TIMED AUTOMATA

LECTURE 1

PLAN FOR TODAY

- Timed languages
- Timed automata
- closure properties
- Motivation for the model.

Timed Languages

Σ : alphabet $\{a, b\}$

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

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

$L_1 :=$ {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 $\{a, b\}$

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 "}

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

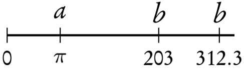
$L_2 :=$ {set of words with a non-zero even length }

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

Finite automata, pushdown automata, Turing machines, …

Σ : alphabet $\{a, b\}$

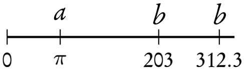
$T\Sigma^*$: timed words



$(aa;\ 0.8, 2.5)$ $(abb;\ \pi, 203, 312.3)$

Σ : alphabet $\{a, b\}$

$T\Sigma^*$: timed words



$(aa;\ 0.8, 2.5)$ $(abb;\ \pi, 203, 312.3)$

(w, τ)

Word

Time sequence

$w = a_1 \ldots a_n$

$a_i \in \Sigma$

$\tau = \tau_1 \ldots \tau_n$

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

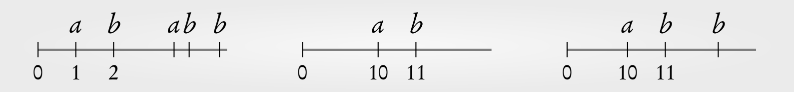
$\tau_1 \leq \cdots \leq \tau_n$

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

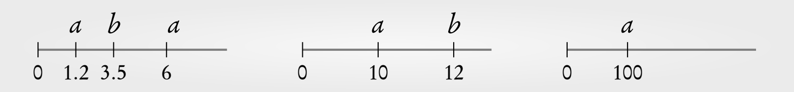
a b a b

100 102

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

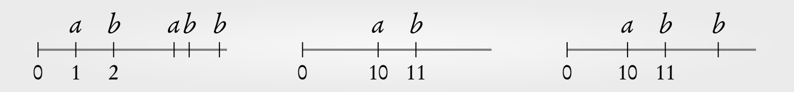


$$L_2 := \{ (w, \tau) \mid \tau_{i+1} - \tau_i \geq 2 \text{ for all } i < |w|\}$$

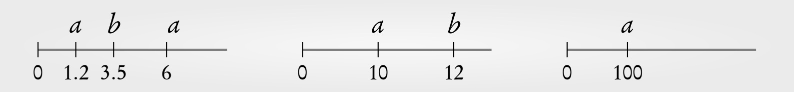


$L \subseteq T\Sigma^*$: Timed language $\longrightarrow$ *property over timed words*

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



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



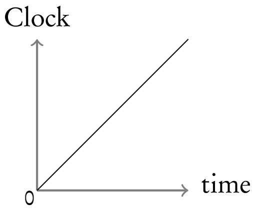
Timed automata

PLAN FOR TODAY

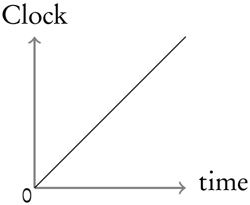
- Timed languages ✓
- Timed automata
- closure properties
- Motivation for the model.

TIMED AUTOMATA

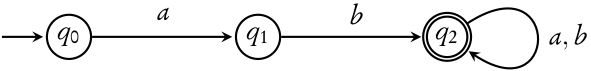
Timed automaton: Finite automaton + Finite no. of *Clocks*



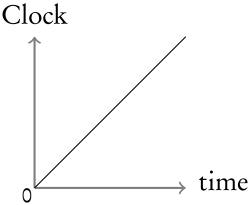
Timed automaton: Finite automaton + Finite no. of *Clocks*



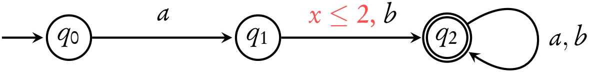
$$\{(\,ab(a+b)^*, \tau)\mid \tau_2 \leq 2\}$$



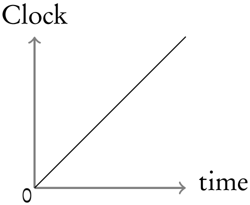
Timed automaton: Finite automaton + Finite no. of *Clocks*



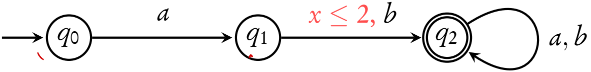
$$\{(\, ab(a+b)^*, \tau) \mid \tau_2 \leq 2\}$$

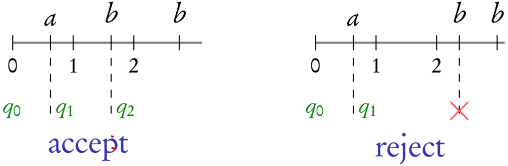


Timed automaton: Finite automaton + Finite no. of *Clocks*

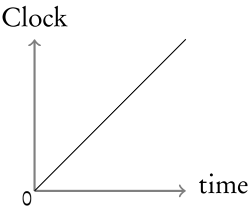


$$\{(\,ab(a+b)^*, \tau) \mid \tau_2 \leq 2\}$$





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

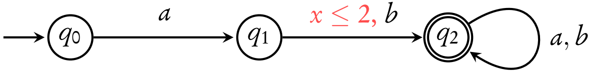


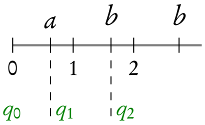
Guards

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

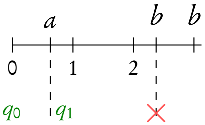
$$x \in Clocks\ , \ \ c \in \cancel{\mathbb{Q}_{\geq 0}}\ \mathbb{N}$$ (natural no.s)

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



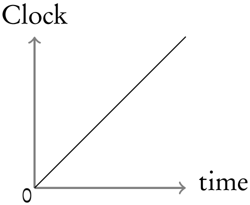


accept



reject

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

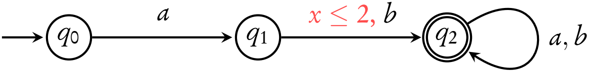


Guards

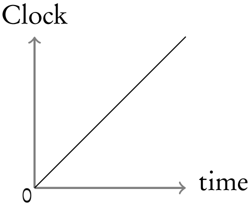
$$\phi := x \leq c \mid x \geq c \mid \neg\phi \mid \phi \wedge \phi$$

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

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



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



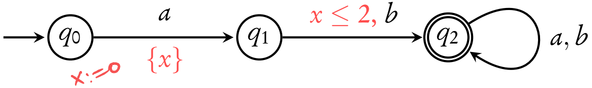
Guards

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

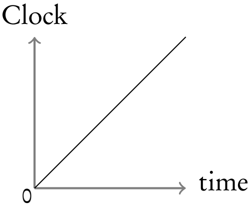
$$x \in \mathit{Clocks}\ ,\ \ c \in \mathbb{Q}_{\geq 0}$$

Resets

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



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



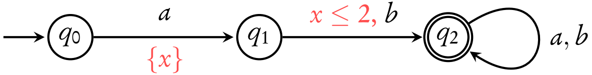
Guards

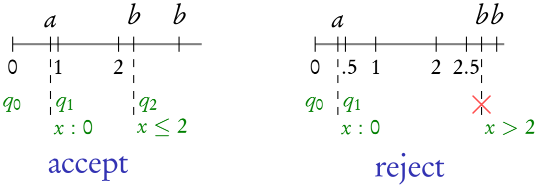
$$\phi := x \leq c \mid x \geq c \mid \neg\phi \mid \phi \wedge \phi$$

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

Resets

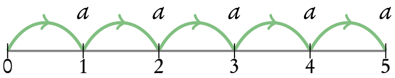
$$\{(\, ab(a+b)^*, \tau) \mid \tau_2 - \tau_1 \leq 2\}$$

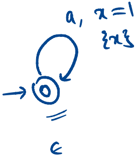




$$L_3 := \{ (a^k, \tau) \mid k > 0,\ \tau_i = i \ \text{ for all } i \le k\}$$

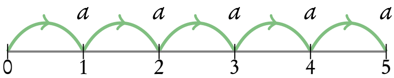
An "a" occurs in every integer from $1, \ldots, k$

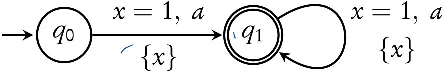




$$L_3 := \{\ (\ a^k, \tau\ ) \mid k > 0,\ \tau_i = i \ \text{ for all } i \leq k\}$$

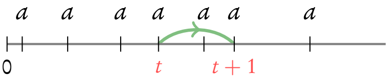
An "a" occurs in every integer from $1, \ldots, k$





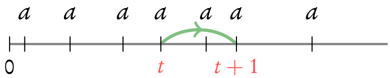
$$L_4 := \{\ (\ a^k, \tau\ )\ |\ \text{exist } i, j \text{ s.t. } \tau_j - \tau_i = 1\}$$

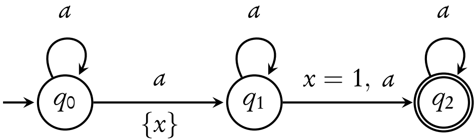
There are 2 "a"s which are at distance 1 apart



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

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



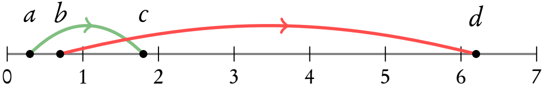


Three **mechanisms** to exploit:

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

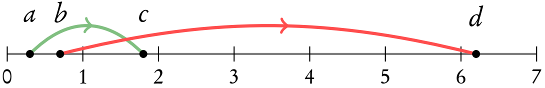
$$L_5 := \{ \; (\; abcd.\Sigma^*, \tau \;) \mid \tau_3 - \tau_1 \leq 2 \text{ and } \tau_4 - \tau_2 \geq 5 \}$$

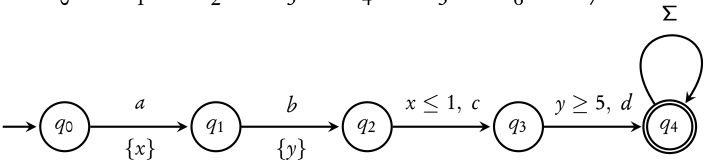
Interleaving distances



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

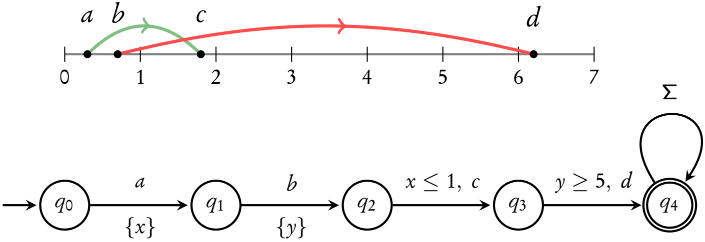
Interleaving distances





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

Interleaving distances

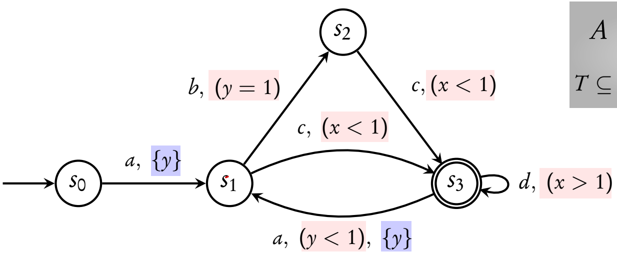


Exercise: Prove that L_5 cannot be accepted by a one-clock TA.

n interleavings $\Rightarrow$ need n clocks

$n+1$ clocks more expressive than n clocks

SYNTAX:



states alphabet clocks Transition initial stat. acc stat.

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

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

Semantics of a timed automaton:

$\mathcal{A}$: Timed automaton.

What is the timed language of $\mathcal{A}$?

↳ When does a timed automaton accept a timed word ?

$$\begin{matrix} a_1 & a_2 & a_3 & \cdots & a_k \\ \tau_1 & \tau_2 & \tau_3 & & \tau_k \end{matrix}$$

Configurations:

$$(q, v)$$

state

valuation

Valuation: $X \longmapsto \mathbb{R}_{\geq 0}$

↳ non-negative real

Transitions on configurations:

$$(q, v) \xrightarrow[\text{delay}]{\delta} (q, v+\delta)$$

↳ $v(x) + \delta \quad \forall x$

v: $x = 2.5$ $\quad \delta = 2$

$y = 1.5$

$v+\delta$: $x = 4.5$

$y = 3.5$

$$(q, v) \xrightarrow{a} (q_1, v_1) \qquad \exists \quad q \xrightarrow[R]{g,\, a} q_1$$

if $\quad v \models g \quad$ (v satisfies g)

and $\quad v_1 = [R]\, v$

$[R]\, v$ is a valuation s.t.

$$[R]\, v(x) = \begin{cases} 0 & \text{if } x \in R \\ v(x) & \text{otherwise} \end{cases}$$

$$v: \begin{array}{l} x = 5 \\ y = 2 \end{array} \xrightarrow[R = \{x\}]{} v_1: \begin{array}{l} x = 0 \\ y = 2 \end{array}$$

Runs of a T.A:

Timed word: $\quad \begin{array}{ccccc} a_1 & a_2 & \cdots & a_k \\ \tau_1 & \tau_2 & & \tau_k \end{array}$

$$(q_0, v_0) \xrightarrow{\tau_1} (q_0, v_0 + \tau_1) \xrightarrow{a_1} (q_1, v_1) \xrightarrow{\tau_2 - \tau_1} (q_1, v_1 + (\tau_2 - \tau_1))$$

$\downarrow a_2$

$\vdots$

$v_0(x) = 0 \quad \forall x \in$ Clocks

if $\exists\ (q_0, a_1, g_1, R_1, q_1)$

s.t. $v_0 + \tau_1 \models g_1$

$v_1 = [R]\, v_0$

$q_0 \in Q_0$

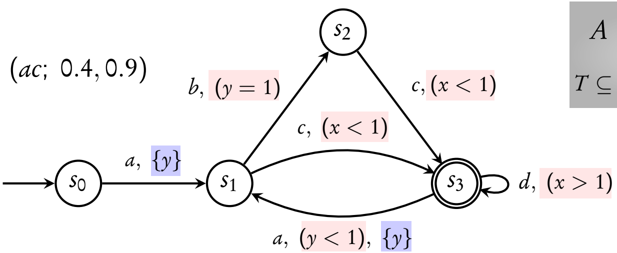
Accepting run:

A run is accepting if it ends in an accepting state.

Language of a T.A.:

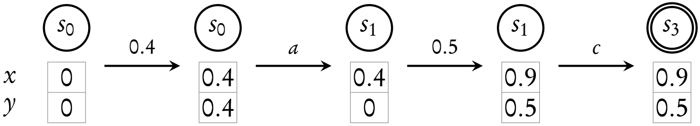
$$\mathcal{L}(\mathcal{A}) = \{ \underbrace{(w, \tau)}_{\text{Timed words}} \mid \text{there exists an accepting run of } \mathcal{A} \text{ on } (w, \tau) \}$$

$(ac;\ 0.4, 0.9)$



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

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



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

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

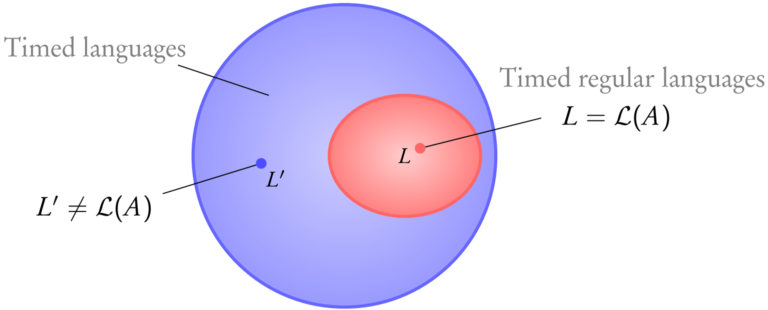
$(w, \tau) \in \mathcal{L}(A)$ if A has an **accepting** run over (w, τ)

PLAN FOR TODAY

- Timed languages ✓
- Timed automata ✓
- closure properties
- Motivation for the model.

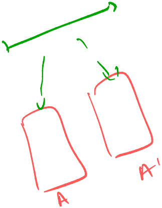
Closure Properties

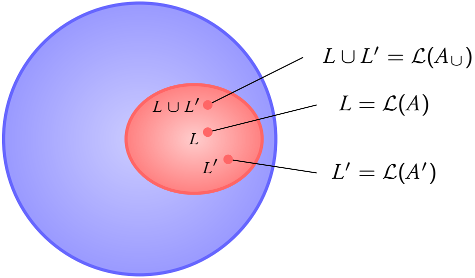
Timed regular languages



Definition

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



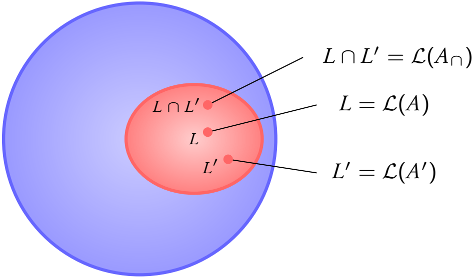


$$A = (Q, \Sigma, X, T, Q_0, F) \qquad\qquad A' = (Q', \Sigma, X', T', Q_0', F')$$

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

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

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



A

A'

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

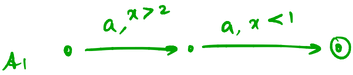
$$A_\cap = (\, Q \times Q' \,,\, \Sigma \,,\, X \cup X' \,,\, T_\cap \,,\, Q_0 \times Q_0' \,,\, F \times F' \,)$$

$q_1 \xrightarrow[R]{g,\, a} q_2$ $\qquad$ $q_1' \xrightarrow[R']{a,\, g'} q_2'$

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

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

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



$\mathcal{L}(A_1) = \{\}$

$\{aa\}$

L : a timed language over Σ

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

Untiming construction

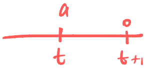
For **every timed** automaton A there is a **finite automaton** A_u s.t.

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

more about this later . . .

Complementation

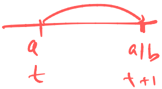
$$\Sigma : \{a, b\}$$



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

timed regular

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



Complementation

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

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

$\overline{L} \;=\; \{\, (w, \tau) \mid$ every a 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 $\}$

$L_1' = \{(a^* b^*, \tau) \mid$ all a's occur before 1 $\}$

Clearly L' is timed regular

$L_2' = \{(a^* b^*, \tau) \mid$ no two a's happen at same time $\}$

A_1' for L_1'

A_2' for L_2'

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

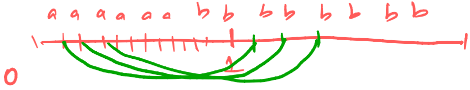
Suppose $\overline{L}$ is timed regular

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

$\overline{L} \cap L'$:



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

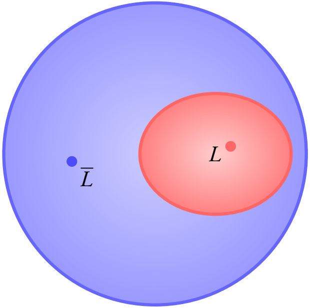
Step 2: Let $L' = \{\, (a^* b^*, \tau) \mid$ all a's occur before time 1 and no two a's happen at same time $\}$

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** $\square$



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

PLAN FOR TODAY

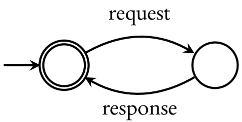
- Timed languages ✓
- Timed automata ✓
- closure properties ✓
- Motivation for the model.

MOTIVATION FOR THE MODEL

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

$$\begin{tikzcd} \text{System} \arrow[d] & \text{Property} \arrow[d] \\ \text{Automaton } \mathcal{A} & \text{Automaton } \mathcal{B} \end{tikzcd}$$

System $\downarrow$ Automaton $\mathcal{A}$

Property $\downarrow$ Automaton $\mathcal{B}$

Does system **satisfy** property?

System $\downarrow$ Automaton $\mathcal{A}$

Property $\downarrow$ Automaton $\mathcal{B}$

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

Does system **satisfy** property?

Model-checking

System

$\downarrow$

Automaton $\mathcal{A}$

Property

$\downarrow$

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 → Higher-level description $\xrightarrow{\text{translation}}$ Automaton $\mathcal{A}$

Property → Higher-level description $\xrightarrow{\text{translation}}$ Automaton $\mathcal{B}$

Model-Checker

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

In practice...

Huge system $\downarrow$ Higher-level description $\xrightarrow{\text{translation}}$ Automaton $\mathcal{A}$

Property $\downarrow$ Higher-level description $\xrightarrow{\text{translation}}$ Automaton $\mathcal{B}$

Model-Checker

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

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

In practice...

Huge system $\downarrow$ Higher-level description $\xrightarrow{\text{translation}}$ Automaton $\mathcal{A}$

Property $\downarrow$ Higher-level description $\xrightarrow{\text{translation}}$ Automaton $\mathcal{B}$

Model-Checker

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

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

SUMMARY

- Timed languages ✓
- Timed automata ✓
- closure properties ✓
- Motivation for the model. ✓