TIMED AUTOMATA

LECTURE 3

GOALS OF TODAY'S LECTURE:

Theorem: Given a timed automaton $\mathcal{A}$, we can construct a finite automaton for Untime $(\mathcal{L}(\mathcal{A}))$

Untime $(\mathcal{L}(\mathcal{A}))$:

$\mathcal{L}(\mathcal{A})$: timed language accepted by $\mathcal{A}$.

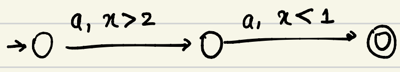
$$\{(\vec{w}, \tau) \mid \mathcal{A} \text{ has an accepting run}\}$$

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

Naïve construction:

Take $\mathcal{A}$ – remove all guards and resets – NFA $\mathcal{B}$

Claim: Untime $(\mathcal{L}(\mathcal{A})) = \mathcal{L}(\mathcal{B})$ Wrong!



$\mathcal{A}$:

$\mathcal{L}(\mathcal{A}) = \Phi$

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

Moral:

Need to store "some property" about clocks in the state.

$$\mathcal{A} \xrightarrow{1} S_{\mathcal{A}} \xrightarrow{2} \text{Untime-}S_{\mathcal{A}} \xrightarrow{3} Nbd(\mathcal{A}) \xrightarrow{4} Reg(\mathcal{A})$$

$S_{\mathcal{A}}$	Untime-$S_{\mathcal{A}}$	$Nbd(\mathcal{A})$	$Reg(\mathcal{A})$
- infinite states	- infinite states	infinite states	finite states
- infinite alphabet	- finite alphabet	- finite alphabet	finite alp
- $\mathcal{L}(\mathcal{A})$	- Untime $(\mathcal{L}(\mathcal{A}))$	- Untime $(\mathcal{L}(\mathcal{A}))$	Untime $(\mathcal{L}(\mathcal{A}))$
	uncountable	countable	

Step 1: Set of clocks X

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

$X = \{x, y, z\}$

$v_1 : \langle 3.2, 5.7, 100 \rangle$

$S_{\mathcal{A}}$:

States : (q, v) — state of $\mathcal{A}$, Valuation

$(q_1 \langle 3.2, 5.7, 100 \rangle)$

Alphabet : $(a_1\ a_2\ a_3 \ldots a_n, \tau_1, \tau_2 \ldots \tau_n)$

$$\begin{bmatrix} a_1 \\ \tau_1 \end{bmatrix} \begin{bmatrix} a_2 \\ \tau_2 \end{bmatrix} \cdots \begin{bmatrix} a_n \\ \tau_n \end{bmatrix}$$

$\Sigma \times \mathbb{R}_{\geq 0}$ $\quad (a_1, \tau_1)\ (a_2, \tau_2) \ldots (a_n, \tau_n)$

$(\tau_1, a_1)\ (\tau_2, a_2) \ldots (\tau_n, a_n)$

Transitions: $(q, v) \xrightarrow{(\delta, a)} (q_1, v_1)$

if $\exists$ a transition in $\mathcal{A}$: $(q, a, g, R, \quad)$

$$q \xrightarrow[R]{a,\, g} q_1$$

<u>Transitions:</u> $\quad (q, v) \xrightarrow{(\delta,\, a)} (q_1, v_1)$

if 1) $\exists$ a transition in $\mathcal{A}$: (q, a, g, R, q_1)

$$q \xrightarrow[R]{a,\, g} q_1$$

2) $v + \delta \models g$ (satisfies the constraint g)

3) $v_1 = \underline{[R](v + \delta)}$

$\downarrow$

$x \in R \rightarrow$ to 0

$x \notin R \rightarrow v + \delta(x)$

$$u = \langle \overset{x}{3.2},\ \overset{y}{5.7},\ \overset{z}{10.8} \rangle \longrightarrow \langle 0, 0, 10.8 \rangle$$

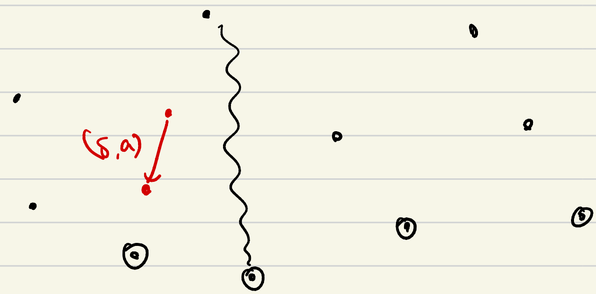
$R = \{x, y\}$

$$[\{x,y\}]\langle 3.2, 5.7, 10.8 \rangle = \langle 0, 0, 10.8 \rangle$$

<u>Initial states:</u> $\quad (q_0, \langle 0, 0, 0, \ldots 0 \rangle) \quad q_0 \in Q_0$

<u>Final states:</u> $\quad (q, v) \quad q \in F$

$S_{\mathcal{A}}$



$S_{\mathcal{A}}$

space of accepting runs.

$\bar{0}$

(δ_1, a_1)

(δ_2, a_2)

(δ_3, a_3)

$\vdots$

(δ_n, a_n)

(δ_1, a_1)

$(\delta_1+\delta_2, a_2)$

$(\delta_1+\delta_2+\delta_3, a_3)$

$\vdots$

$(\delta_1+\dots+\delta_n, a_n)$

$$\mathcal{L}(S_{\mathcal{A}}) = \{ (a_1 a_2 \dots a_n,\ \delta_1, \delta_1+\delta_2, \dots \delta_1+\dots\delta_n) \mid S_{\mathcal{A}} \text{ has an accepting run on } (\delta_1,a_1)(\delta_2,a_2)\dots(\delta_n,a_n) \}$$

Lemma: $\mathcal{L}(S_{\mathcal{A}}) = \mathcal{L}(\mathcal{A})$

Step 2: **Question:** Can we perform the "naive construction" on $S_{\mathcal{A}}$?

- Suppose in every transition $\bullet \xrightarrow{(\delta, a)} \bullet$, we erase the time component.

$$\bullet \xrightarrow{a} \bullet$$

- Will the resulting "untimed automaton with infinite states" accept $Untime(\mathcal{L}(\mathcal{A}))$? Yes.

Untime - $S_{\mathcal{A}}$: same states as $S_{\mathcal{A}}$.

Transitions: $(q, v) \xrightarrow{a} (q_1, v_1)$

if there exists $(q, v) \xrightarrow{(\delta, a)} (q_1, v_1)$ in $S_{\mathcal{A}}$

Step 2: **Question:** Can we perform the "naive construction" on S_A?

- Suppose in every transition $\bullet \xrightarrow{(\delta, a)} \bullet$, we erase the time component.

$$\bullet \xrightarrow{a} \bullet$$

- Will the resulting "untimed automaton with infinite states" accept $Untime(\mathcal{L}(A))$? Yes.

Untime-S_A : same states as S_A.

Transitions: $(q, v) \xrightarrow{a} (q_1, v_1)$

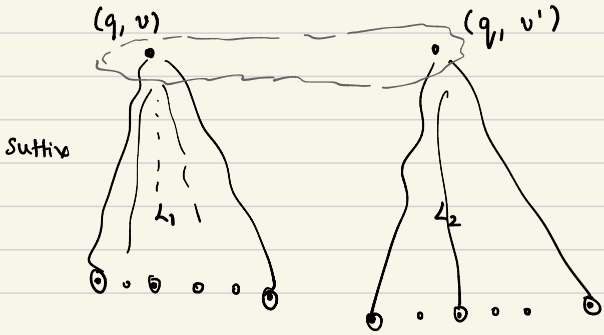
if there exists $(q, v) \xrightarrow{(\delta, a)} (q_1, v_1)$ in S_A

$$\begin{tikzcd}
(q, v) \arrow[d, bend right, "{(\delta_1, a)}"'] \arrow[d, bend left, "{\delta_2, a}"] \\
(q_1, v_1)
\end{tikzcd}$$

$\mathcal{L}(\text{Untime-}S_A)$: $\{ w \in \Sigma^* \mid \text{Untime-}S_A$ has an accepting run on $w \}$

Step 3: Clubbing valuations together

(Untime - S_t)

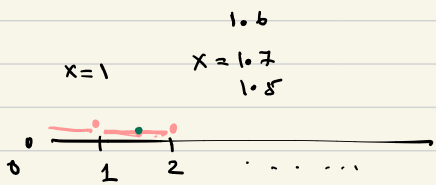


If the suffix languages are same, we can club them together into a single state.

Goal: Define an equivalence between valuations so that equivalent valuations have the same suffix language.

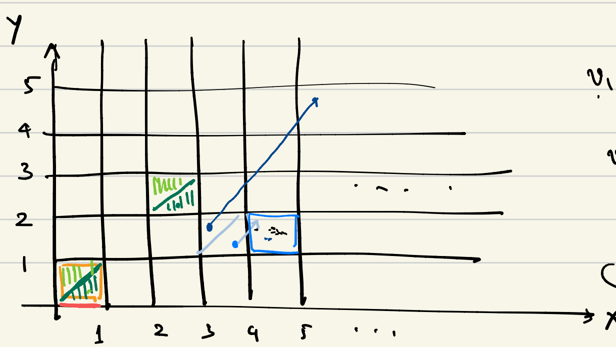
Neighbourhood equivalence:

1-D: single clock, $X = \{x\}$



$y < 2 \wedge x > 4$ $\longrightarrow$

2-D



v_1: $\langle 3.6, 1.2 \rangle$

v_2: $\langle 3.2, 1.6 \rangle$

$v_1 \sim v_2$

$(0.5, a)$

$\langle 4.1, 1.7 \rangle$

$\cancel{\checkmark} \; \delta'$

Cannot satisfy $y < 2 \wedge x > 4$

$\{x\} < \{y\}$

$\{y\} < \{x\}$

$0 < x < 1$
$0 < y < 1$

$0 < x < 1$
$y = 0$

Neighbourhood equivalence $\simeq_{nbd}$

↳ a binary relation over valuations.

$v \simeq_{nbd} v'$ if:

1) $\forall x \in X$: $\lfloor v(x) \rfloor = \lfloor v'(x) \rfloor$

2) $\forall x \in X$: $\{v(x)\} = 0$ iff $\{v'(x)\} = 0$

fractional part

3) $\forall x, y \in X$: i) $\{v(x)\} < \{v(y)\} \iff \{v'(x)\} < \{v'(y)\}$

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

$0.1 < 0.3 < 0.6 < 0.8$

	0.8	0.1	0.6	0.1	0.3	
v:	10.8	5.1	4.6	7.1	5.3	3.0
v':	10.6	5.2	4.5	7.2	5.35	3.0
	0.6	0.2	0.5	0.2	0.35	

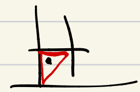
$v \simeq_{nbd} v'$

$0.2 < 0.35 < 0.5 < 0.6$

Neighbourhood automaton : NBD($\mathcal{A}$) :

Neighbourhood automaton : NBD($\mathcal{A}$) :

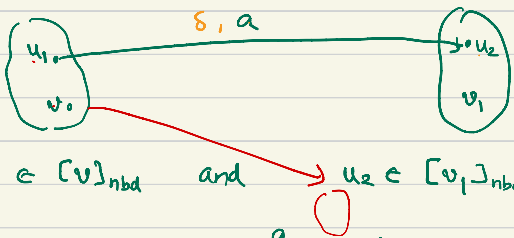
- State: $(q_1\ [v]_{nbd})$



- Alphabet - Σ.
- Transition:

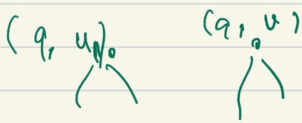
$$(q,\ [v]_{nbd}) \xrightarrow{a} (q_1,\ [v_1]_{nbd})$$

if $\exists$



$u_1 \in [v]_{nbd}$ and $u_2 \in [v_1]_{nbd}$ s.t.

$(q_1\ u_1) \xrightarrow{a} (q_1,\ u_2)$ in Untime-S_A.



To show: $u \simeq_{nbd} v \Rightarrow \forall$ transition in Untime-S_A: $(q, u) \xrightarrow{a} (q_1, u_1)$

there exists $(q, v) \xrightarrow{a} (q_1, v_1)$

s.t. $u_1 \simeq_{nbd} v_1$