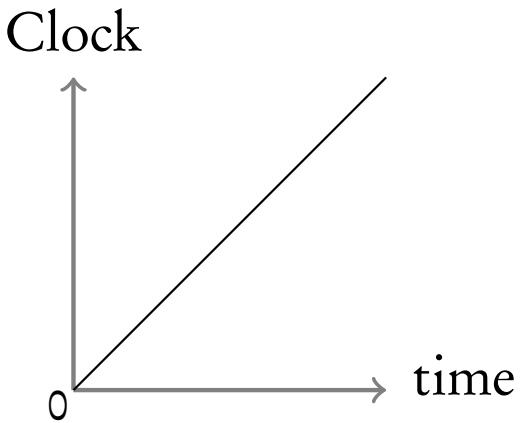


accept



reject

Timed automaton: Finite automaton + Finite no. of *Clocks*

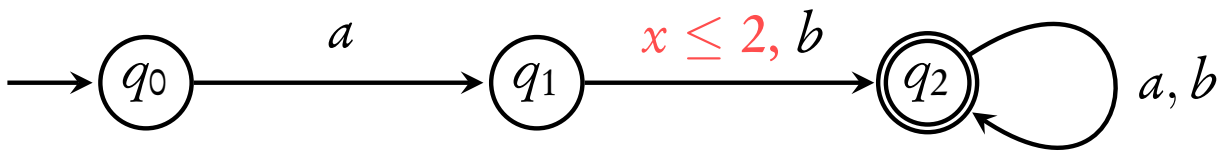


Guards

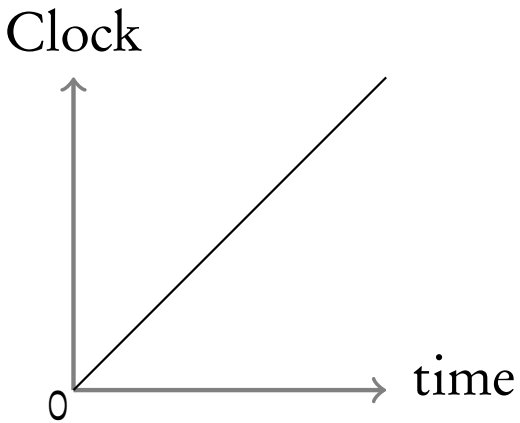
$$\phi := x \leq c \mid x \geq c \mid \neg\phi \mid \phi \wedge \phi$$

$$x \in \mathit{Clocks}\,,\ c \in \mathbb{Q}_{\geq 0}$$

$$\{(\, ab(a+b)^*, \tau) \mid \tau_2 - \tau_1 \leq 2\}$$



Timed automaton: Finite automaton + Finite no. of *Clocks*



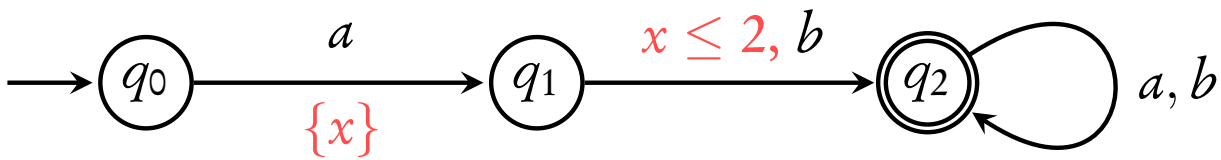
Guards

$$\phi := x \leq c \mid x \geq c \mid \neg\phi \mid \phi \wedge \phi$$

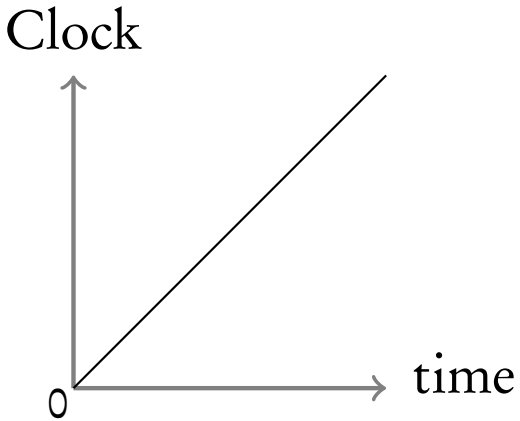
$$x \in \mathit{Clocks}\,,\ c \in \mathbb{Q}_{\geq 0}$$

Resets

$$\{(\, ab(a+b)^*, \tau) \mid \tau_2 - \tau_1 \leq 2\}$$



Timed automaton: Finite automaton + Finite no. of *Clocks*



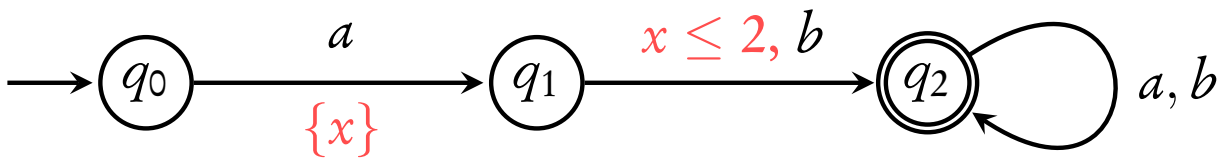
Guards

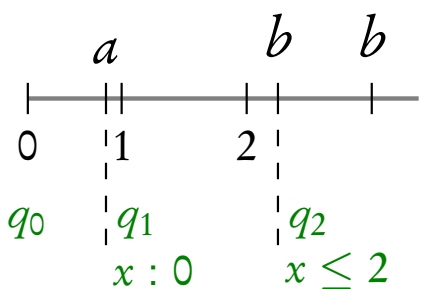
$$\phi := x \leq c \mid x \geq c \mid \neg\phi \mid \phi \wedge \phi$$

$$x \in \mathit{Clocks}\,,\ c \in \mathbb{Q}_{\geq 0}$$

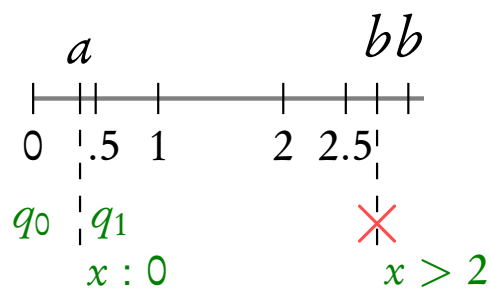
Resets

$$\{(\, ab(a+b)^*, \tau) \mid \tau_2 - \tau_1 \leq 2\}$$





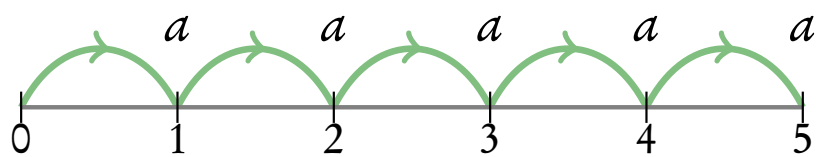
accept



reject

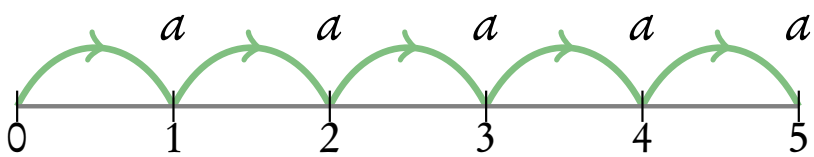
$$L_3 := \{ (a^k, \tau) \mid k > 0,\ \tau_i = i \text{ for all } i \leq k \}$$

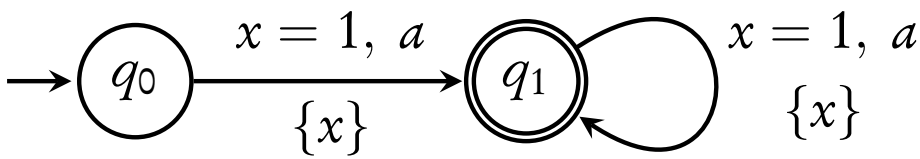
An "a" occurs in every integer from $1, \ldots, k$



$$L_3 := \{ (a^k, \tau) \mid k > 0,\ \tau_i = i \text{ for all } i \leq k \}$$

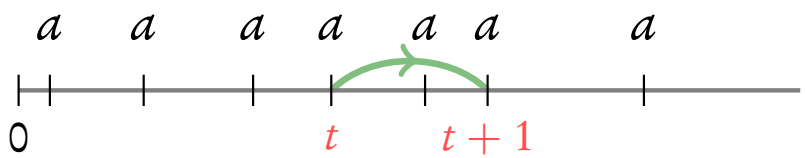
An "a" occurs in every integer from $1, \ldots, k$





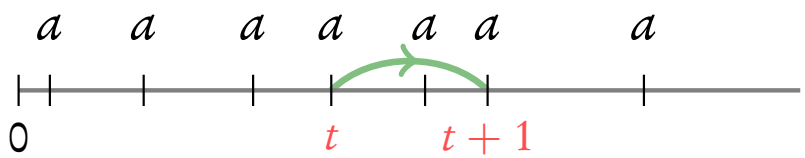
$$L_4 := \{ (a^k, \tau) \mid \text{exist } i, j \text{ s.t. } \tau_j - \tau_i = 1 \}$$

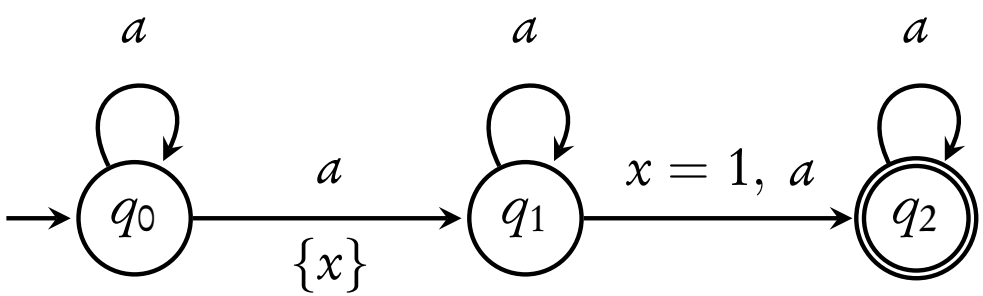
There are 2 "a"s which are at distance 1 apart



$$L_4 := \{ (a^k, \tau) \mid \text{exist } i, j \text{ s.t. } \tau_j - \tau_i = 1 \}$$

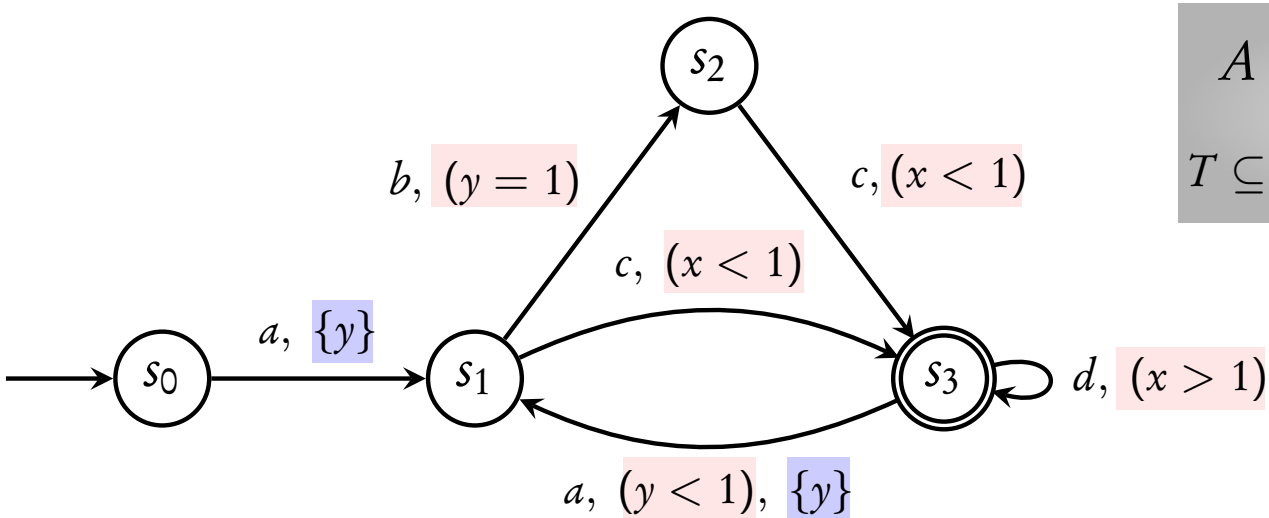
There are 2 "a"s which are at distance 1 apart





Three **mechanisms** to exploit:

- Reset: to **start** measuring time
- Guard: to **impose** time constraint on action
- Non-determinism: for **existential** time constraints

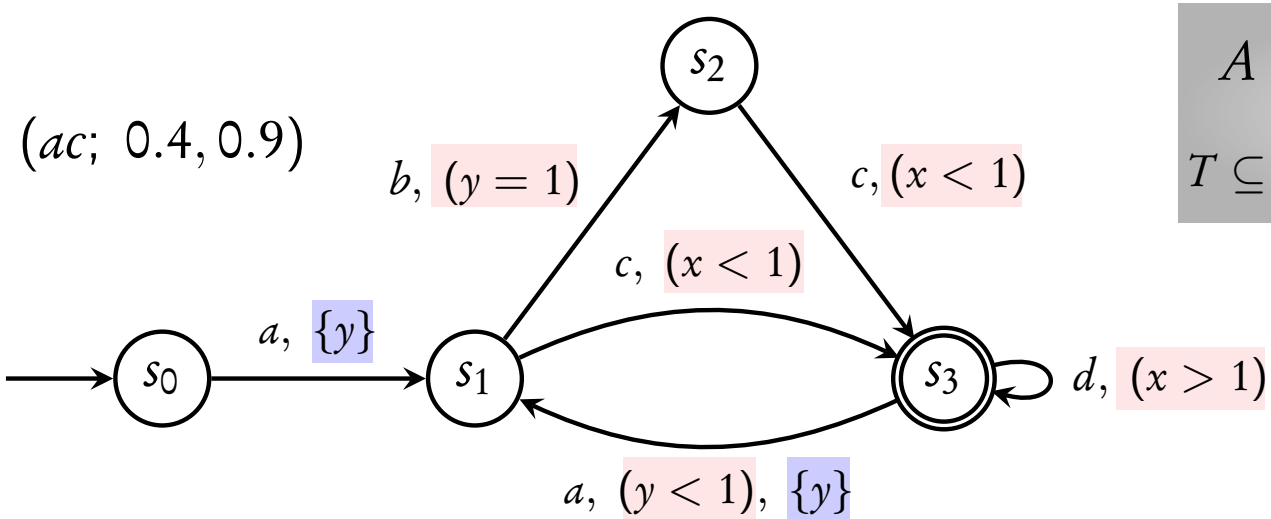


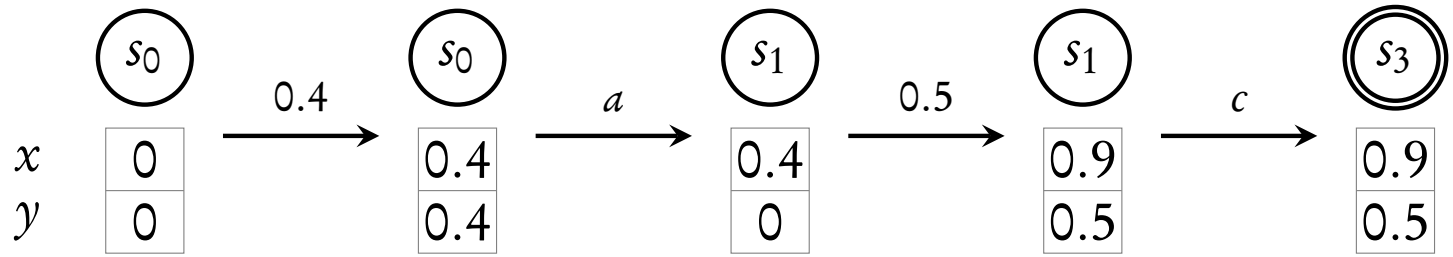
$$A = (Q, \Sigma, X, T, Q_0, F)$$

$$T \subseteq Q \times \Sigma \times \text{guard} \times \text{reset} \times Q$$

$$A = (Q, \Sigma, X, T, Q_0, F)$$

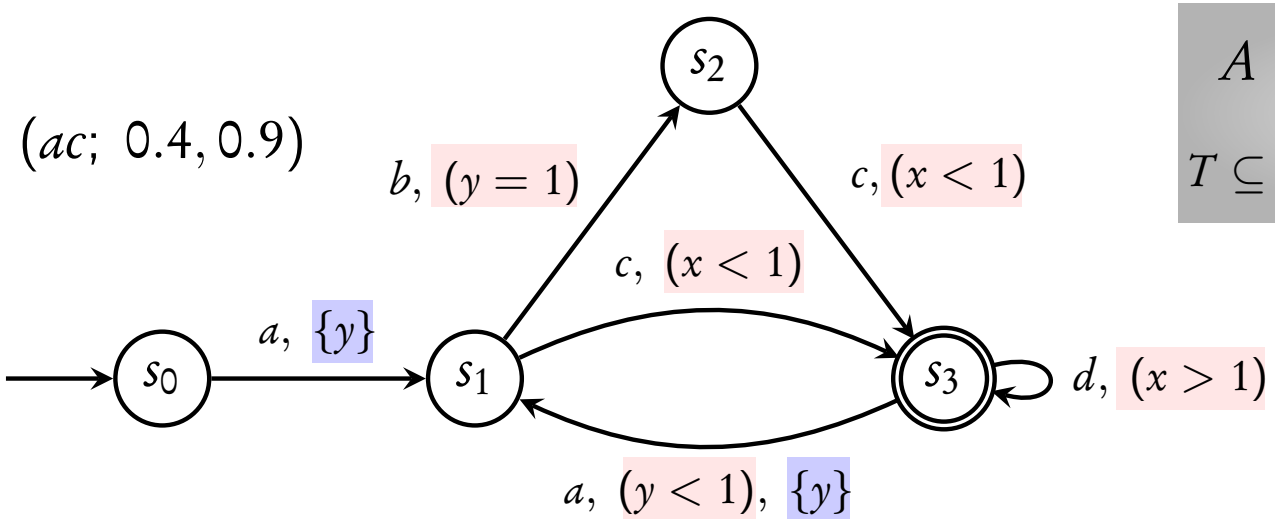
$$T \subseteq Q \times \Sigma \times \text{guard} \times \text{reset} \times Q$$

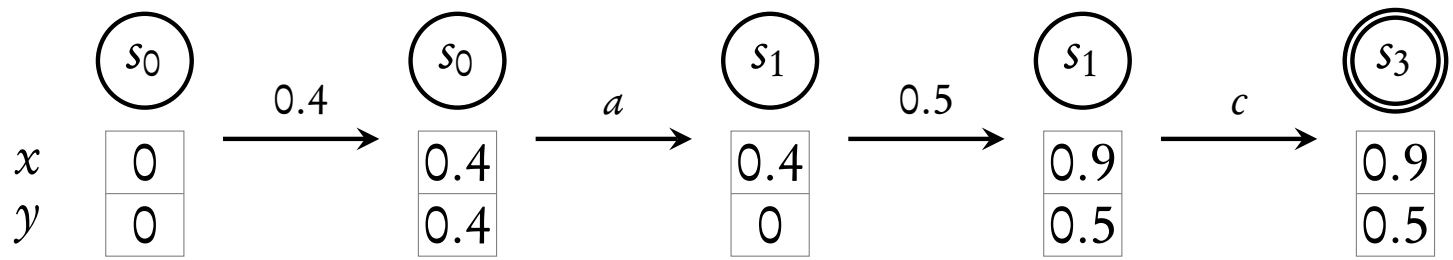




$A = (Q, \Sigma, X, T, Q_0, F)$

$T \subseteq Q \times \Sigma \times \text{guard} \times \text{reset} \times Q$





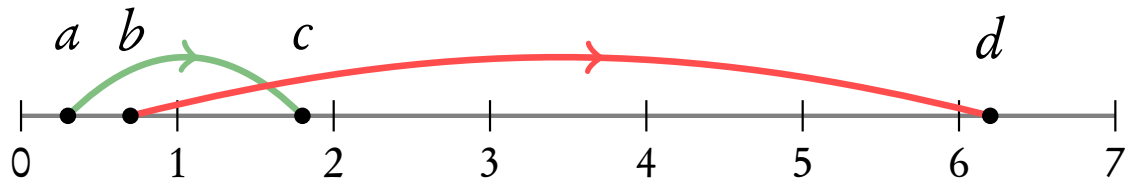
Run of A over $(a_1 a_2 \ldots a_k;\ \tau_1 \tau_2 \ldots \tau_k)$ $\qquad \delta_i := \tau_i - \tau_{i-1};\ \tau_0 := 0$

$$(q_0, v_0) \xrightarrow{\delta_1} (q_0, v_0 + \delta_1) \xrightarrow{a_1} (q_1, v_1) \xrightarrow{\delta_2} (q_1, v_1 + \delta_2) \cdots \xrightarrow{a_k} (q_k, v_k)$$

$(w, \tau) \in \mathcal{L}(A)$ if A has an **accepting** run over (w, τ)

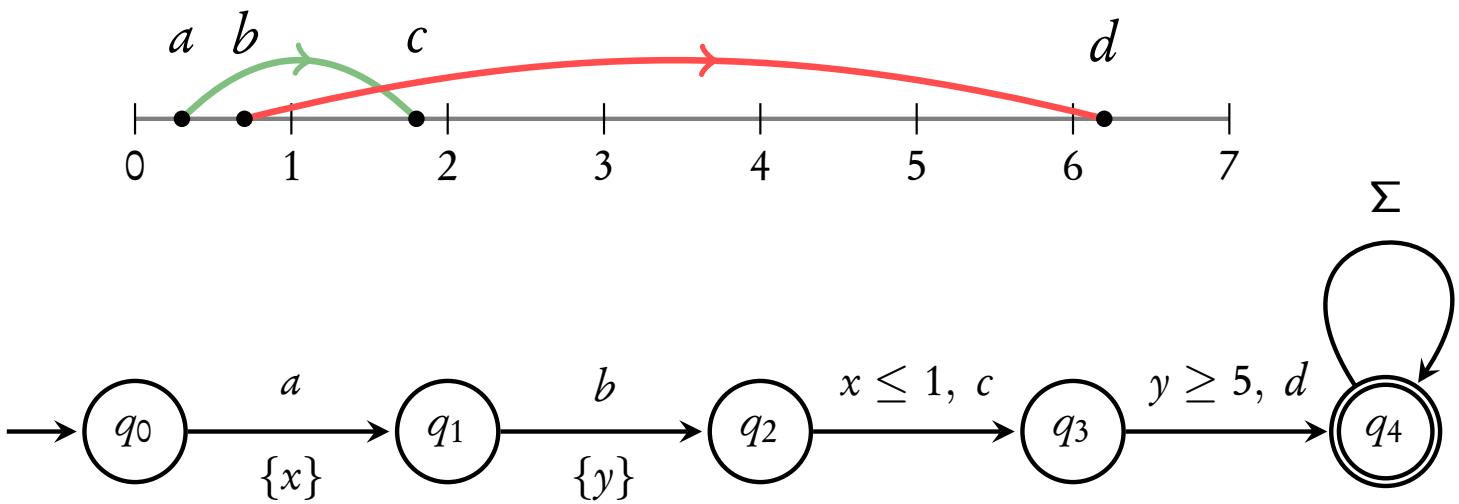
$$L_5 := \{ (\, abcd.\Sigma^*, \tau\,) \mid \tau_3 - \tau_1 \leq 2 \text{ and } \tau_4 - \tau_2 \geq 5\}$$

Interleaving distances



$$L_5 := \{ (\, abcd.\Sigma^*, \tau\,) \mid \tau_3 - \tau_1 \leq 2 \text{ and } \tau_4 - \tau_2 \geq 5\}$$

Interleaving distances



Timed automata

Examples

Formal definition

Runs