TIMED AUTOMATA

LECTURE 2

Recall:

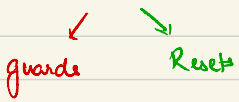
- Timed word $(a_1 a_2 \dots a_n, \tau_1, \tau_2 \dots \tau_n)$

$$\begin{matrix} a_1 & a_2 & \dots & a_n \\ \tau_1 & \tau_2 & & \tau_n \end{matrix}$$

$\tau_i \leq \tau_{i+1}$ $\qquad$ $\tau_i \in \mathbb{R}_{\geq 0}$

- Timed language

- Timed automaton : clocks



- Closure properties: - closed under union, intersection, untime
  - non-closure under complementation

Today:

Theorem: For a timed automaton $\mathcal{A}$, the language Untime $(\mathcal{L}(\mathcal{A}))$ is regular.

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

(L: timed lang.)

$\{ (w_1, \tau_1) (w_2, \tau_2) , \dots \dots \dots \}$

Given a timed automaton $\mathcal{A}$.

Naïve approach:

Remove all guards and resets from $\mathcal{A}$.

└ This does not work.

$\mathcal{L}(\mathcal{A}) = \phi$ (timed language)

Untime $(\mathcal{L}(\mathcal{A})) = \underline{\phi}$

$\mathcal{B}$ obtained by the above construction

$\mathcal{L}(\mathcal{B}) = \{aa\}$

$\mathcal{L}(\mathcal{B}) \neq$ Untime $(\mathcal{L}(\mathcal{A}))$

Modification: We will construct an NFA which maintains information about clock values in its states

---

$\mathcal{A}$ $\xrightarrow{1}$ $S_{\mathcal{A}}$ $\xrightarrow{2}$ Nbd $(\mathcal{A})$ $\xrightarrow{3}$ Region-automaton $(\mathcal{A})$

infinite state infinite alphabet - some correspondence bet. timed language of $\mathcal{A}$ and runs of $S_{\mathcal{A}}$	infinite state finite alphabet $\mathcal{L}(\text{Nba}(\mathcal{A}))$ $=$ Untime $(\mathcal{L}(\mathcal{A}))$	finite state finite alphabet $\mathcal{L}(RA(\mathcal{A}))$ $=$ Untime $(\mathcal{L}(\mathcal{A}))$

$\downarrow$ 1.5

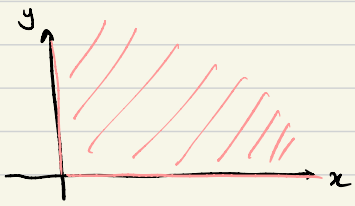
Untime - $S_{\mathcal{A}}$) -

Step 1:

States: (q, v) v is a valuation over clocks

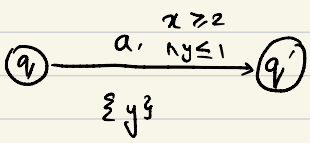
X: clocks. $v: X \to \mathbb{R}_{\geq 0}$

$X = \{x, y\}$ $v_1: \langle x = 1.7,\ y = 10.5 \rangle$



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

Transitions: $(q, v) \xrightarrow{\delta, a} (q', v')$



v

$(q, \langle x = 2.5,\ y = 0.5 \rangle)$

$0, a$

$(q', \langle x = 2.5,\ y = 0 \rangle)$

$0.5, a$

$0.6, a$

$v + \delta = \langle x = 3, y = 1 \rangle$

$(q', \langle x = 3,\ y = 0 \rangle)$

$[\{y\}](v + \delta)$

There is a transition:

$$(q, v) \xrightarrow{\delta, a} (q', v')$$

if i) there exists some transition in the timed automaton in this form:

$$q \xrightarrow[R]{a,\ g} q'$$

ii) $v + \delta \models g$ ($v + \delta$ satisfies guard g)

iii) $v' = [R](v+\delta)$ $\quad [R](v+\delta)(x) = \begin{cases} 0 & x \in R \\ v+\delta(x) & x \notin R \end{cases}$

Let's call this automaton S_A:

State: $Q \times \mathbb{R}_{\geq 0}^{|X|}$ (valuation) $\quad$ Transitions: as above

Alphabet: $\mathbb{R}_{\geq 0} \times \Sigma$ $\quad$ Initial states: (q_0, v_0)

where $q_0 \in Q_0$

v_0: every clock is set to 0.

Accepting states: $F \times \mathbb{R}_{\geq 0}^{|X|}$

What can we say about S_A?

- Suppose there is a run:

$$(q_0, v_0) \xrightarrow{\delta_1, a_1} (q_1, v_1) \xrightarrow{\delta_2, a_2} (q_2, v_2) \cdots \xrightarrow{\delta_n, a_n} (q_n, v_n)$$

s.t. $q_n \in F$

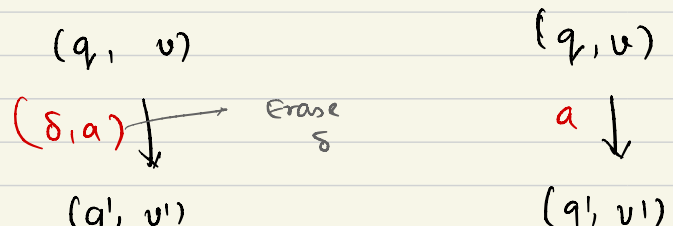
Then timed word $\begin{pmatrix} a_1 & a_2 & \cdots & a_n \\ \delta_1 & \delta_1+\delta_2 & & \delta_1+\delta_2+\cdots+\delta_n \end{pmatrix} \in \mathcal{L}(A)$

- Conversely, suppose $\begin{pmatrix} a_1 & a_2 & \cdots & a_n \\ \tau_1 & \tau_2 & & \tau_n \end{pmatrix} \in \mathcal{L}(A)$

then there is a run in S_A of the form:

$$(q_0, v_0) \xrightarrow{\tau_1, a_1} (q_1, v_1) \xrightarrow{\tau_2-\tau_1, a_2} (q_2, v_2) \longrightarrow \cdots \xrightarrow{\tau_n-\tau_{n-1}} (q_n, v_n)$$

Modify S_A to get untime($\mathcal{L}(A)$)



Untiming S_A : - Untime - S_A

- Erase the δ part in the transitions

States: $Q \times \mathbb{R}_{\geq 0}^{|X|}$, Alphabet = Σ Acc state: $F \times \mathbb{R}_{\geq 0}^{|X|}$

Initial state: $Q_0 \times \{\vec{v}_0\}$

Transitions: $(q, v) \xrightarrow{a} (q', v')$

if $\exists\ \delta \geq 0$ s.t.

$(q, v) \xrightarrow{\delta, a} (q', v')$ is a transition in S_A.

Claim: $\mathcal{L}(\text{Untime} \cdot S_A) = \text{Untime}(\mathcal{L}(A))$

Proof: $\mathcal{L}(\text{Untime} \cdot S_A) \subseteq \text{Untime}(\mathcal{L}(A))$

Pick $a_1\ a_2\ a_3 \cdots a_n \in \mathcal{L}(\text{Untime} \cdot S_A)$.

$\therefore$ there is a run:

$$(q_0, v_0) \xrightarrow{a_1} (q_1, v_1) \xrightarrow{a_2} \cdots \xrightarrow{a_n} (q_n, v_n) \quad q_n \in F$$

in Untime - S_A

$\therefore$ there exists a run:

$$(q_0, v_0) \xrightarrow{\delta_1, a_1} (q_1, v_1) \xrightarrow{\delta_2, a_2} \cdots \xrightarrow{\delta_n, a_n} (q_n, v_n) \text{ in } S_A$$

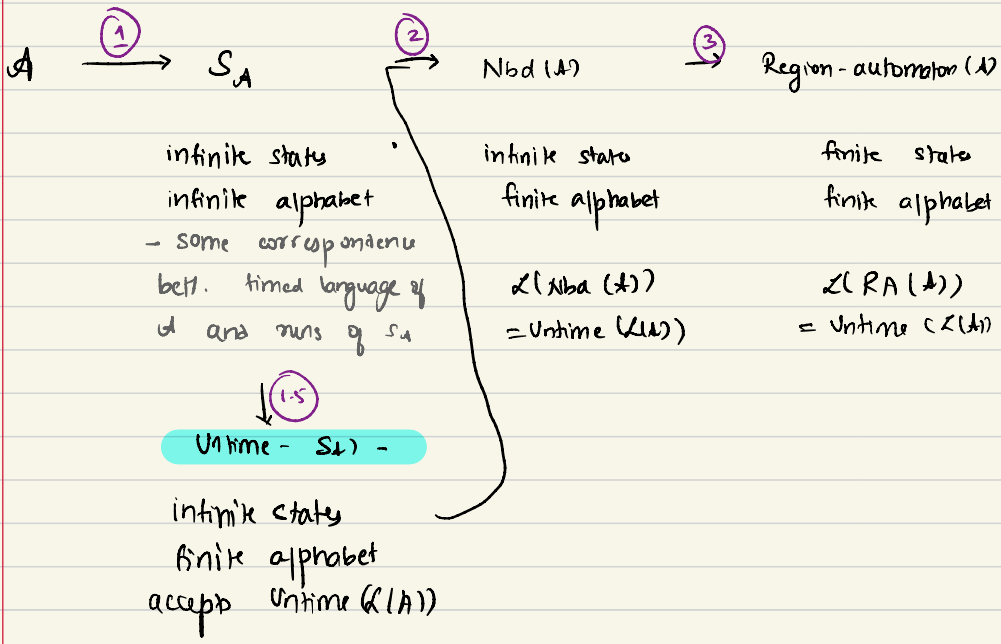
$\Rightarrow$ $\begin{pmatrix} a_1 & a_2 & \cdots & a_n \\ \delta_1 & \delta_1+\delta_2 & & \delta_1+\cdots+\delta_n \end{pmatrix} \in \mathcal{L}(A)$

$\Rightarrow$ $a_1\ a_2 \cdots a_n \in \text{Untime}(\mathcal{L}(A))$

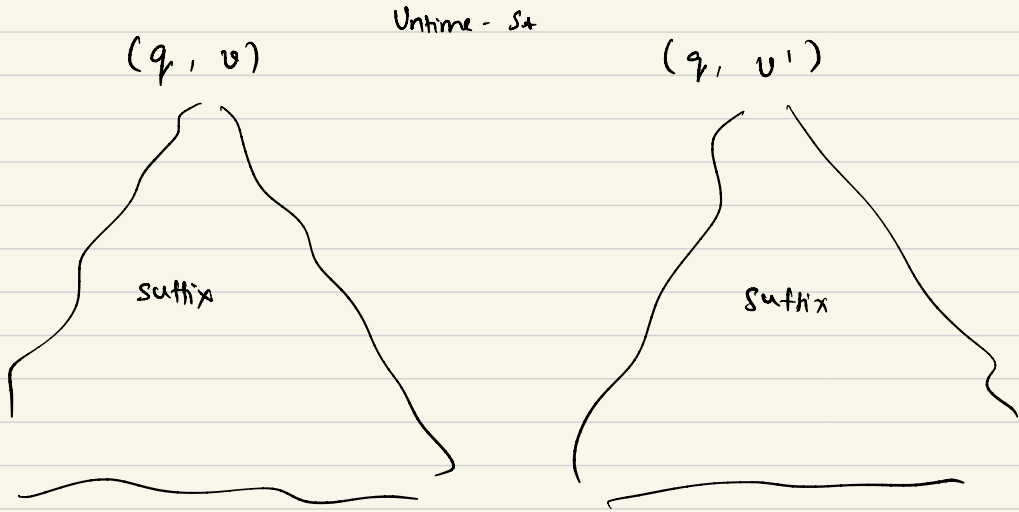
The other direction: $\text{Untime}(\mathcal{L}(A)) \subseteq \mathcal{L}(\text{Untime} \cdot S_A)$ Exercise.

Automaton untime - S_A :

- Infinitely many states
- finite alphabet
- accepts untime ($\mathcal{L}(A)$)



Step 2: Neighbourhood automaton.



Merge states (q, v) (q, v') if set of suffixes that are accepted from these states is the same.

Idea: Find sufficient conditions on v and v' so that the set of suffixes is the same from (q, v) and (q, v') for every q.

Neighbourhoods in 2-D:

$0 < x < 1$

$0 < y < 1$

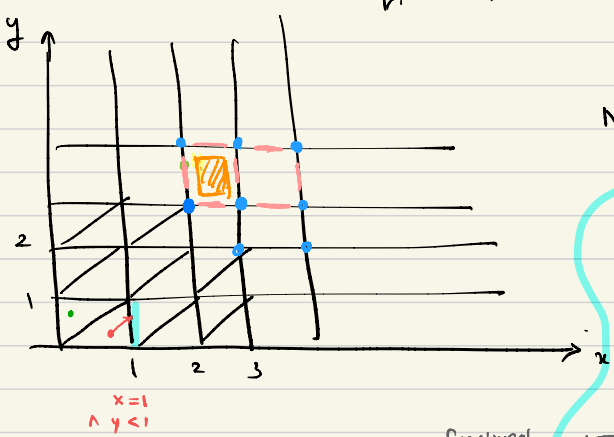
$x > y$

What we want:

$$(q_1, v) \xrightarrow{\delta_1 a_1} \xrightarrow{\delta_2 a_2} \cdots$$

$$\simeq_{nbd}$$

$$(q_1, v') \xrightarrow{\delta'_1 a_1} \xrightarrow{\delta'_2 a_2} \cdots$$



$x = 1$

$\wedge\ y < 1$

Neighbourhood equivalence in 2D:

$v \simeq_{nbd} v'$ if

i) $\lfloor v(x) \rfloor = \lfloor v'(x) \rfloor$

$\lceil v(y) \rceil = \lceil v'(y) \rceil$

ii) $\{v(x)\} = 0 \iff \{v'(x)\} = 0$

$0 = \{v(y)\} \iff 0 = \{v'(y)\}$

(fractional part of $v(x)$)

iii) $\{v(x)\} < \{v(y)\} \iff \{v'(x)\} < \{v'(y)\}$

$\{v(x)\} = \{v(y)\} \iff \{v'(x)\} = \{v'(y)\}$

$\simeq_{nbd}$ equivalence with $\{\eta, \gamma\}$.

- It is an equivalence relation:

$v \simeq_{nbd} v'$ if

i) $\lceil v(x) \rceil = \lceil v'(x) \rceil$
$\lceil v(y) \rceil = \lceil v'(y) \rceil$

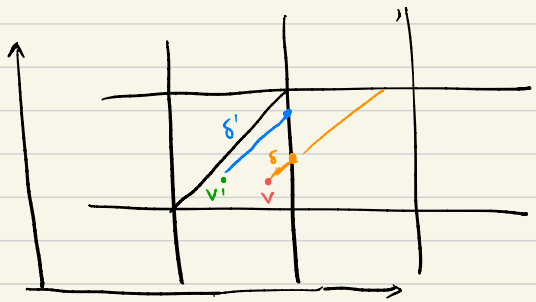
ii) $v(x) = \gamma \iff v'(x) = \gamma$
$v(y) = \gamma \iff v'(y) = \gamma$

iii) $v(x) < v(y) \iff v'(x) < v'(y)$
$v(x) = v(y) \iff v'(x) = v'(y)$

Claim: Suppose $v \simeq_{nbd} v'$.

For every $\delta \geq 0$. $\exists\ \delta'$ s.t.

$$v + \delta \simeq_{nbd} v' + \delta'$$



Next class: We will complete construction of neighbourhood automaton.