

TIMED AUTOMATA

GOALS OF TODAY'S LECTURE

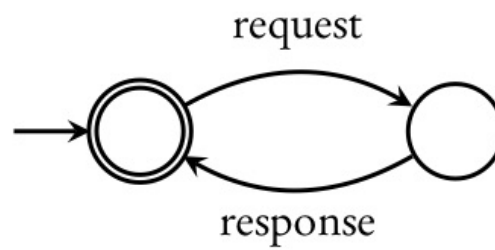
- 1. Motivations
- 2. Timed words & timed languages
- 3. Timed automaton model
- 4. Examples

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



Automaton \mathcal{A}

Property

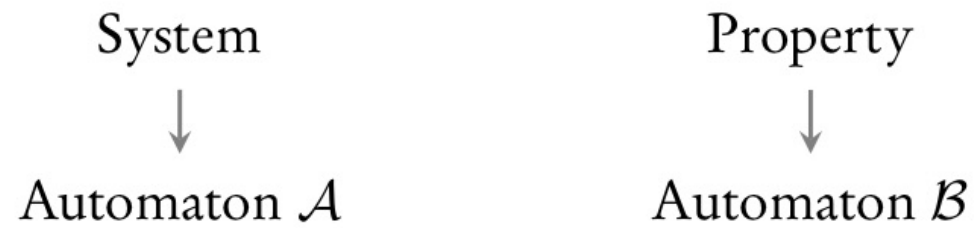


Automaton \mathcal{B}

System
↓
Automaton \mathcal{A}

Property
↓
Automaton \mathcal{B}

Does system **satisfy** property?



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

Does system **satisfy** property?

Model-checking



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

Does system **satisfy** property?

In practice...

Huge system

Property

In practice...

Huge system



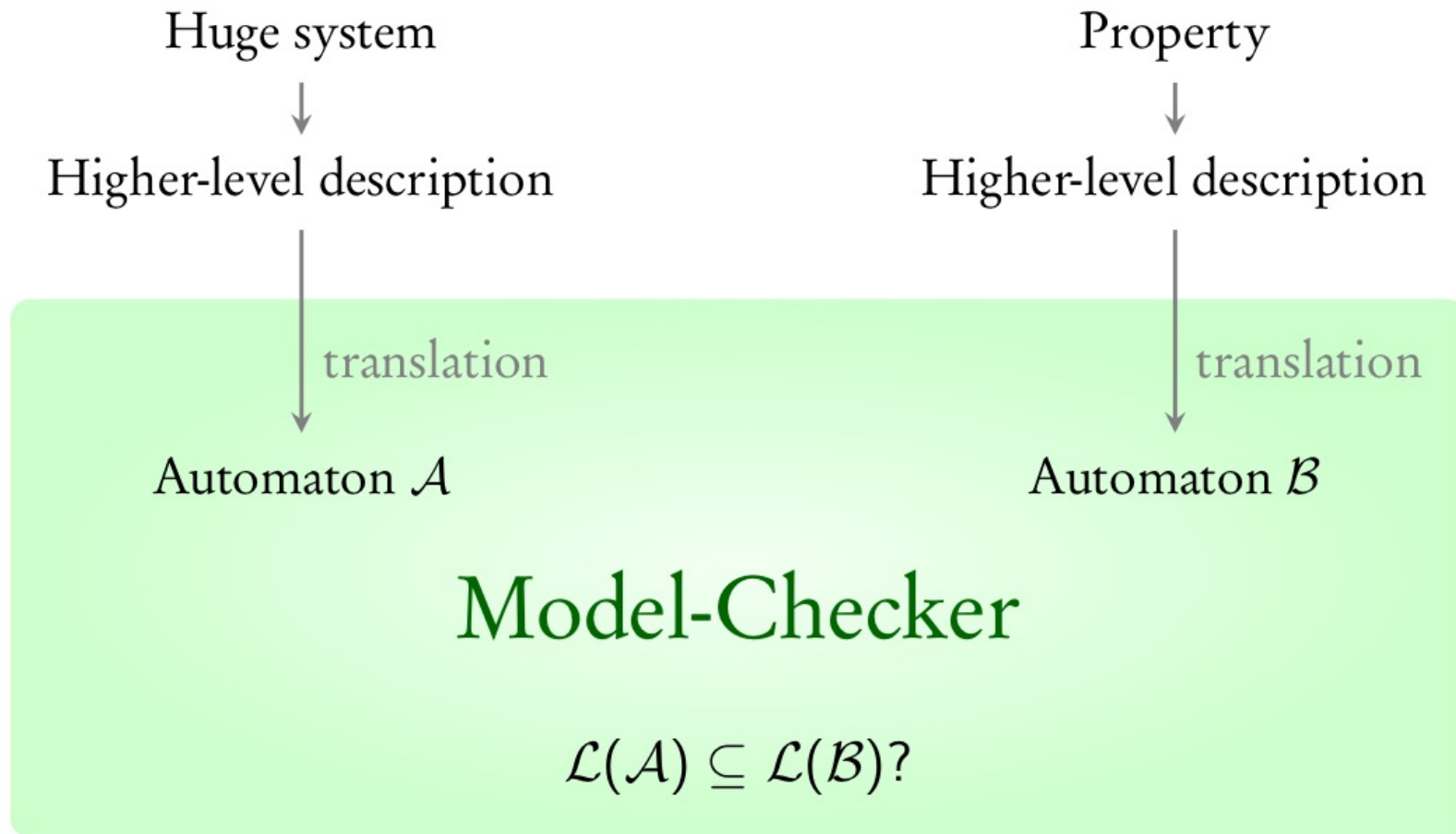
Higher-level description

Property

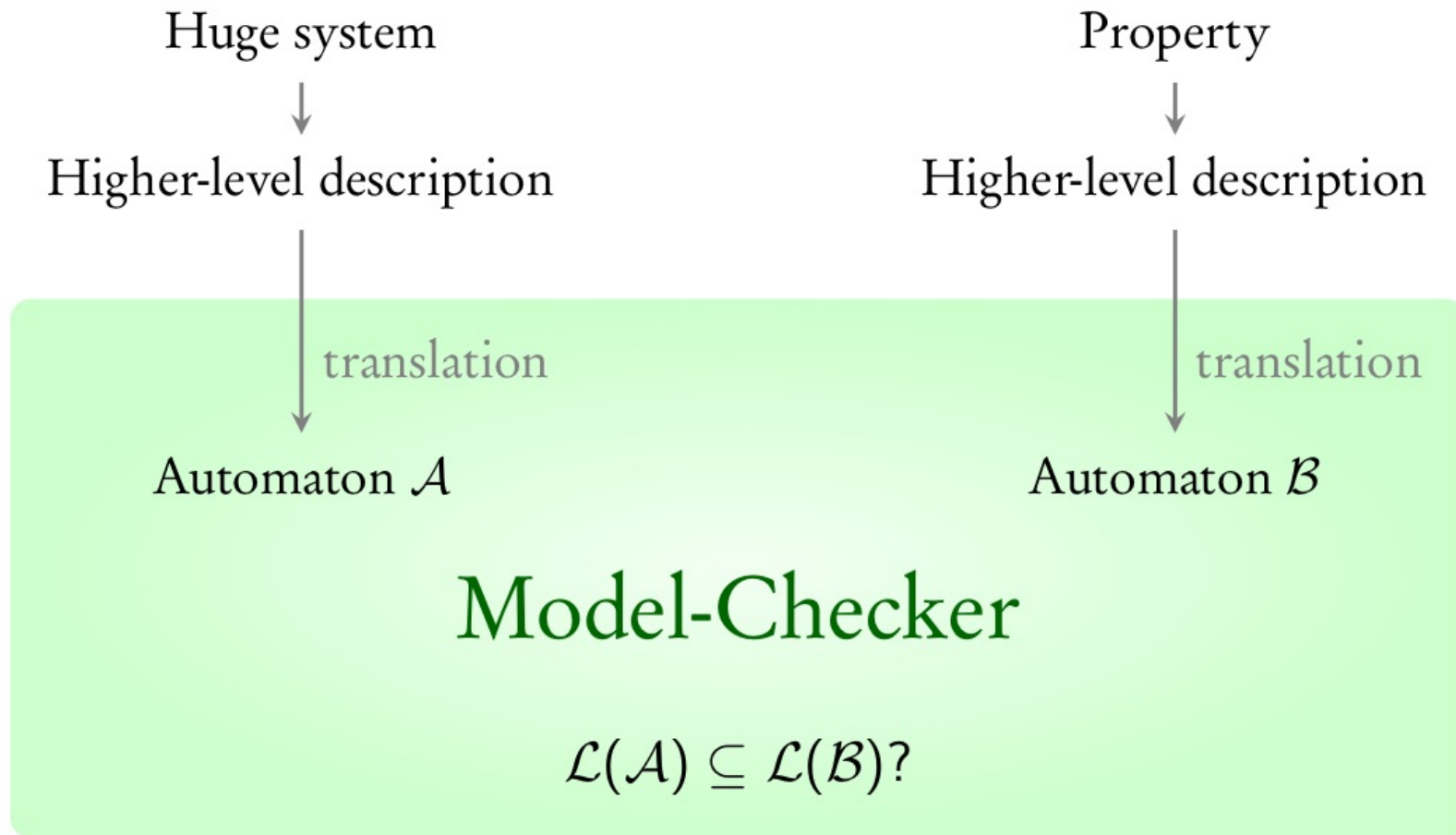


Higher-level description

In practice...

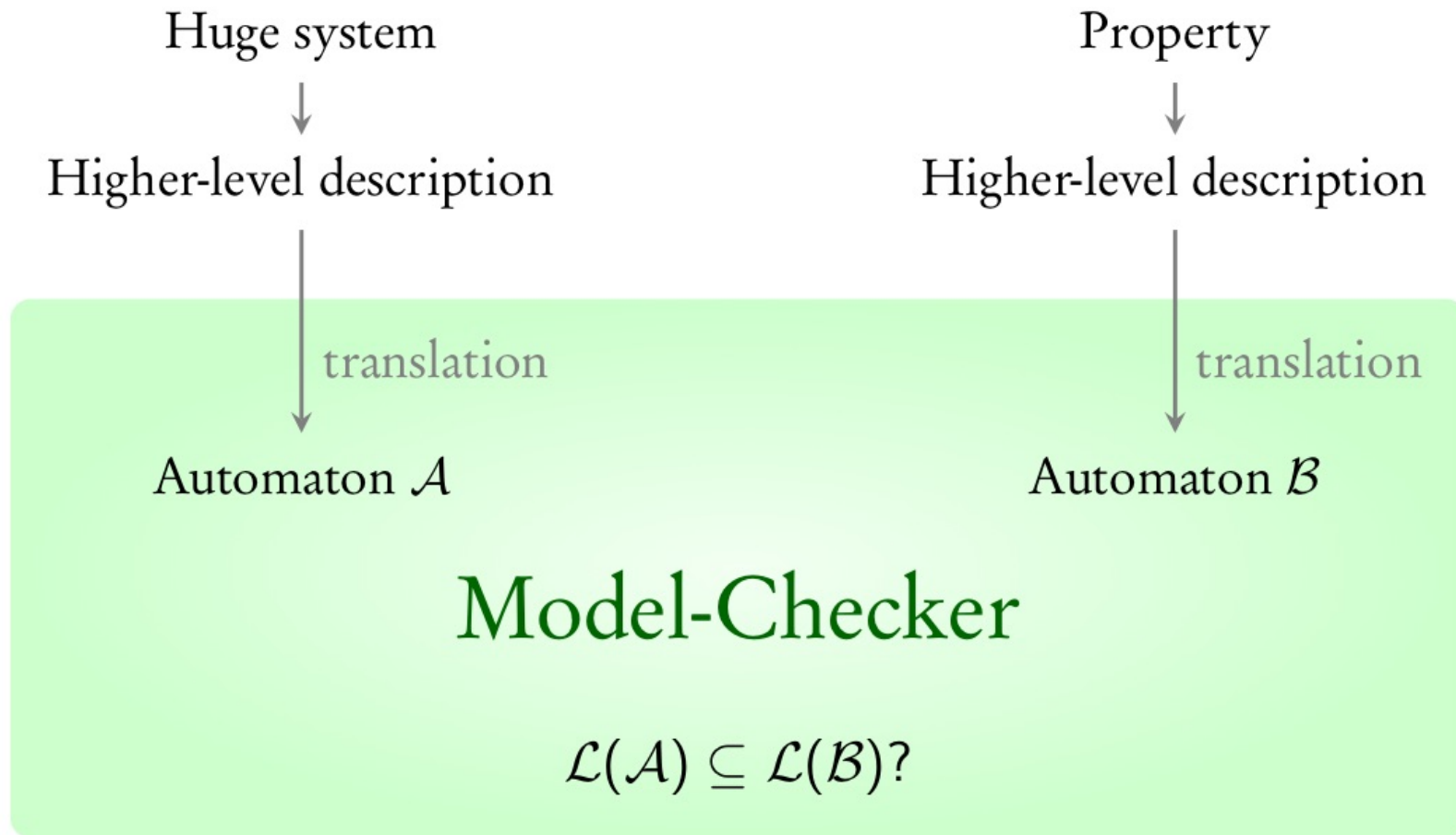


In practice...



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, ...

TCHECKER

Goals of our course

- ▶ Understand **language theoretic** properties of timed automata
- ▶ Study **algorithms** used in model-checkers

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

Timed languages and timed automata

Σ : alphabet $\{a, b\}$

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

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

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

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

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

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

Σ : alphabet $\{a, b\}$

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

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

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

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

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

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

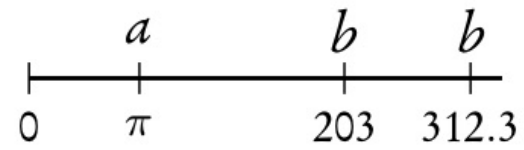
Finite automata, pushdown automata, Turing machines, ...

Σ : alphabet $\{a, b\}$

$T\Sigma^*$: timed words



$(aa; 0.8, 2.5)$



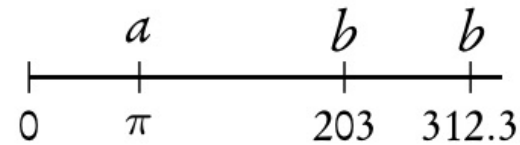
$(abb; \pi, 203, 312.3)$

Σ : alphabet $\{a, b\}$

$T\Sigma^*$: timed words



$(aa; 0.8, 2.5)$



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

abb,
1 2.2 2.2



$$w = a_1 \dots a_n$$

$$a_i \in \Sigma$$

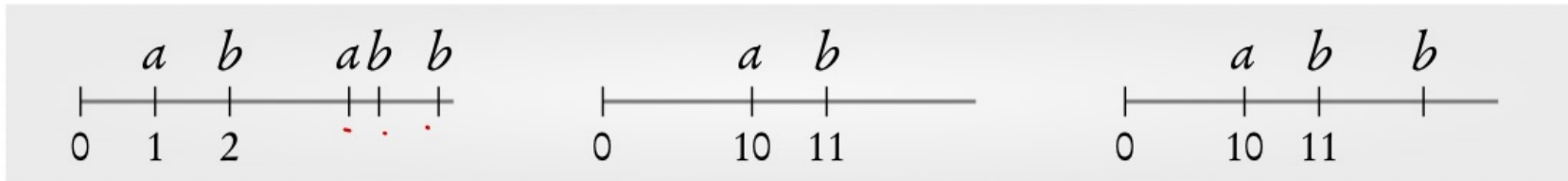
$$\tau = \tau_1 \dots \tau_n$$

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

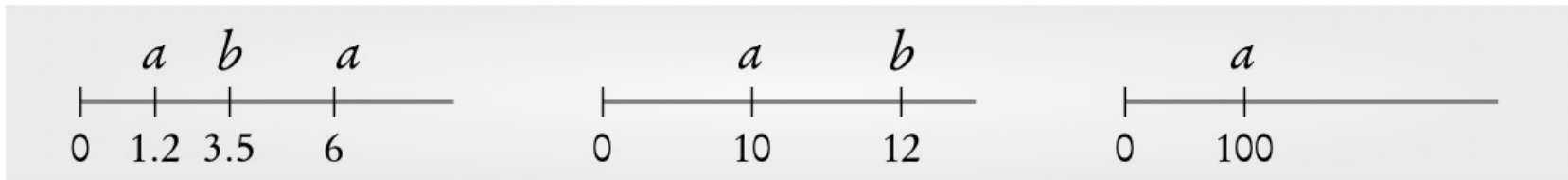
$$\tau_1 \leq \dots \leq \tau_n$$

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

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

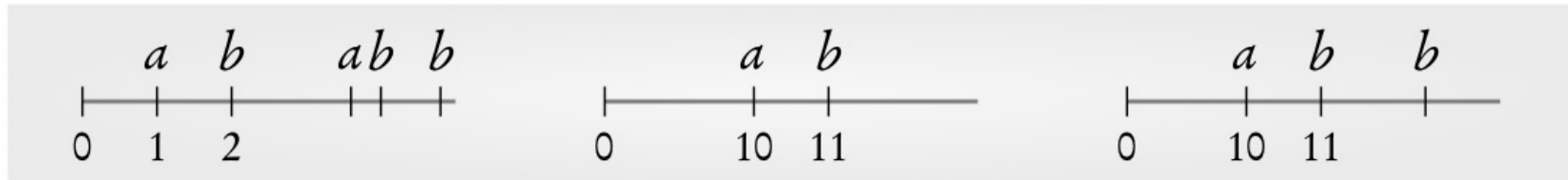


$$L_2 := \{ (\omega, \tau) \mid \tau_{i+1} - \tau_i \geq 2 \text{ for all } i < |\omega| \}$$

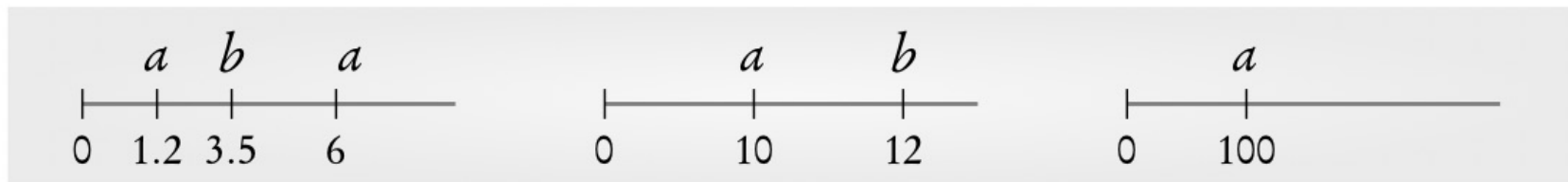


$L \subseteq T\Sigma^*$: Timed language \longrightarrow *property over timed words*

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

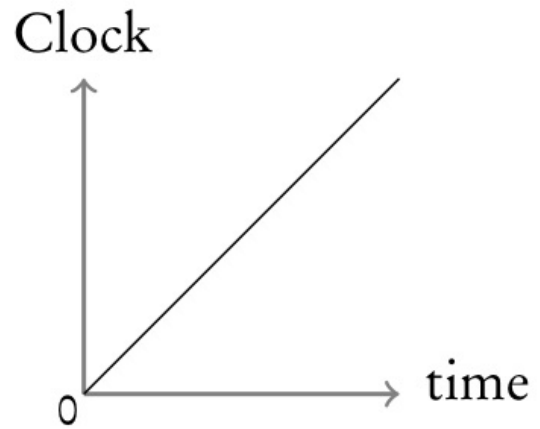


$$L_2 := \{ (\omega, \tau) \mid \tau_{i+1} - \tau_i \geq 2 \text{ for all } i < |\omega| \}$$

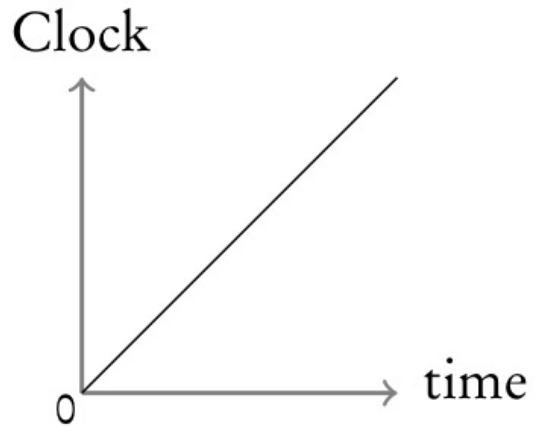


Timed automata

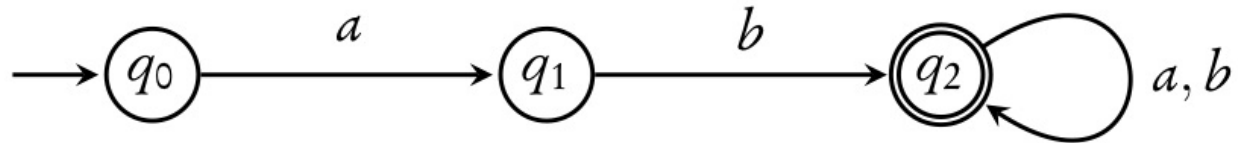
Timed automaton: Finite automaton + Finite no. of *Clocks*



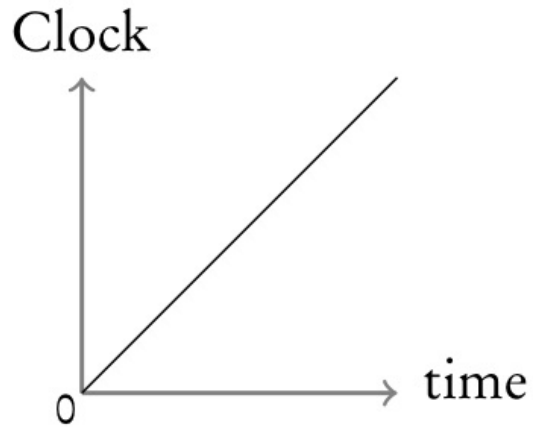
Timed automaton: Finite automaton + Finite no. of *Clocks*



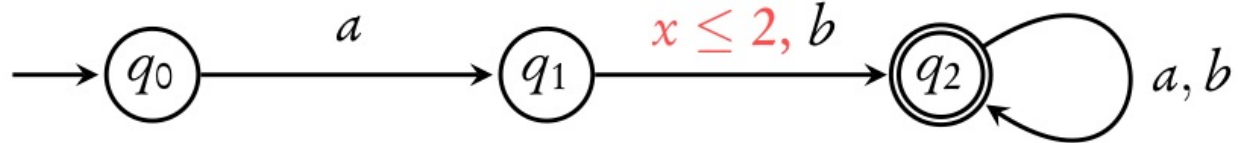
$$\{(ab(a+b)^*, \tau) \mid \tau_2 \leq 2\}$$



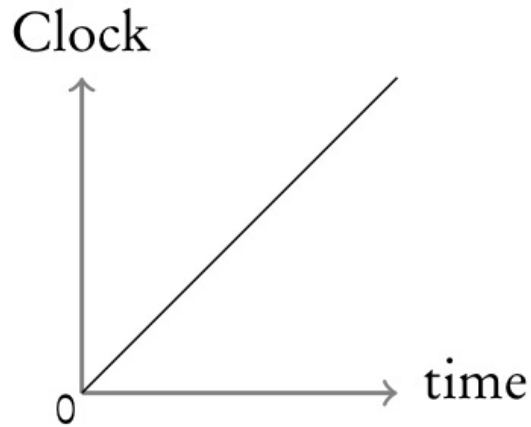
Timed automaton: Finite automaton + Finite no. of *Clocks*



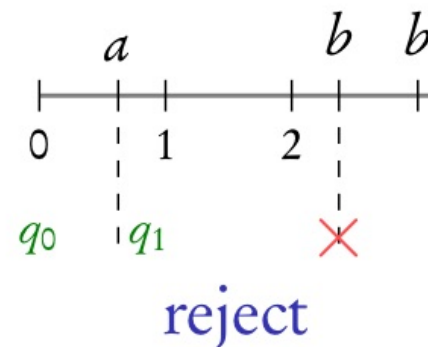
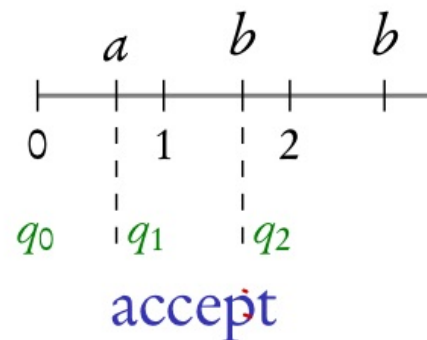
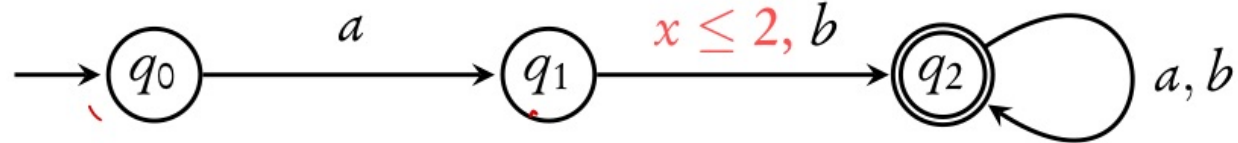
$$\{(ab(a+b)^*, \tau) \mid \tau_2 \leq 2\}$$



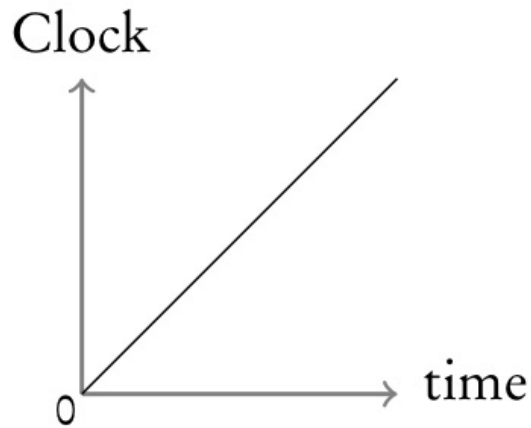
Timed automaton: Finite automaton + Finite no. of *Clocks*



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



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

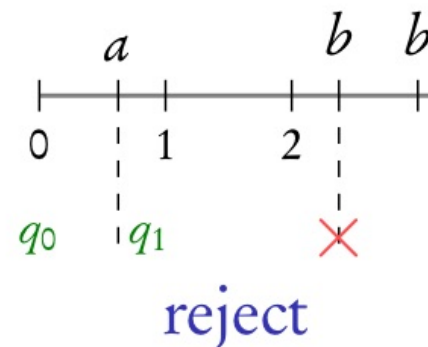
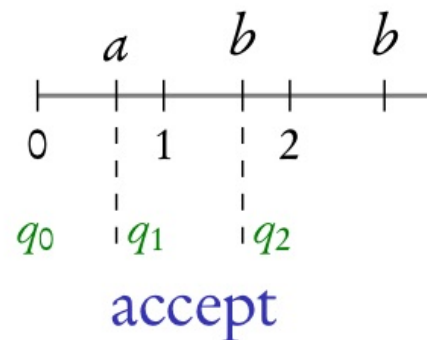
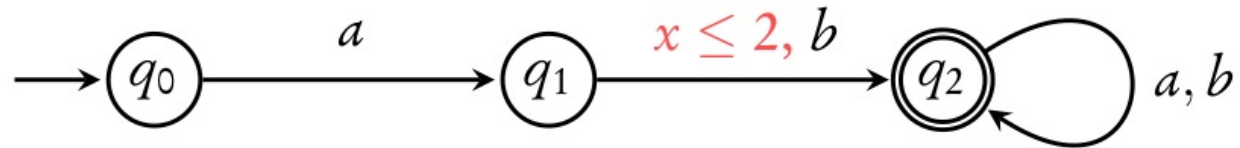


Guards

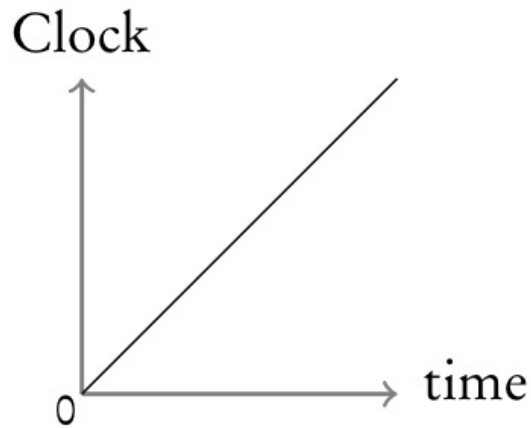
$$\phi := x \leq c \mid x \geq c \mid \neg\phi \mid \phi \wedge \phi$$

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

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



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

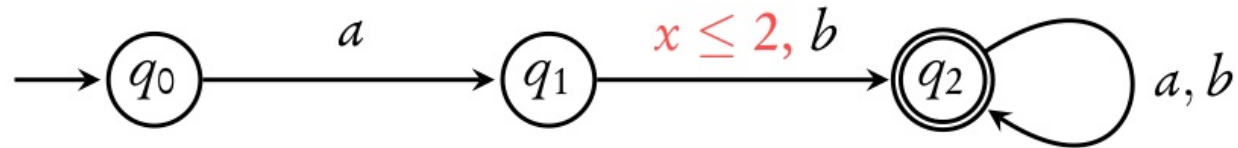


Guards

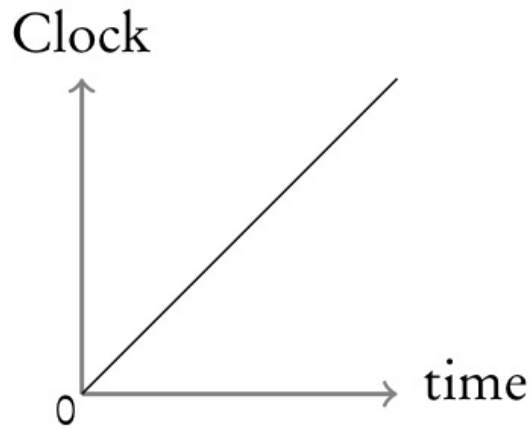
$$\phi := x \leq c \mid x \geq c \mid \neg\phi \mid \phi \wedge \phi$$

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

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



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



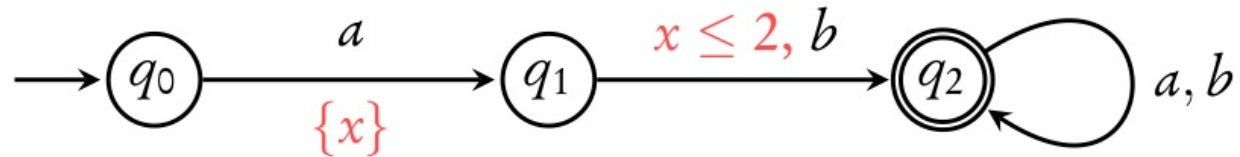
Guards

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

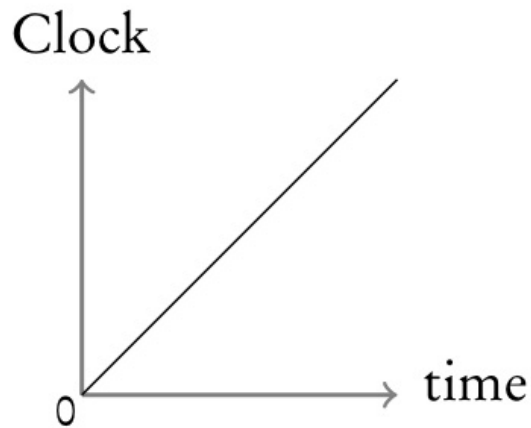
$$x \in \text{Clocks}, c \in \mathbb{Q}_{\geq 0}$$

Resets

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



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



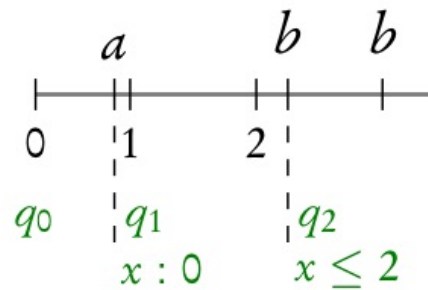
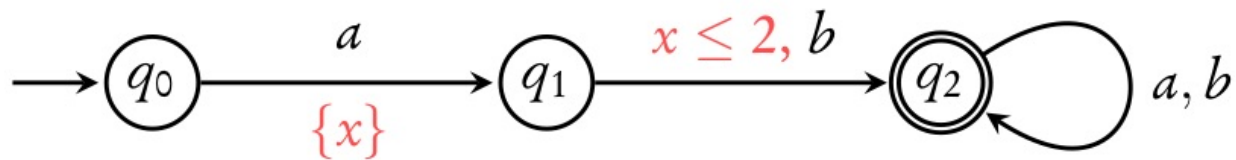
Guards

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

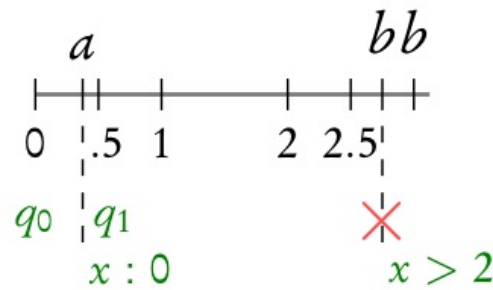
$x \in \text{Clocks}, c \in \mathbb{Q}_{\geq 0}$

Resets

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



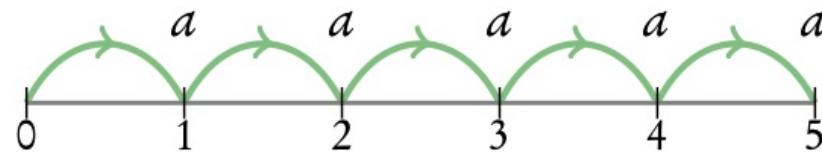
accept



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 $1, \dots, k$

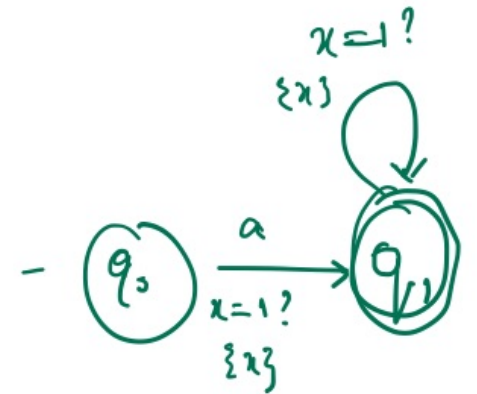


$(a, 1)$ ✓

$(a, 1.1)$ ✗

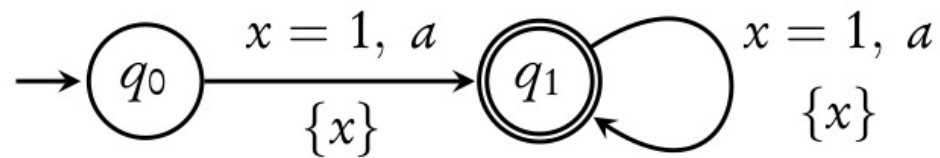
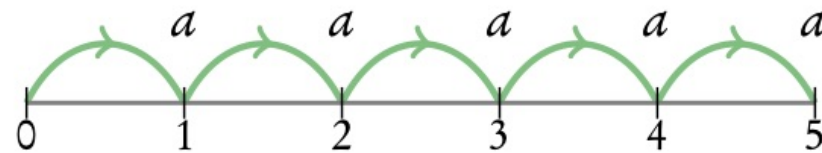
$(aa, 1, 2)$ ✓

$(aaaa, 1, 3, 5, 6)$ ✗



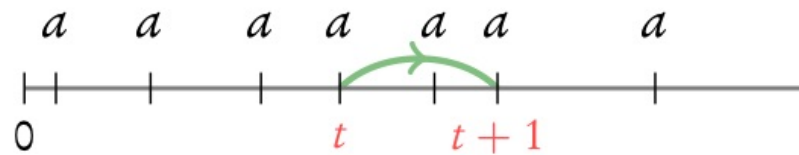
$$L_3 := \{ (a^k, \tau) \mid k > 0, \tau_i = i \text{ for all } i \leq k \}$$

An “ a ” occurs in every integer from $1, \dots, k$



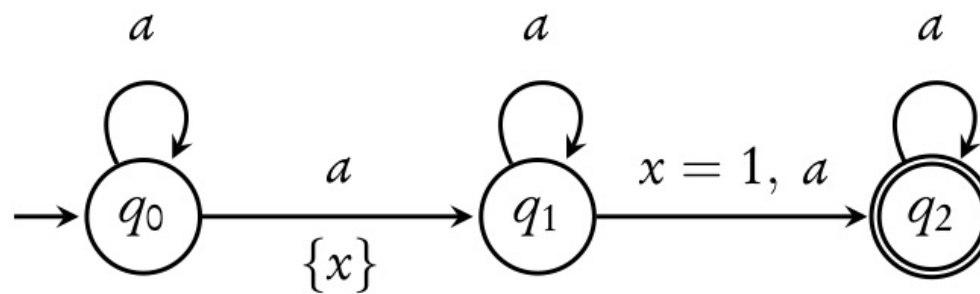
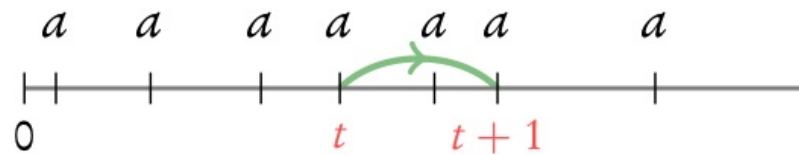
$$L_4 := \{ (a^k, \tau) \mid \text{exist } i, j \text{ s.t. } \tau_j - \tau_i = 1 \}$$

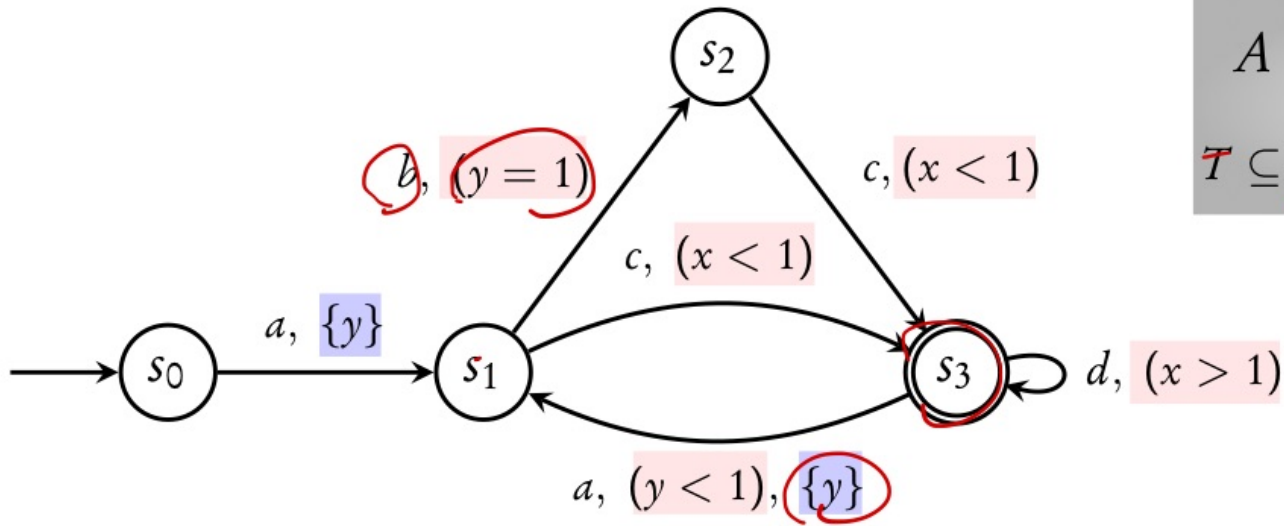
There are 2 “ a ”s which are at distance 1 apart



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

There are 2 “a”s which are at distance 1 apart





$A = (Q, \Sigma, X, T, Q_0, F)$
 $T \subseteq Q \times \Sigma \times \text{guard} \times \text{reset} \times Q$

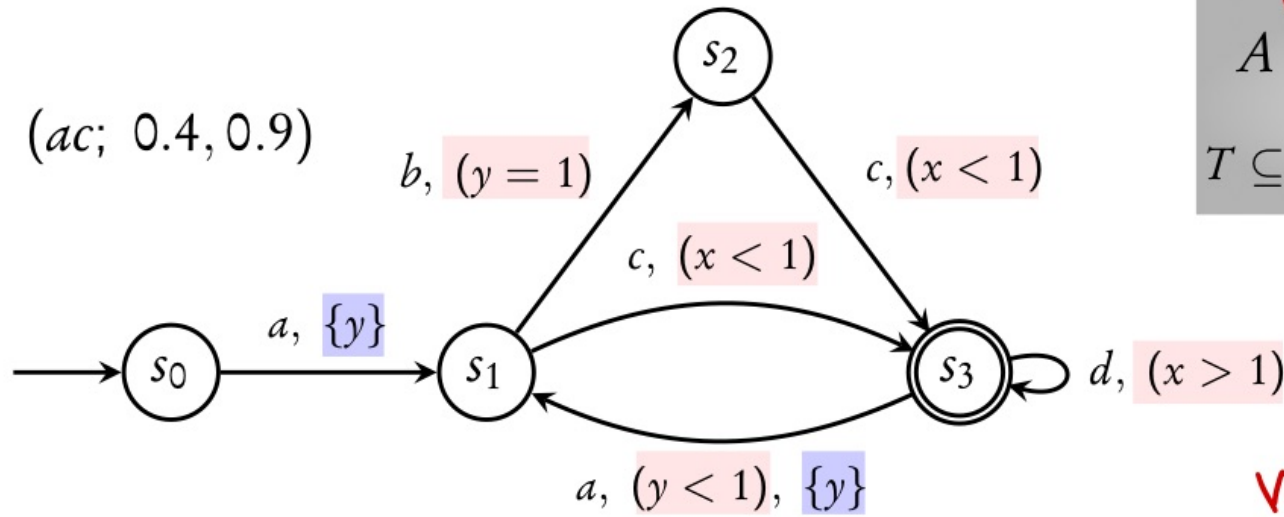
Handwritten annotations: "Clocks" points to X , "Transitions" points to T , "Initial" points to Q_0 , and "final" points to F .

$\Phi(x) \ 2^x$

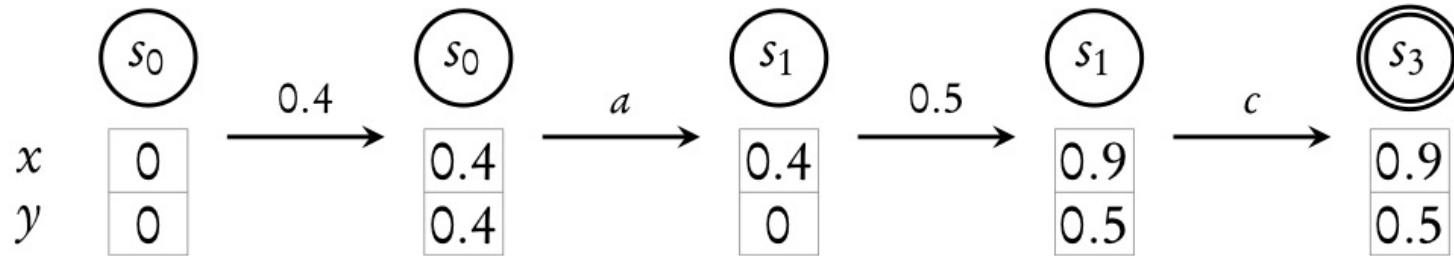
Semantics is a timed language: set of timed words that are accepted by the automaton

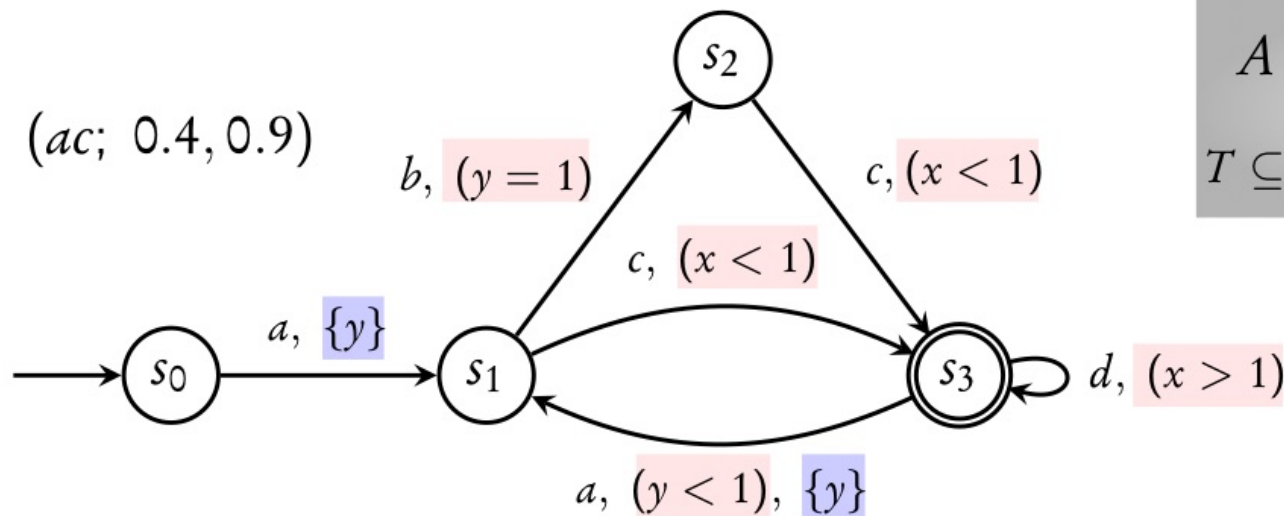
$$A = (Q, \Sigma, X, T, Q_0, F)$$

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



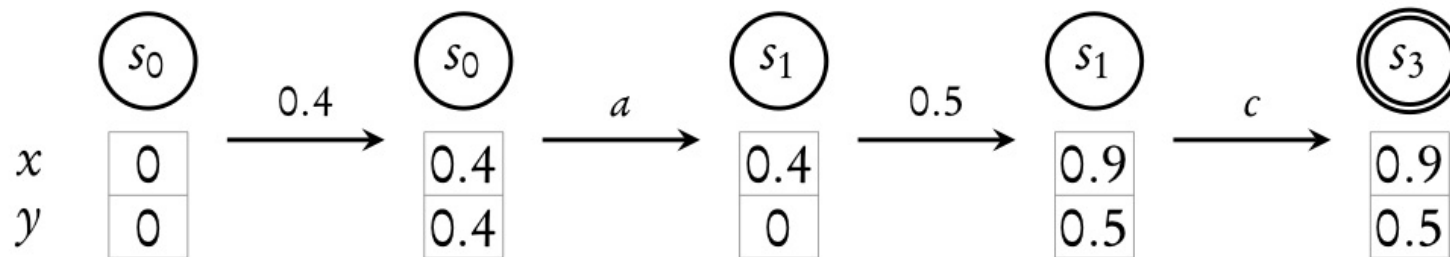
Valuation: tuple of reals $\mathbb{R}_{\geq 0}^{|X|}$





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

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



Run of A over $(a_1 a_2 \dots a_k; \tau_1 \tau_2 \dots \tau_k)$

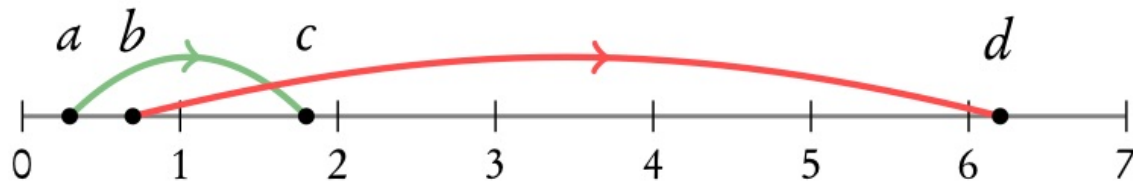
$$\delta_i := \tau_i - \tau_{i-1}; \tau_0 := 0$$

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

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

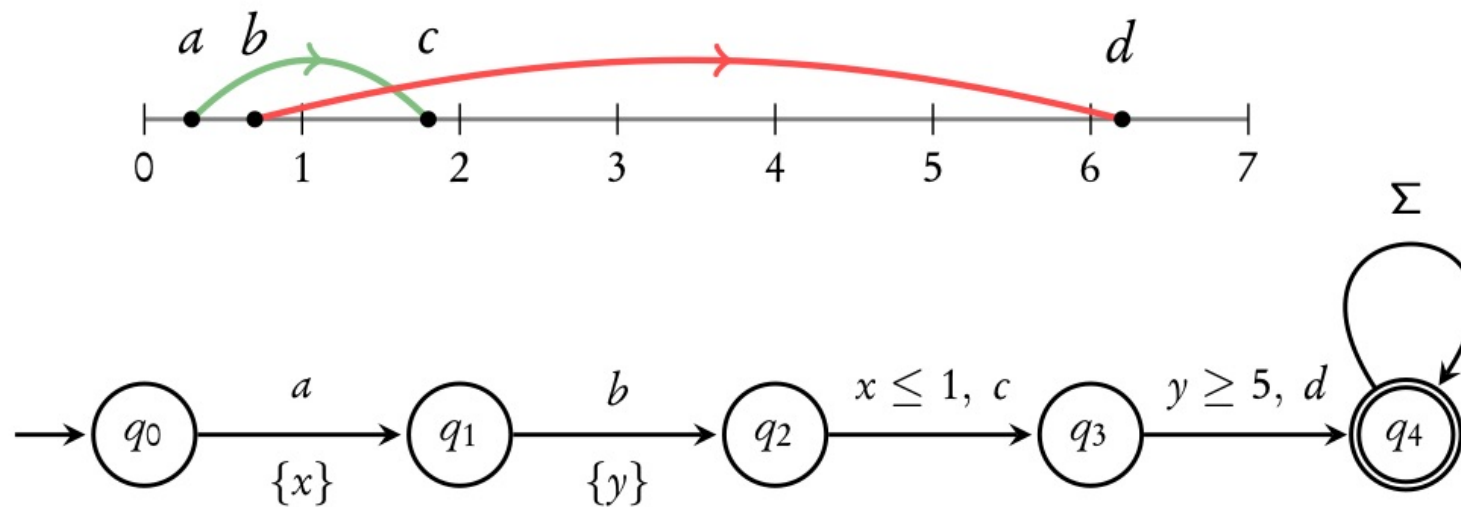
$$L_5 := \{ (abcd.\Sigma^*, \tau) \mid \tau_3 - \tau_1 \leq 2 \text{ and } \tau_4 - \tau_2 \geq 5 \}$$

Interleaving distances



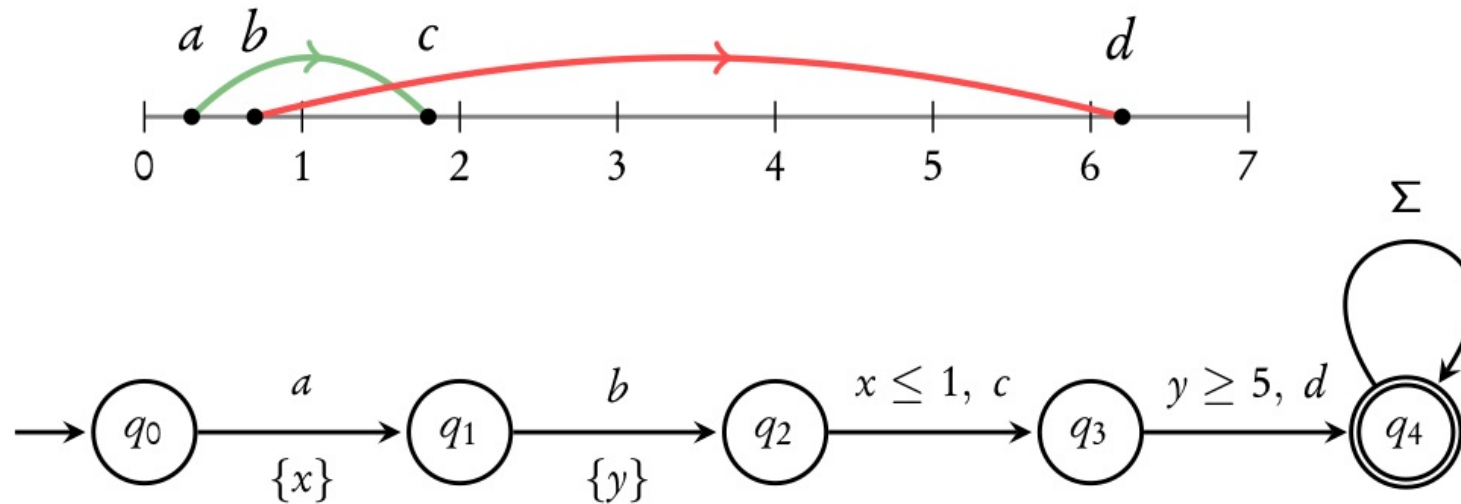
$$L_5 := \{ (abcd.\Sigma^*, \tau) \mid \tau_3 - \tau_1 \leq 2 \text{ and } \tau_4 - \tau_2 \geq 5 \}$$

Interleaving distances



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

Interleaving distances



Exercise: Prove that L_5 cannot be accepted by a one-clock TA.

n interleavings \Rightarrow need n clocks

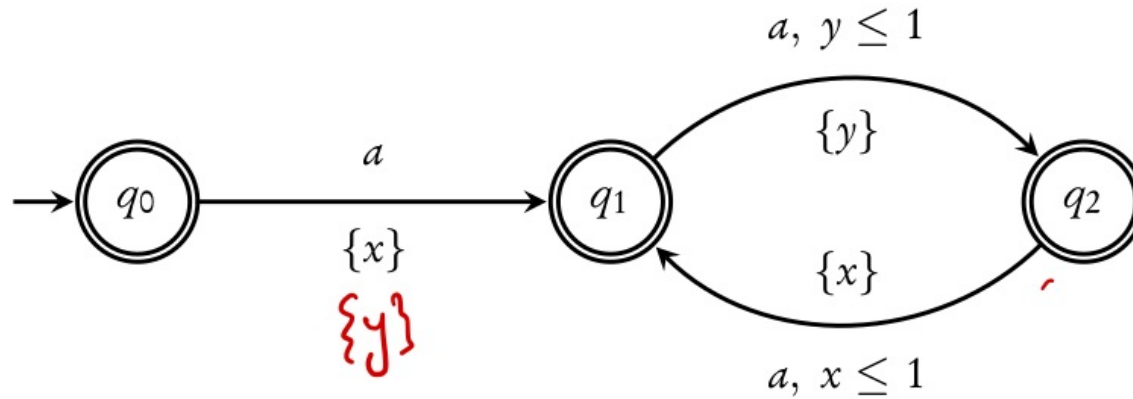
$n + 1$ clocks more expressive than n clocks

$$\{ (a^k, \tau) \mid \tau_{i+2} - \tau_i \leq 1 \text{ for all } i \leq k - 2 \}$$

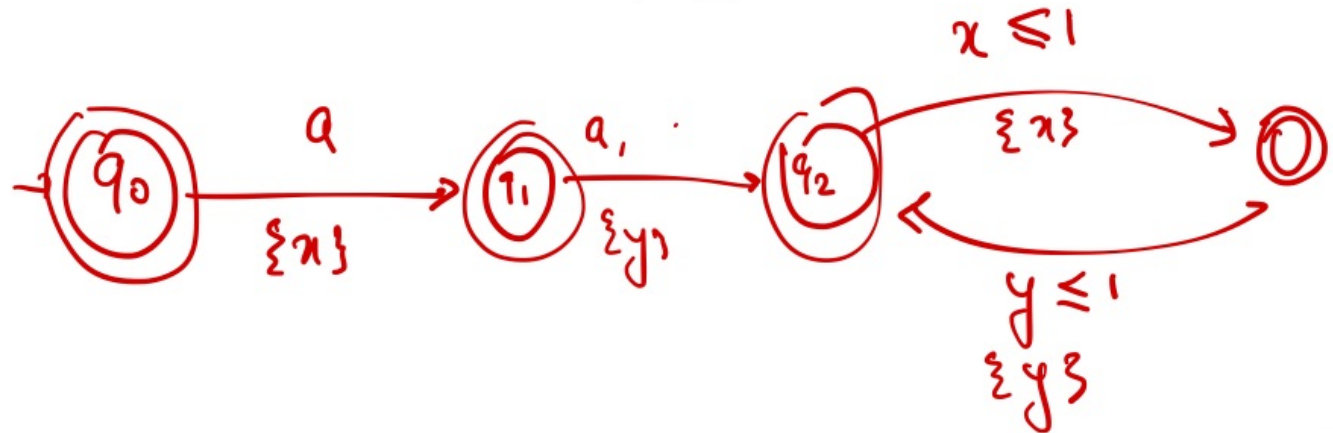


$$\{ (a^k, \tau) \mid \tau_{i+2} - \tau_i \leq 1 \text{ for all } i \leq k - 2 \}$$

$$\begin{aligned} \tau_0 &= 0 \\ 0 &\leq i \leq k-2 \\ k &\geq 0 \\ \tau_2 - \tau_0 &\leq 1 \end{aligned}$$



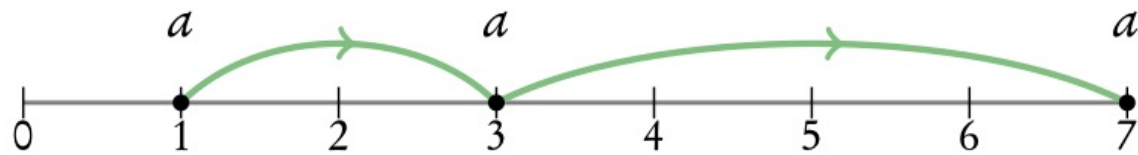
$a \leftarrow a$ a
2 2.7 2.8



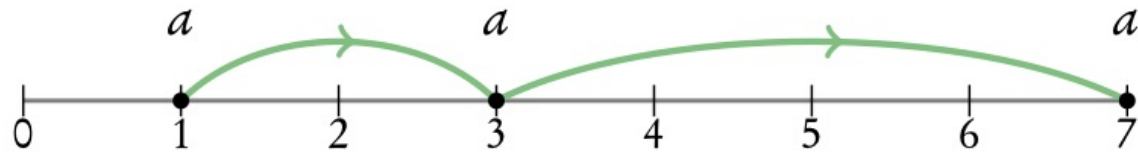
SEEN SO FAR...

- Runs of a timed automaton
- 1 clock < 2 clock <

$$L_6 := \{ (a^k, \tau) \mid \tau_i \text{ is some integer for each } i \}$$



$$L_6 := \{ (a^k, \tau) \mid \tau_i \text{ is some integer for each } i \}$$



Claim: No timed automaton can accept L_6

Step 1: *Suppose* $L_6 = \mathcal{L}(A)$

Let c_{max} be the maximum constant appearing in a guard of A

Step 1: *Suppose* $L_6 = \mathcal{L}(A)$

Let c_{max} be the maximum constant appearing in a guard of A

Step 2: For a clock x ,

$$x = \lceil c_{max} \rceil + 1 \quad \text{and} \quad x = \lceil c_{max} \rceil + 1.1$$

satisfy the same guards

Step 1: *Suppose* $L_6 = \mathcal{L}(A)$

Let c_{max} be the maximum constant appearing in a guard of A

Step 2: For a clock x ,

$$x = \lceil c_{max} \rceil + 1 \text{ and } x = \lceil c_{max} \rceil + 1.1$$

satisfy the same guards

Step 3: $(a; \lceil c_{max} \rceil + 1) \in L_6$ and so A has an accepting run

$$(q_0, v_0) \xrightarrow{\delta = \lceil c_{max} \rceil + 1} (q_0, v_0 + \delta) \xrightarrow{a} (q_F, v_F)$$

Step 1: *Suppose* $L_6 = \mathcal{L}(A)$

Let c_{max} be the maximum constant appearing in a guard of A

Step 2: For a clock x ,

$$x = \lceil c_{max} \rceil + 1 \text{ and } x = \lceil c_{max} \rceil + 1.1$$

satisfy the same guards

Step 3: $(a; \lceil c_{max} \rceil + 1) \in L_6$ and so A has an accepting run

$$(q_0, v_0) \xrightarrow{\delta = \lceil c_{max} \rceil + 1} (q_0, v_0 + \delta) \xrightarrow{a} (q_F, v_F)$$

Step 4: By Step 2, the following is an accepting run

$$(q_0, v_0) \xrightarrow{\delta' = \lceil c_{max} \rceil + 1.1} (q_0, v_0 + \delta') \xrightarrow{a} (q_F, v'_F)$$

Step 1: *Suppose* $L_6 = \mathcal{L}(A)$

Let c_{max} be the maximum constant appearing in a guard of A

Step 2: For a clock x ,

$$x = \lceil c_{max} \rceil + 1 \text{ and } x = \lceil c_{max} \rceil + 1.1$$

satisfy the same guards

Step 3: $(a; \lceil c_{max} \rceil + 1) \in L_6$ and so A has an accepting run

$$(q_0, v_0) \xrightarrow{\delta = \lceil c_{max} \rceil + 1} (q_0, v_0 + \delta) \xrightarrow{a} (q_F, v_F)$$

Step 4: By Step 2, the following is an accepting run

$$(q_0, v_0) \xrightarrow{\delta' = \lceil c_{max} \rceil + 1.1} (q_0, v_0 + \delta') \xrightarrow{a} (q_F, v'_F)$$

Hence $(a; \lceil c_{max} \rceil + 1.1) \in \mathcal{L}(A) \neq L_6$

Therefore **no timed automaton** can accept L_6 \square