TIMED AUTOMATA

LECTURE 2

GOALS OF TODAY'S LECTURE

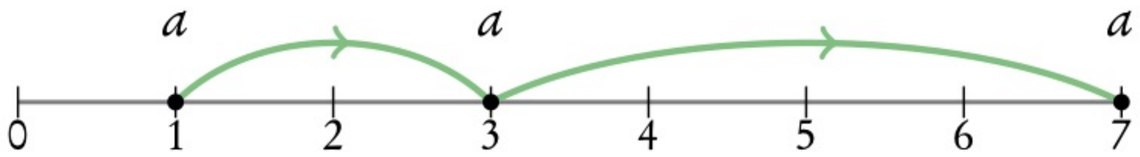
-1. Languages not accepted by Timed Automata

-2. Timed regular languages

-3. Closure properties

-4. Course plan

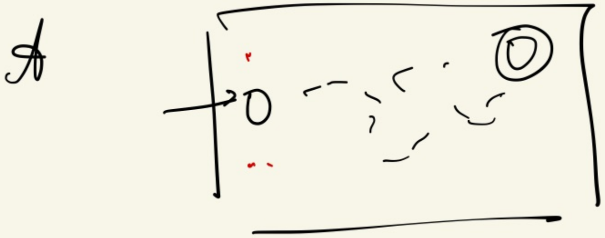
$L_6 := \{ \, (\, a^k, \tau \,) \mid \tau_i$ is some integer for each $i\}$



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

Proof idea:

Suppose there is a TA accepting L_b (we have only natural no.s in guards)



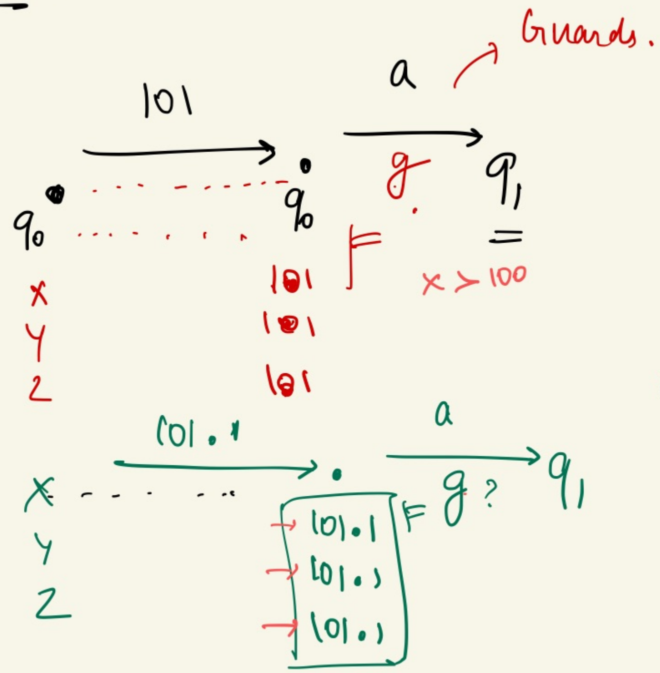
maximum constant

100

(a, 90) $x = 90$

(a, 90.1)

$\boxed{(a, 101) \in L_b}$



$\boxed{(a, 101.1)} \in L_b$

using the same accepting run.

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

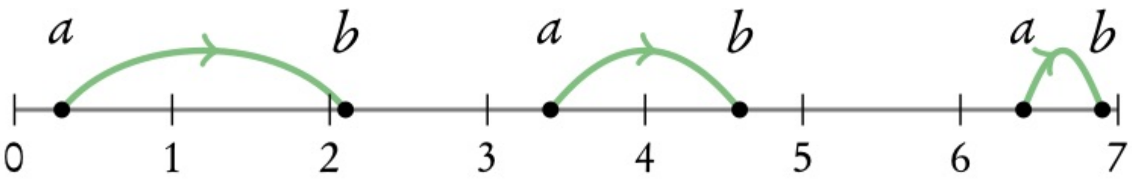
$$(q_0, v_0) \xrightarrow{\delta'\, =\, \lceil c_{max} \rceil + 1.1} (q_0, v_0 + \delta') \xrightarrow{a} (q_F, v'_F)$$

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

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

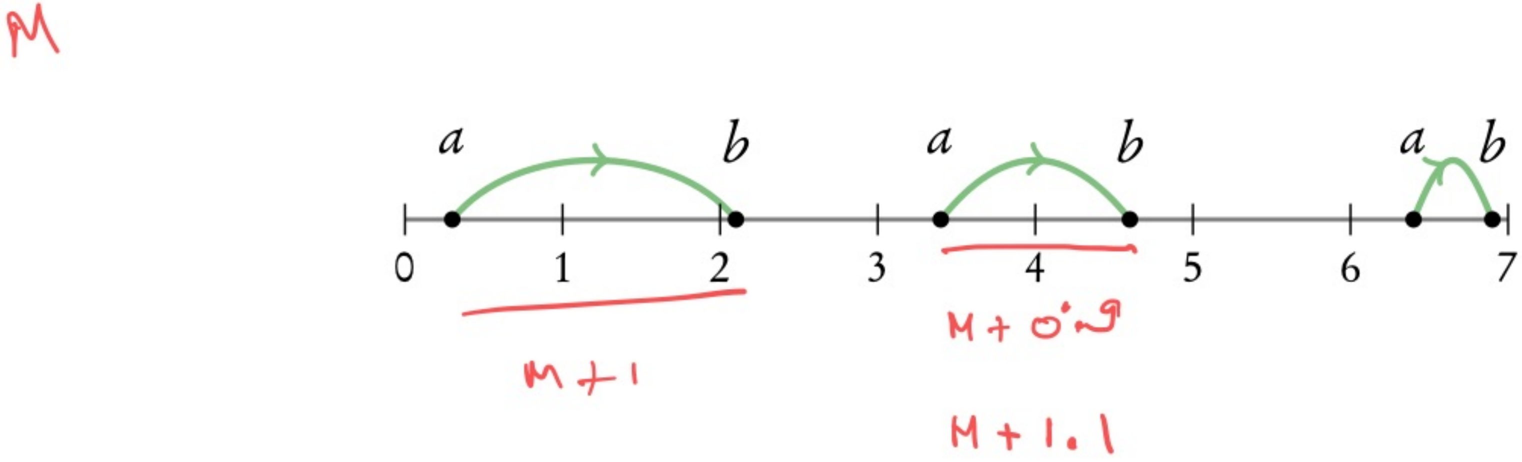
$$L_7 = \{ \ (\ (ab)^k, \tau \ ) \mid \tau_{2i+2} - \tau_{2i+1} < \tau_{2i} - \tau_{2i-1} \text{ for each } i \geq 1\}$$

Converging *ab* distances



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

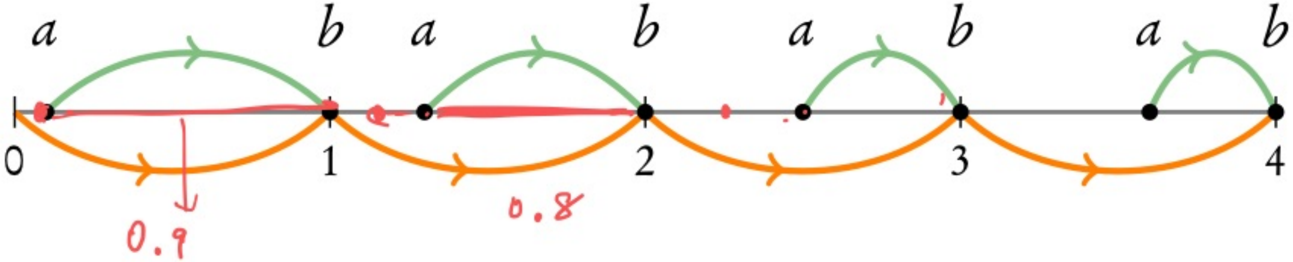
Converging *ab* distances



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

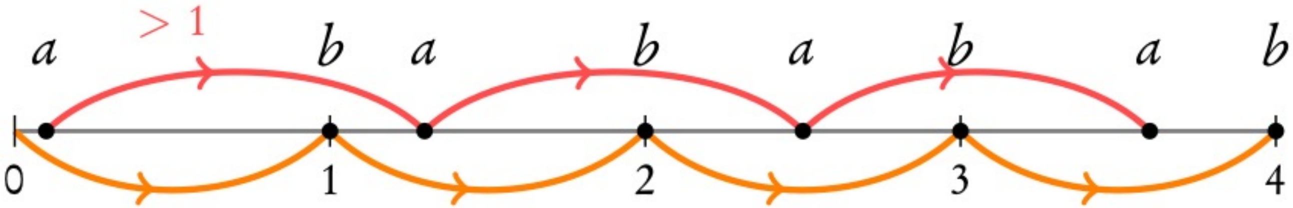
$$L_7 = \{\ (\ (ab)^k, \tau\ )\ |\ \tau_{2i} = i \ \text{ and }\ \tau_{2i+2} - \tau_{2i+1} < \tau_{2i} - \tau_{2i-1}\}$$

Pivoted converging *ab* distances



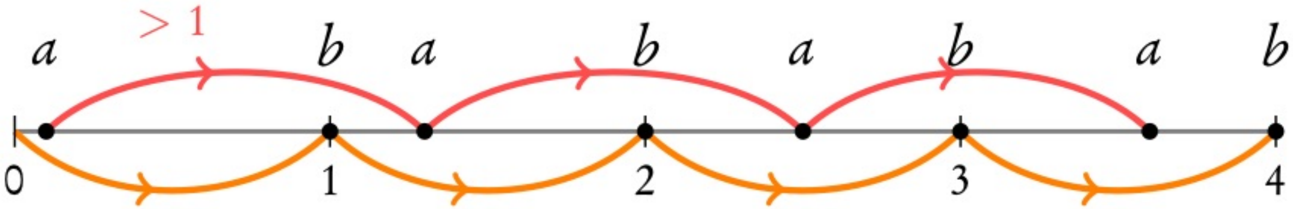
$$L_7 = \{ \; (\; (ab)^k, \tau \;) \mid \tau_{2i} = i \; \text{ and } \; \tau_{2i+2} - \tau_{2i+1} < \tau_{2i} - \tau_{2i-1} \}$$

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

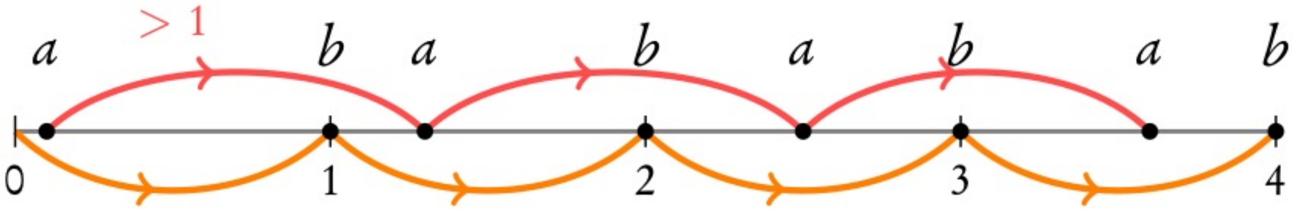


$$\tau_{2i+2} - \tau_{2i+1} < \tau_{2i} - \tau_{2i-1} \Leftrightarrow \tau_{2i+2} - \tau_{2i} < \tau_{2i+1} - \tau_{2i-1}$$

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

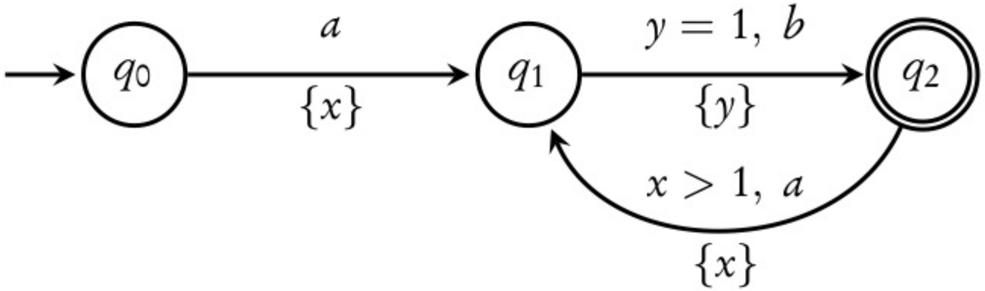
$$L_7 = \{\ (\ (ab)^k, \tau\ )\ |\ \tau_{2i} = i \ \text{ and } \ \tau_{2i+2} - \tau_{2i+1} < \tau_{2i} - \tau_{2i-1}\}$$

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

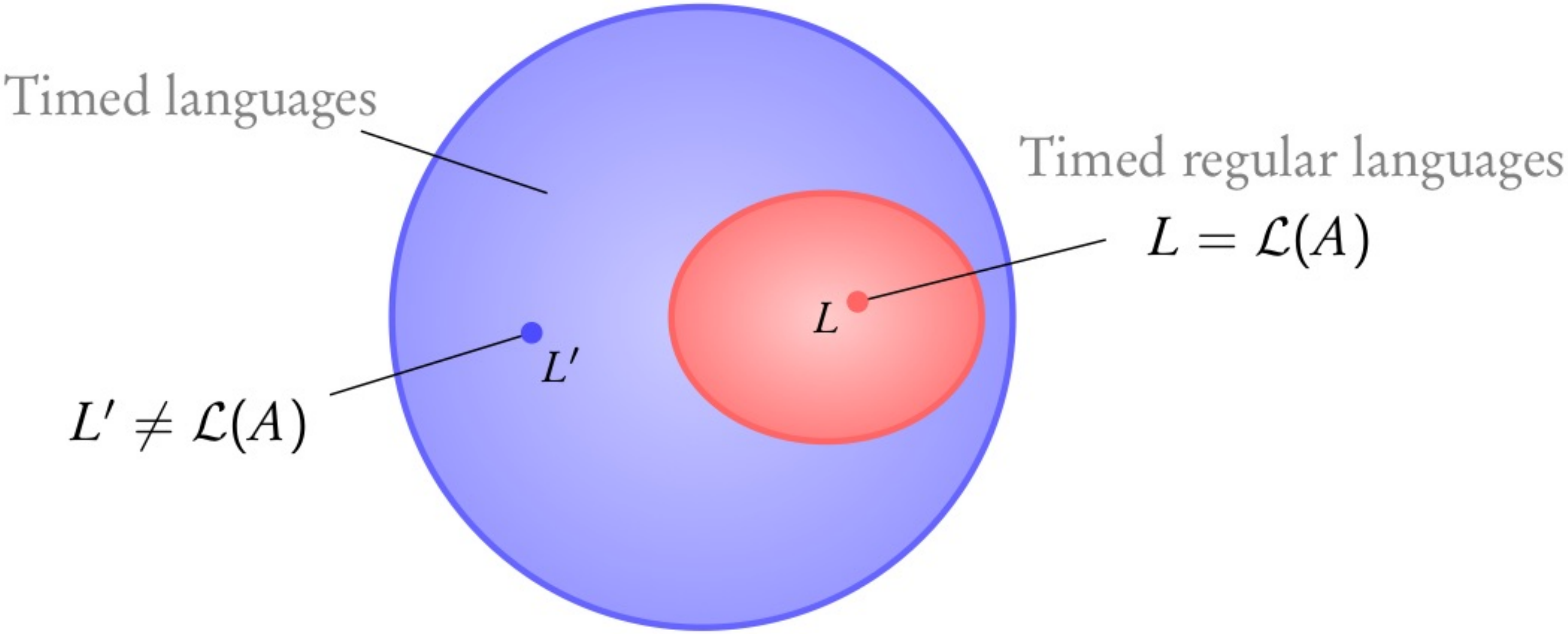
Runs

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

Role of max constant

Timed regular lngs.

Timed regular languages

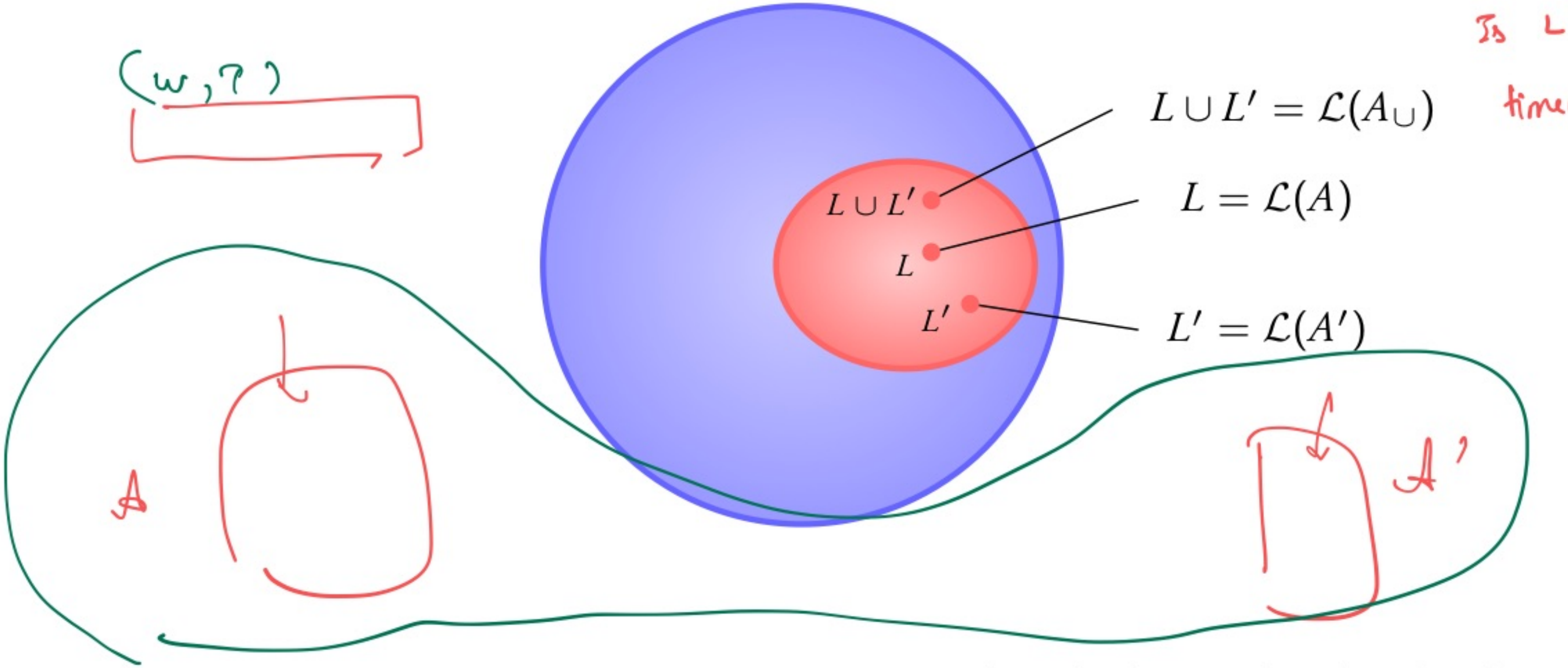


Definition

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

(w, ?)

Is $L \cap L'$ timed regular?

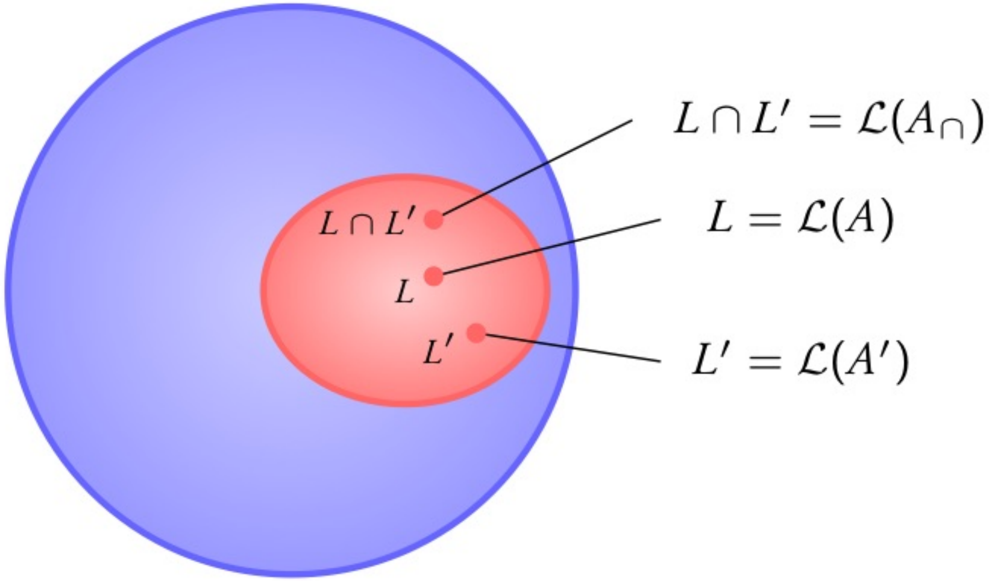


$A = (Q, \Sigma, X, T, Q_0, F)$ $A' = (Q', \Sigma, X', T', Q'_0, F')$

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

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

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



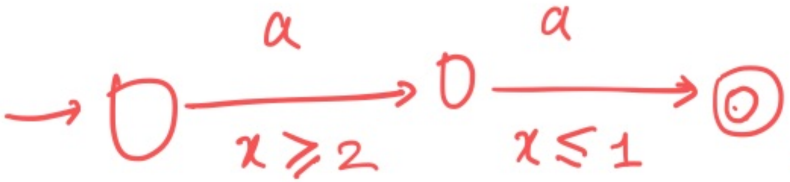
$$A = (Q, \Sigma, X, T, Q_0, F) \qquad A' = (Q', \Sigma, X', T', Q'_0, F')$$

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

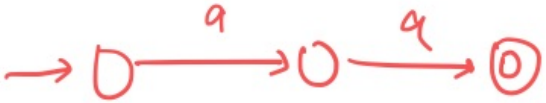
$$T_\cap :\quad (q_1, q'_1) \xrightarrow[R \cup R']{a,\ g \wedge g'} (q_2, q'_2) \text{ if}$$

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

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



Empty.



L : a timed language over Σ

aa

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

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 later . . .

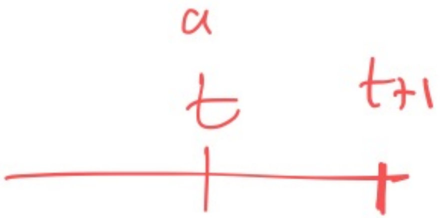
Closure under.

Complementation

a b ×

1 1

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

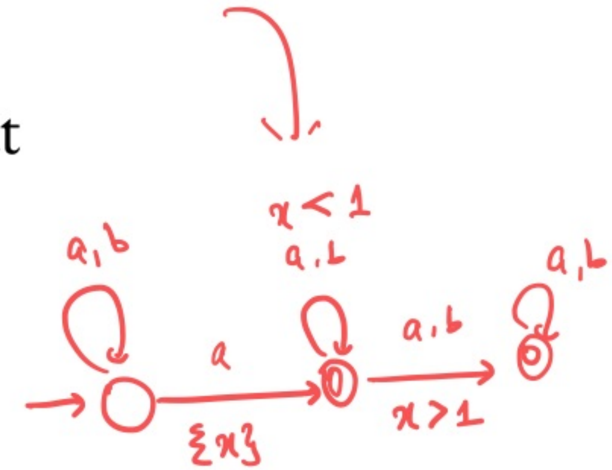


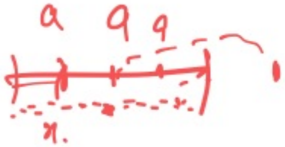
a b a b ✓

1 1 2 2.5

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

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





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*

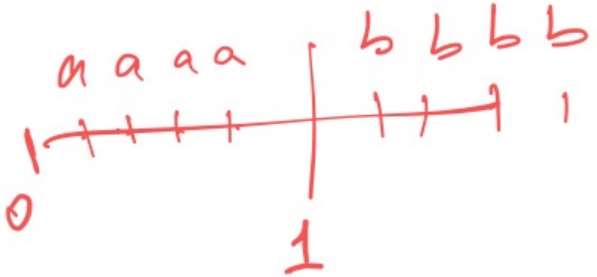
Step 1: $\overline{L} \;=\; \{\, (w, \tau) \mid$ every *a* has an action at a distance 1 from it $\}$

Suppose $\overline{L}$ is timed regular

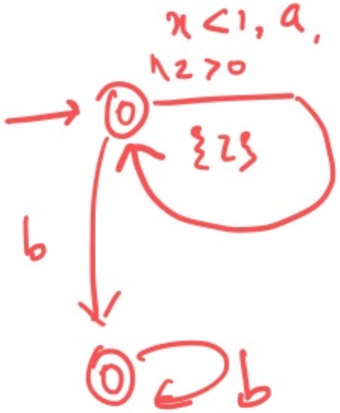
Step 1: $\overline{L} = \{ (w, \tau) \mid$ every a has an action at
a distance 1 from it $\}$

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 $\}$



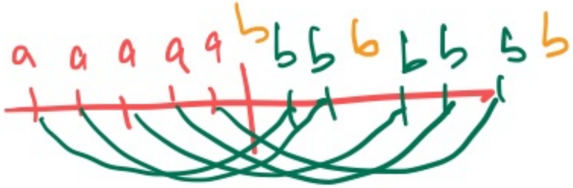
Clearly L' is timed regular



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

$\underline{L' \cap \overline{L}}$ should be 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

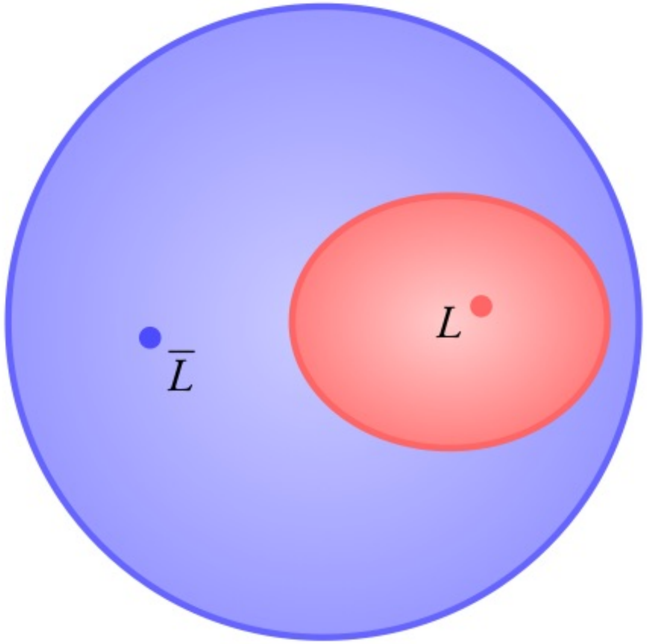
Step 2: Let $L' = \{\, (a^* b^*, \tau) \mid$ all a's occur before time 1 and no two a's happen at same time $\}$

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 $< \dots$

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

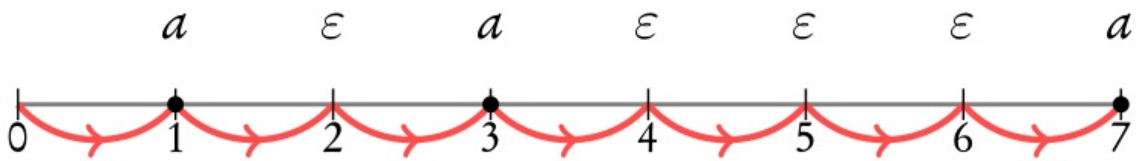
Timed regular lngs.

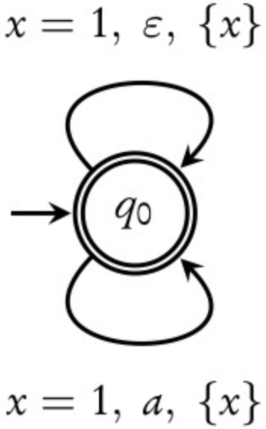
Closure under $\cup$, $\cap$

Non-closure under complement

ε-transitions

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





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 $\cup$, $\cap$

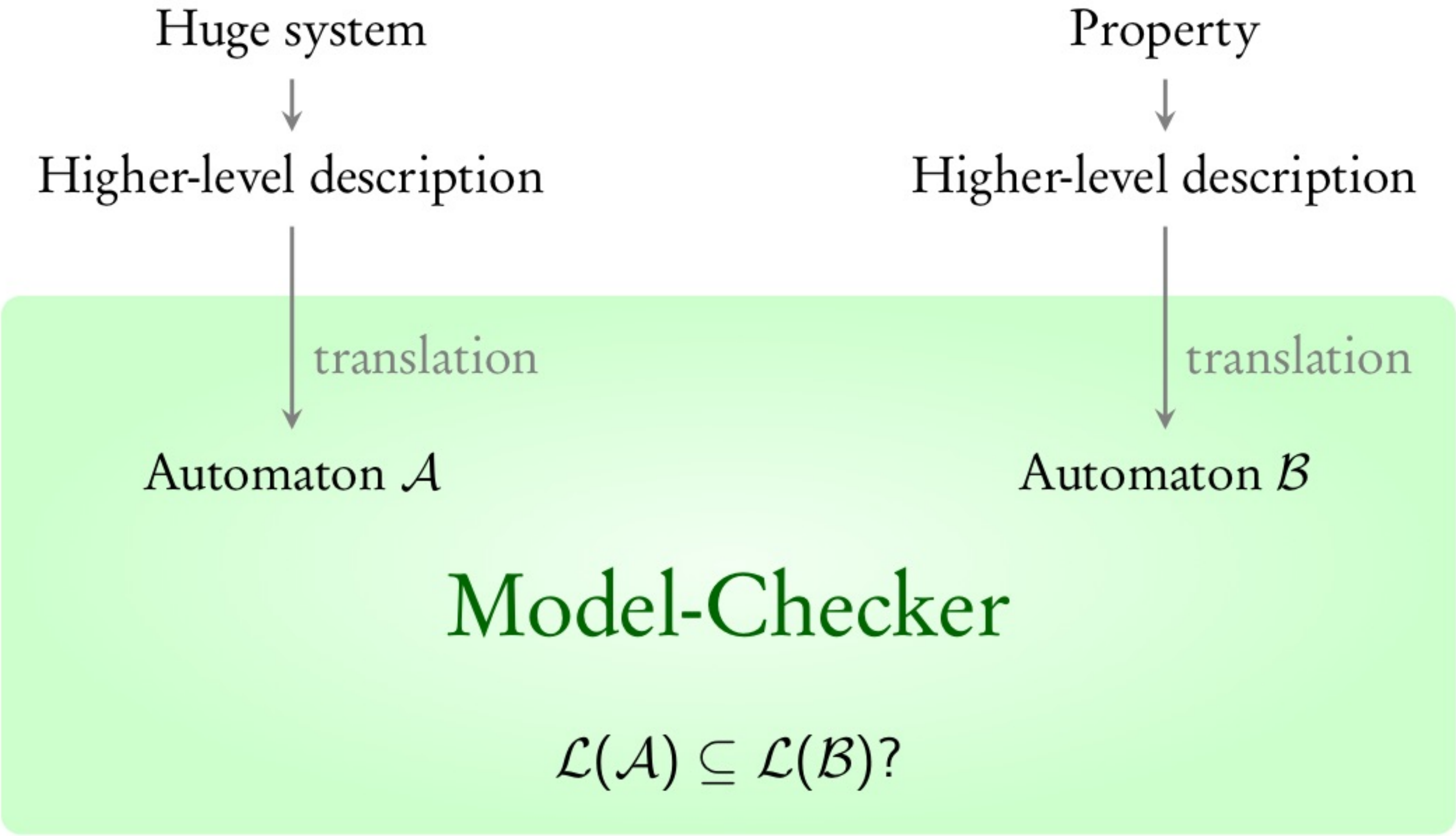
Non-closure under complement

ε-transitions

More expressive

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

Recall...



Huge system

Higher-level description

translation

Automaton $\mathcal{A}$

Property

Higher-level description

translation

Automaton $\mathcal{B}$

Model-Checker

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

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

iff

$$\mathcal{L}(\mathcal{A}) \cap \overline{\mathcal{L}(\mathcal{B})} = \emptyset$$

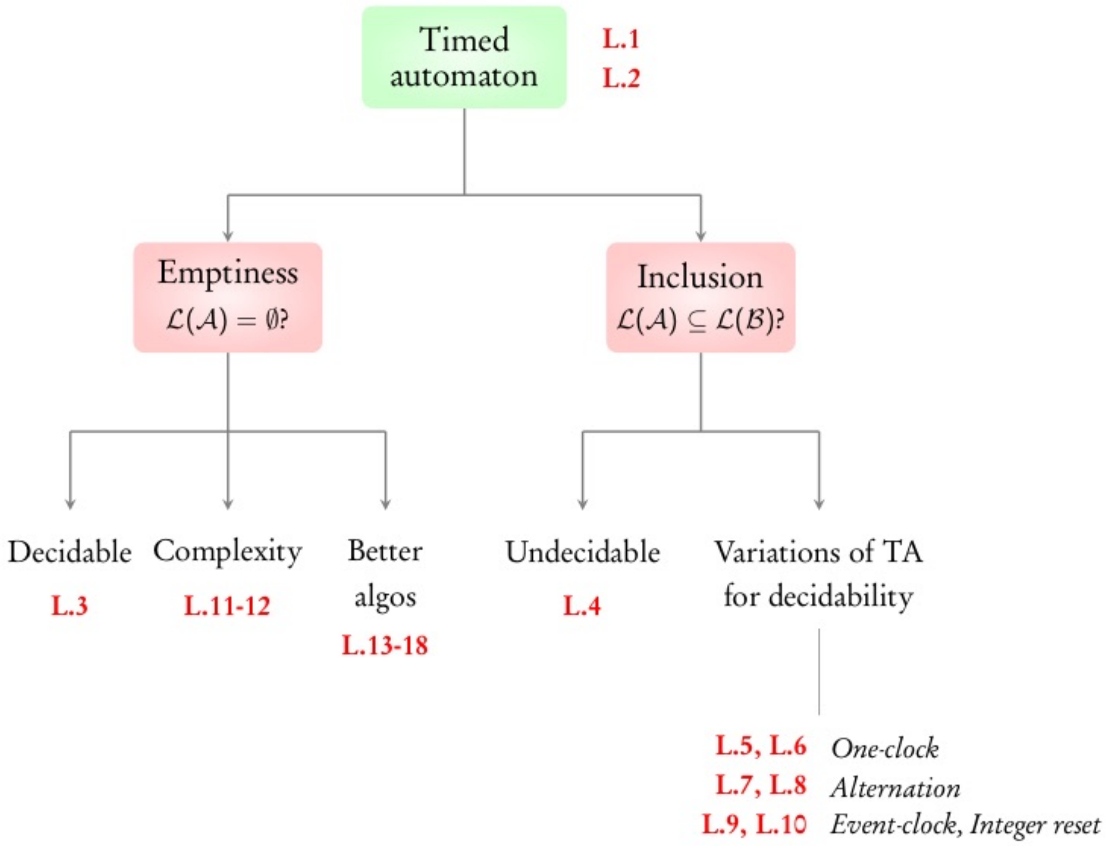
$$\mathcal{L}(\mathcal{A}) \subseteq \mathcal{L}(\mathcal{B})$$

iff

$$\mathcal{L}(\mathcal{A}) \ \cap \ \overline{\mathcal{L}(\mathcal{B})} = \emptyset$$

non-closure under complement ⇒ the above **cannot be done** for TA!

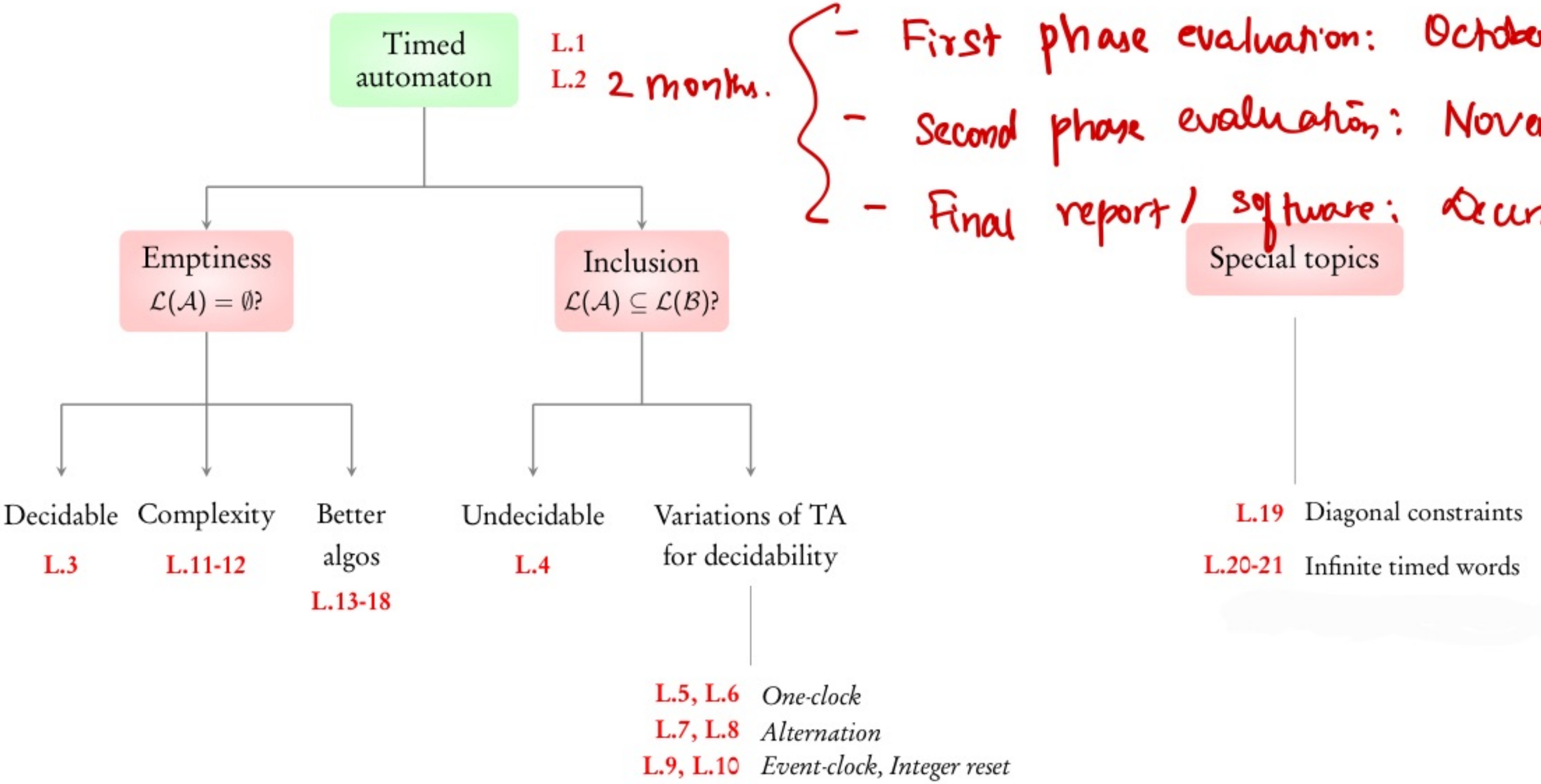
Course plan


Timed automaton
L.1
L.2
Emptiness
$\mathcal{L}(\mathcal{A}) = \emptyset$?
Inclusion
$\mathcal{L}(\mathcal{A}) \subseteq \mathcal{L}(\mathcal{B})$?
Decidable
L.3
Complexity
L.11-12
Better algos
L.13-18
Undecidable
L.4
Variations of TA for decidability
L.5, L.6 One-clock
L.7, L.8 Alternation
L.9, L.10 Event-clock, Integer reset

Course plan

Project plan:

- October 7÷15 : Choose your project.
- First phase evaluation: October 30
- Second phase evaluation: November 30
- Final report / software: December 15



Timed automaton

L.1
L.2 2 months.

Emptiness $\mathcal{L}(\mathcal{A}) = \emptyset$?

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

Special topics

Decidable L.3

Complexity L.11-12

Better algos L.13-18

Undecidable L.4

Variations of TA for decidability

L.5, L.6 *One-clock*
L.7, L.8 *Alternation*
L.9, L.10 *Event-clock, Integer reset*

L.19 Diagonal constraints
L.20-21 Infinite timed words