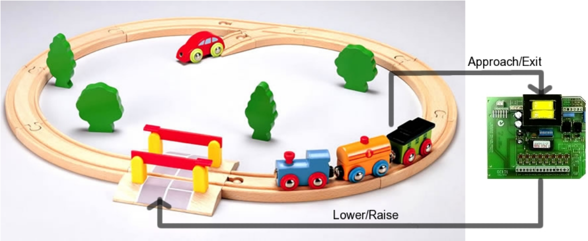
Topics in Timed Automata

B. Srivathsan

RWTH-Aachen

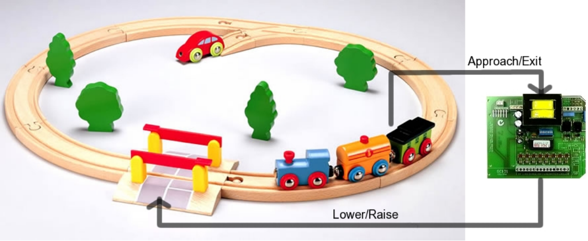
Software modeling and Verification group



Timed Automata

A theory of timed automata

R. Alur and D. Dill, *TCS'94*



Timed Automata

Language theory

Lectures 1 , . . . , 5

Verification

Lectures 6 , . . . , 10

A theory of timed automata

R. Alur and D. Dill, *TCS'94*

Lecture 1:

Timed languages and timed automata

Σ : alphabet $\{a, b\}$

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

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

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

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

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

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

Σ : alphabet $\{a, b\}$

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

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

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

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

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

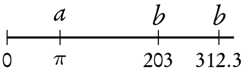
$\{aa, bb, ab, ba, abab, aaaa, \ldots\}$

Finite automata, pushdown automata, Turing machines, …

Σ : alphabet $\quad \{a, b\}$

$T\Sigma^*$: 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 ← w, τ → Time sequence

$w = a_1 \ldots a_n$

$a_i \in \Sigma$

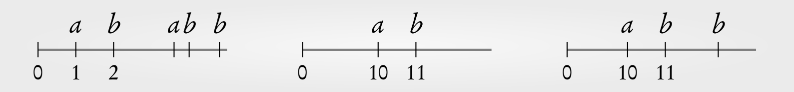
$\tau = \tau_1 \ldots \tau_n$

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

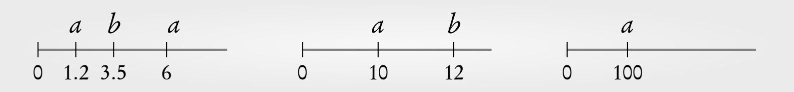
$\tau_1 \leq \cdots \leq \tau_n$

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

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

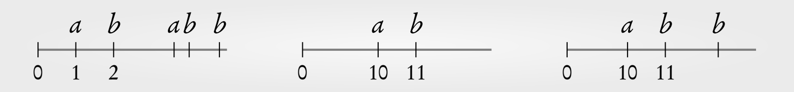


$$L_2 := \{\,(w, \tau) \mid \tau_{i+1} - \tau_i \geq 2 \text{ for all } i < |w|\}$$

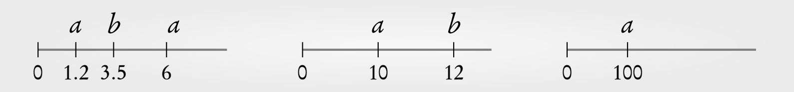


$L \subseteq T\Sigma^*$: Timed language $\longrightarrow$ *property over timed words*

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



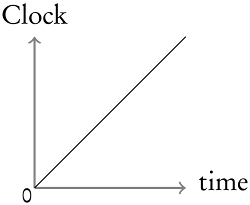
$$L_2 := \{\ (w, \tau) \mid \tau_{i+1} - \tau_i \geq 2 \text{ for all } i < |w|\}$$



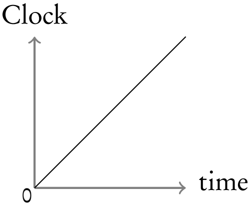
Timed automata

Timed automata

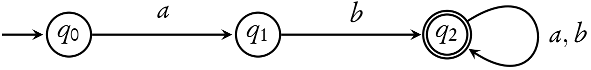
Timed automaton: Finite automaton + Finite no. of *Clocks*



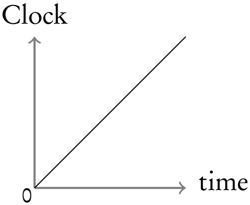
Timed automaton: Finite automaton + Finite no. of *Clocks*



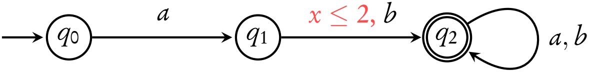
$$\{(\, ab(a+b)^*, \tau) \mid \tau_2 \leq 2\}$$



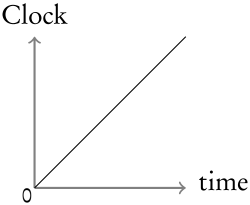
Timed automaton: Finite automaton + Finite no. of *Clocks*



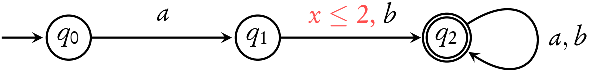
$$\{(\, ab(a+b)^*, \tau) \mid \tau_2 \leq 2\}$$

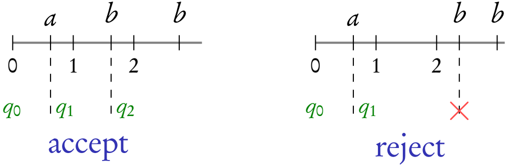


Timed automaton: Finite automaton + Finite no. of *Clocks*

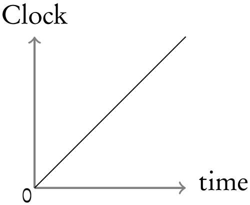


$$\{(\, ab(a+b)^*, \tau) \mid \tau_2 \leq 2\}$$





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

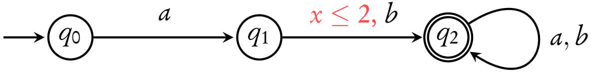


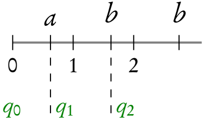
Guards

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

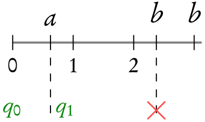
$$x \in \mathit{Clocks}\ , \quad c \in \mathbb{Q}_{\geq 0}$$

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



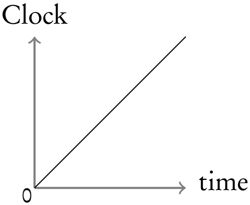


accept



reject

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

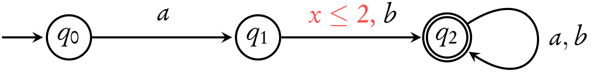


Guards

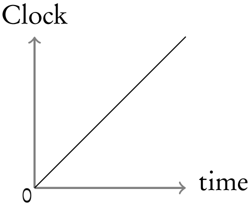
$$\phi := x \leq c \mid x \geq c \mid \neg\phi \mid \phi \wedge \phi$$

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

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



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



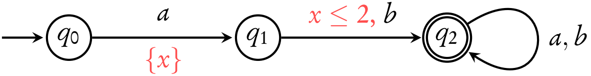
Guards

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

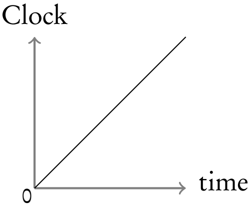
$$x \in Clocks\ ,\ \ c \in \mathbb{Q}_{\geq 0}$$

Resets

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



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



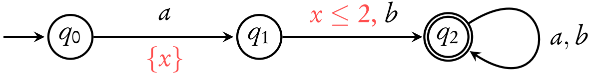
Guards

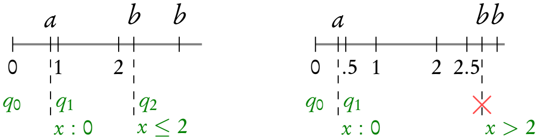
$$\phi := x \leq c \mid x \geq c \mid \neg\phi \mid \phi \wedge \phi$$

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

Resets

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

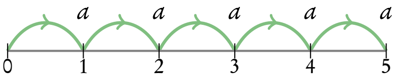




accept reject

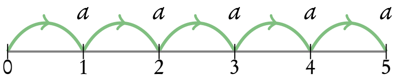
$$L_3 := \{\ (\ a^k, \tau\ ) \mid k > 0,\ \tau_i = i \text{ for all } i \leq k\}$$

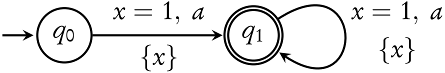
An "a" occurs in every integer from $1, \ldots, k$



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

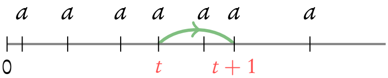
An "*a*" occurs in every integer from $1, \ldots, k$





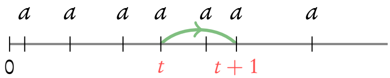
$$L_4 := \{\ (\ a^k, \tau\ ) \mid \text{exist } i, j \text{ s.t. } \tau_j - \tau_i = 1\}$$

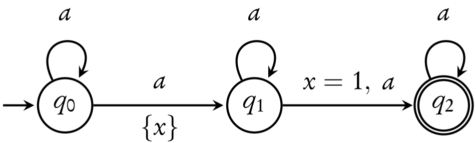
There are 2 "a"s which are at distance 1 apart



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

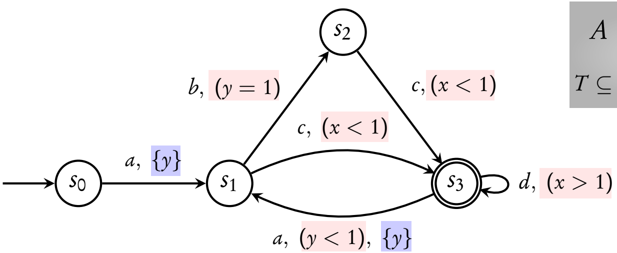
There are 2 "a"s which are at distance 1 apart





Three **mechanisms** to exploit:

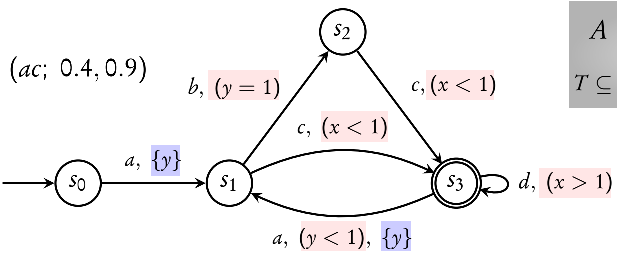
- Reset: to **start** measuring time
- Guard: to **impose** time constraint on action
- Non-determinism: for **existential** time constraints



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

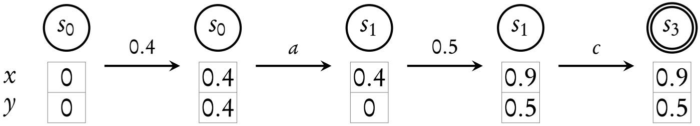
$$T \subseteq Q \times \Sigma \times \text{guard} \times \text{reset} \times Q$$

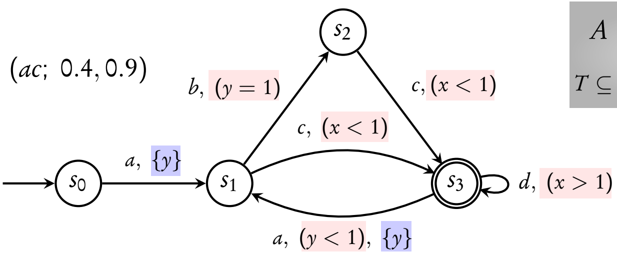
$(ac;\ 0.4, 0.9)$



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

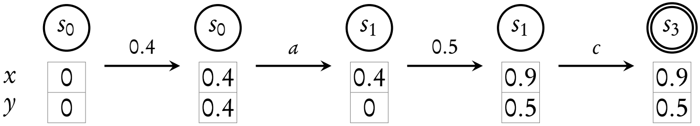
$$T \subseteq Q \times \Sigma \times \text{guard} \times \text{reset} \times Q$$





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

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



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

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

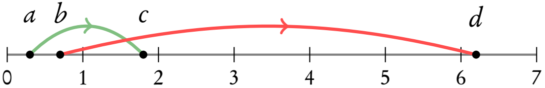
$(w, \tau) \in \mathcal{L}(A)$ if A has an **accepting** run over (w, τ)

Timed automata

Runs

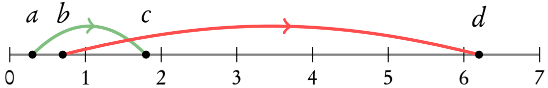
$$L_5 := \{\ (\ abcd.\Sigma^*, \tau\ )\ |\ \tau_3 - \tau_1 \leq 2 \text{ and } \tau_4 - \tau_2 \geq 5\}$$

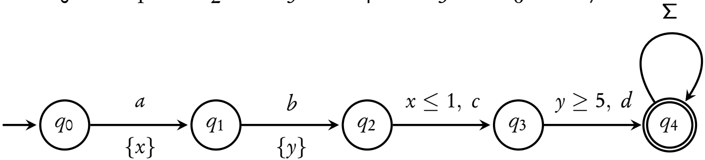
Interleaving distances



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

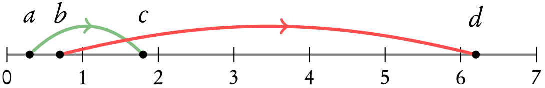
Interleaving distances

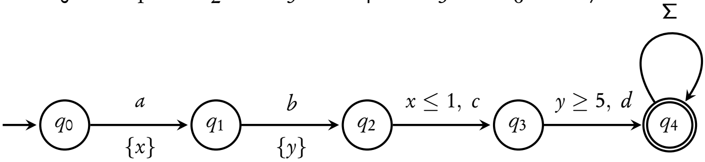




$$L_5 := \{ \, (\, abcd.\Sigma^*, \tau \,) \mid \tau_3 - \tau_1 \leq 2 \text{ and } \tau_4 - \tau_2 \geq 5 \}$$

Interleaving distances





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

n interleavings $\Rightarrow$ need n clocks

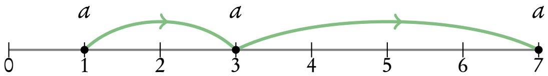
$n + 1$ clocks more expressive than n clocks

Timed automata

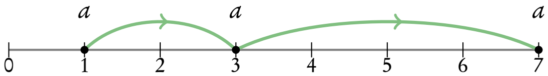
Runs

1 clock < 2 clocks < …

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



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



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,

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

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

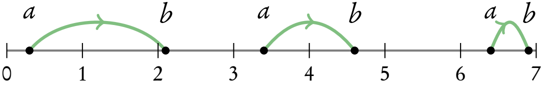
$$(q_0, v_0) \xrightarrow{\delta' = \lceil c_{max} \rceil + 1.1} (q_0, v_0 + \delta') \xrightarrow{a} (q_F, v'_F)$$

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

Therefore **no timed automaton** can accept L_6 □

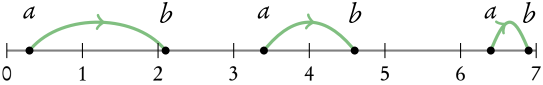
$$L_7 = \{\ (\ (ab)^k, \tau\ )\mid \tau_{2i+2} - \tau_{2i+1} < \tau_{2i} - \tau_{2i-1} \text{ for each } i \geq 1\}$$

Converging *ab* distances



$$L_7 = \{ \; (\; (ab)^k, \tau \;) \mid \tau_{2i+2} - \tau_{2i+1} < \tau_{2i} - \tau_{2i-1} \text{ for each } i \geq 1\}$$

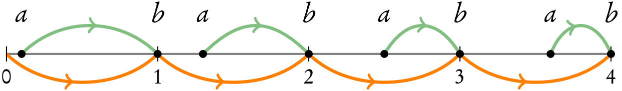
Converging *ab* distances



Exercise: Prove that **no timed automaton** can accept L_7

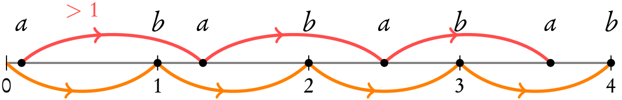
$$L_7 = \{ \; (\; (ab)^k, \tau \;) \mid \tau_{2i} = i \; \text{ and } \; \tau_{2i+2} - \tau_{2i+1} < \tau_{2i} - \tau_{2i-1} \}$$

Pivoted converging *ab* distances



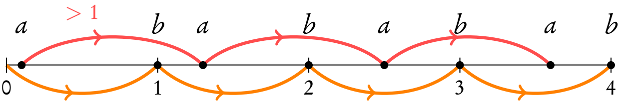
$$L_7 = \{ \; (\; (ab)^k, \tau \;) \mid \tau_{2i} = i \; \text{ and } \; \tau_{2i+2} - \tau_{2i+1} < \tau_{2i} - \tau_{2i-1} \}$$

Pivoted converging *ab* distances



$$L_7 = \{ \ (\ (ab)^k, \tau \ ) \mid \tau_{2i} = i \ \text{ and } \ \tau_{2i+2} - \tau_{2i+1} < \tau_{2i} - \tau_{2i-1} \}$$

Pivoted converging *ab* distances

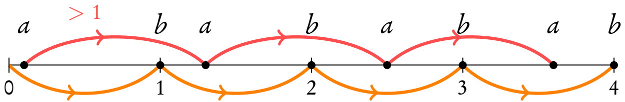


$$\tau_{2i+2} - \tau_{2i+1} < \tau_{2i} - \tau_{2i-1} \iff \tau_{2i+2} - \tau_{2i} < \tau_{2i+1} - \tau_{2i-1}$$

$$\iff 1 < \tau_{2i+1} - \tau_{2i-1}$$

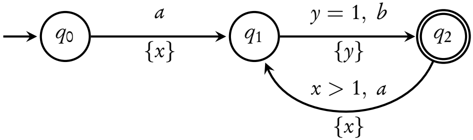
$$L_7 = \{ \, (\, (ab)^k, \tau \,) \mid \tau_{2i} = i \ \text{ and } \ \tau_{2i+2} - \tau_{2i+1} < \tau_{2i} - \tau_{2i-1} \}$$

Pivoted converging *ab* distances



$$\tau_{2i+2} - \tau_{2i+1} < \tau_{2i} - \tau_{2i-1} \Leftrightarrow \tau_{2i+2} - \tau_{2i} < \tau_{2i+1} - \tau_{2i-1}$$

$$\Leftrightarrow 1 < \tau_{2i+1} - \tau_{2i-1}$$



Timed automata

Runs

1 clock $<$ 2 clocks $< \ldots$

Role of max constant

Timed automata

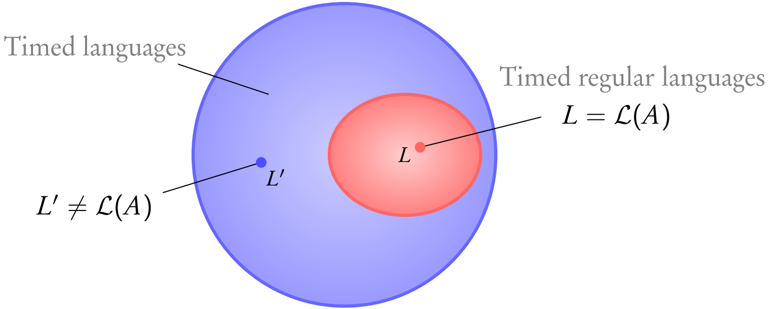
Runs

1 clock $<$ 2 clocks $< \ldots$

Role of max constant

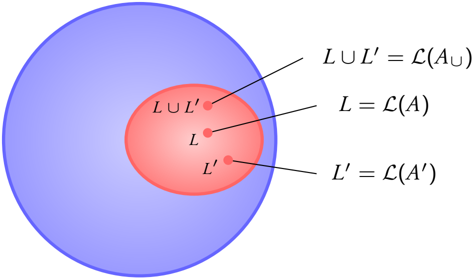
Timed regular lngs.

Timed regular languages



Definition

A timed language is called **timed regular** if it can be **accepted** by a timed automaton

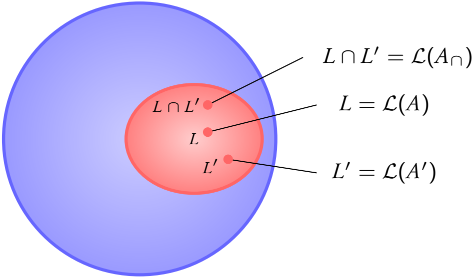


$A = (Q, \Sigma, X, T, Q_0, F)$ $\qquad$ $A' = (Q', \Sigma, X', T', Q'_0, F')$

$$A_\cup = (\ Q \cup Q'\ ,\ \Sigma\ ,\ X \cup X'\ ,\ T \cup T'\ ,\ Q_0 \cup Q'_0\ ,\ F \cup F'\ )$$

$$\mathcal{L}(A) \cup \mathcal{L}(A') = \mathcal{L}(A_\cup)$$

Timed regular languages are **closed** under **union**



$$A = (Q, \Sigma, X, T, Q_0, F) \qquad A' = (Q', \Sigma, X', T', Q'_0, F')$$

$$A_\cap = (\ Q \times Q'\ ,\ \Sigma\ ,\ X \cup X'\ ,\ T_\cap\ ,\ Q_0 \times Q'_0\ ,\ F \times F'\ )$$

$$T_\cap : \quad (q_1, q'_1) \xrightarrow[R \cup R']{a,\ g \wedge g'} (q_2, q'_2) \text{ if}$$

$$q_1 \xrightarrow[R]{a,\ g} q_2 \in T \quad \text{and} \quad q'_1 \xrightarrow[R']{a,\ g'} q'_2 \in T'$$

Timed regular languages are **closed** under **intersection**

L : a timed language over Σ

$$\text{Untime}(L) \equiv \{w \in \Sigma^* \mid \exists \tau.\ (w, \tau) \in L\}$$

Untiming construction

For **every timed** automaton A there is a **finite automaton** A_u s.t.

$$\text{Untime}(\ \mathcal{L}(A)\ ) = \mathcal{L}(A_u)$$

more about this in Lecture 6 . . .

Complementation

$$\Sigma : \{a, b\}$$

$$L = \{ (w, \tau) \mid \text{there is an } a \text{ at some time } t \text{ and no action occurs at time } t+1 \}$$

$$\overline{L} = \{ (w, \tau) \mid \text{every } a \text{ has an action at a distance 1 from it} \}$$

Complementation

$$\Sigma : \{a, b\}$$

$$L = \{ (w, \tau) \mid \text{there is an } a \text{ at some time } t \text{ and no action occurs at time } t+1 \}$$

$$\overline{L} = \{ (w, \tau) \mid \text{every } a \text{ has an action at a distance 1 from it} \}$$

Claim: **No timed automaton** can accept $\overline{L}$

Decision problems for timed automata: A survey

Alur, Madhusudhan. *SFM'04: RT*

Step 1: $\overline{L} \;=\; \{\, (w, \tau) \mid$ every *a* has an action at a distance 1 from it $\}$

Suppose $\overline{L}$ is timed regular

Step 1: $\overline{L} = \{ (w, \tau) \mid$ every a has an action at a distance 1 from it $\}$

Suppose $\overline{L}$ is timed regular

Step 2: Let $L' = \{ (a^* b^*, \tau) \mid$ all a's occur before time 1 and no two a's happen at same time $\}$

Clearly L' is timed regular

Step 1: $\overline{L} \;=\; \{\, (w, \tau) \mid$ every a has an action at a distance 1 from it $\}$

Suppose $\overline{L}$ is timed regular

Step 2: Let $L' = \{\, (a^* b^*, \tau) \mid$ all a's occur before time 1 and no two a's happen at same time $\}$

Clearly L' is timed regular

Step 3: Untime($\overline{L} \cap L'$) should be a regular language

Step 1: $\overline{L} \;=\; \{\,(w,\tau) \mid$ every a has an action at a distance 1 from it $\}$

Suppose $\overline{L}$ is timed regular

Step 2: Let $L' = \{\,(a^*b^*,\tau) \mid$ all a's occur before time 1 and no two a's happen at same time $\}$

Clearly L' is timed regular

Step 3: Untime($\overline{L} \cap L'$) should be a regular language

Step 4: But, Untime($\overline{L} \cap L'$) $= \{a^n b^m \mid m \geq n\}$, ***not regular!***

Step 1: $\overline{L} = \{ (w, \tau) \mid$ every a has an action at a distance 1 from it $\}$

Suppose $\overline{L}$ is timed regular

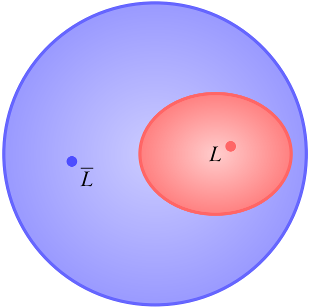
Step 2: Let $L' = \{ (a^* b^*, \tau) \mid$ all a's occur before time 1 and no two a's happen at same time $\}$

Clearly L' is timed regular

Step 3: Untime($\overline{L} \cap L'$) should be a regular language

Step 4: But, Untime($\overline{L} \cap L'$) $= \{a^n b^m \mid m \geq n\}$, *not regular!*

Therefore $\overline{L}$ **cannot be timed regular** □



Timed regular languages are **not closed** under **complementation**

Timed automata

Runs

1 clock $<$ 2 clocks $< \ldots$

Role of max constant

Timed regular lngs.

Closure under $\cup$, $\cap$

Non-closure under complement

Timed automata

Runs

1 clock $<$ 2 clocks $< \ldots$

Role of max constant

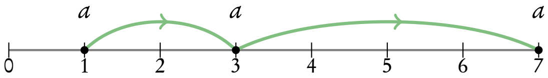
Timed regular lngs.

Closure under $\cup$, $\cap$

Non-closure under complement

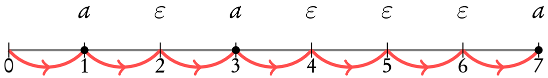
ε-transitions

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

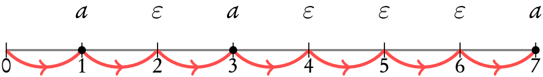


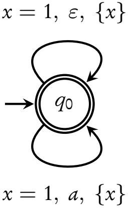
Claim: **No timed automaton** can accept L_6

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



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





ε-transitions

ε-transitions **add expressive power** to timed automata.

Characterization of the expressive power of silent transitions in timed automata

Bérard, Diekert, Gastin, Petit. *Fundamenta Informaticae'98*

ε-transitions

ε-transitions **add expressive power** to timed automata. However, they add power **only** when a clock is **reset** in an ε-transition.

Characterization of the expressive power of silent transitions in timed automata

Bérard, Diekert, Gastin, Petit. *Fundamenta Informaticae'98*

Timed automata

Runs

1 clock $<$ 2 clocks $< \dots$

Role of max constant

Timed regular lngs.

Closure under $\cup$, $\cap$

Non-closure under complement

ε-transitions

More expressive

$\xrightarrow{\varepsilon}$ without reset $\equiv$ TA

Timed automata

Runs

1 clock $<$ 2 clocks $< \dots$

Role of max constant

Timed regular lngs.

Closure under $\cup, \cap$

Non-closure under complement

ε-transitions

More expressive

$\xrightarrow{\varepsilon}$ without reset $\equiv$ TA

Next lecture: **Determinization**