TIMED AUTOMATA

LECTURE 14

Associating a word to each configuration:

$K = 10$

C_1: $\{ (q_1, 5.7), (q_1, 2.3), (q_2, 7.3), (q_2, 10.4), (q_3, 2.9) \}$

$\downarrow$

Encoding:

$H(C_1)$: $\{ (q_1, r_2^3), (q_2, r_7^8) \} \{ (q_1, r_5^6) \} \{ (q_3, r_2^3) \} \{ (q_2, r_{10}^{\infty}) \}$

Time-Successors of H(C):

(iii) (i) (ii)

$$H(C_1) \xrightarrow{\tau} \{(q_3, r_3)\} \; \{(q_1, r_2^3), (q_2, r_7^8)\} \; \{(q_1, r_5^6)\} \; \{(q_2, r_{10}^{\infty})\}$$

$\downarrow \tau$

$$\{(q_3, r_3^4)\} \; \{(q_1, r_2^3), (q_2, r_7^8)\} \; \{(q_1, r_5^6)\} \; \{(q_2, r_{10}^{\infty})\}$$

$\downarrow \tau$

$$\{(q_1, r_6)\} \; \{(q_3, r_3^4)\} \; \{(q_1, r_2^3), (q_2, r_7^8)\} \; \{(q_2, r_{10}^{\infty})\}$$

$\downarrow \tau$

$$\{(q_1, r_6^7)\} \; \{(q_3, r_3^4)\} \; \{(q_1, r_2^3), (q_2, r_7^8)\} \; \{(q_2, r_{10}^{\infty})\}$$

$\downarrow \tau$

$\vdots$

$$\{ (q_1, r_{10}^{\infty}), (q_2, r_{10}^{\infty}), (q_3, r_{10}^{\infty}) \}$$

Claim : Let H_1 , H_2 be encodings s.t.

$$H_1 \xrightarrow{\tau} H_2$$

Then: for every $C_2 \in H_2$, there exists $C_1 \in H_1$ s.t.

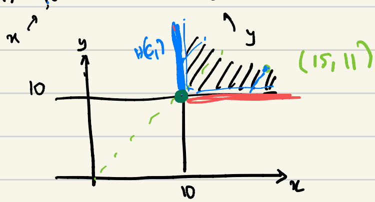
$$C_1 \xrightarrow{\delta} C_2 \quad \text{for some} \quad \delta > 0$$

This claim is false!

- Consider $H(C_1)$: $\{ (q_1, r_{10}) \} \ \{ (q_2, r_{10}^{\infty}) \}$

  $''$ H_1

$$H_1 \xrightarrow{\tau} \{ (q_1, r_{10}^{\infty}), (q_2, r_{10}^{\infty}) \}$$



Lemma: Let H_1, H_2 be encodings s.t.

$$H_1 \xrightarrow{T} H_2$$

Then: for every $C_1 \in H_1$ there exists a $C_2 \in H_2$ s.t.

$$C_1 \xrightarrow{\delta} C_2 . \quad \text{for some } \delta > 0$$

Proof: (Sketch). Given $H_1 = w_1 \, w_2 \, \dots \, w_n \, w_\infty \xrightarrow{T} H_2$

H_2: - 1. If w_1 "is an integer":

$$H_2 = w_1' \, w_2 \, \dots \, w_n \, w_\infty'$$

- 2. If w_1 "is not an integer"

$$H_2 = w_n' \, w_1 \, w_2 \, \dots \, w_n \, w_\infty$$

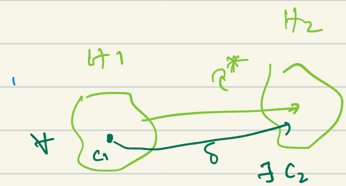
- In each case, find a specific δ based on the given configuration $C_1 \in H_1$.

Reflexive-transitive closure of τ:

τ: immediate successor

$$
\begin{tikzcd}
H_1 \arrow[r, "\tau"] \arrow[loop below, "\tau^*"] \arrow[r, bend right, "\tau^*"'] \arrow[rr, bend right, "\tau^*"'] \arrow[rrr, bend right, "\tau^*"'] \arrow[rrrrr, bend right=60, "\tau^*"'] & H_2 \arrow[r, "\tau"] & H_3 \arrow[r, "\tau"] & H_4 \arrow[r] & \cdots \arrow[r, "\tau"] & H
\end{tikzcd}
$$

τ^*: reflexive transitive closure of τ:



The previous lemma carries over to τ^*:

> **Lemma:** Let H_1, H_2 be encodings s.t.
>
> $$H_1 \xrightarrow{\tau^*} H_2$$
>
> Then: for every $c_1 \in H_1$ there exist a $c_2 \in H_2$ s.t.
>
> $$c_1 \xrightarrow{\delta} c_2 . \quad \text{for some } \delta \geq 0$$

Moral:

- Time successors of an encoding can be effectively computed
- Above lemma.

Discrete - successors of $H(C)$

$$q_1 \xrightarrow{x \leq 3,\, a} q_1'$$

$$q_3 \xrightarrow[\{x\}]{x \geq 2,\, a} q_3'$$

$$\{ (q_1, r_3)(q_2, r_7) \} \quad \{ (q_1, r_2^3), (q_3, r_4^5) \} \quad \{ (q_4, r_{10}^{\infty}) \}$$

$$\Big\downarrow a$$

$$\{ (q_1', r_3), (q_3', r_0) \} \quad \{ (q_1', r_2^3) \}$$

There can also be multiple edges on 'a' from a state.

Same lemma as before applies to discrete successors as well.

Lemma: Let H_1, H_2 be encodings s.t.

$$H_1 \xrightarrow{a} H_2$$

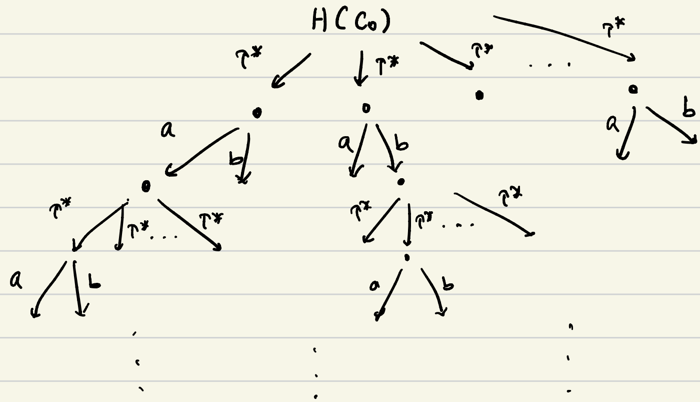
Then: for every $C_1 \in H_1$ there exist a $C_2 \in H_2$ s.t.

$$C_1 \xrightarrow{a} C_2 .$$

Moral:

- Discrete successor computation is also effective
- Above lemma

Construction of a tree:- "Encoding tree"



finitely many successors.

- finitely branching tree.

Soundness:

Proposition: For every path $H_0 \xrightarrow{\tau^*} \cdot \xrightarrow{a_1} H_1 \xrightarrow{\tau^*} \xrightarrow{a_2} H_2 \rightarrow \cdots \xrightarrow{a_n} H_n$

there exists a run: $C_0 \xrightarrow{\delta_1, a_1} C_1 \xrightarrow{\delta_2, a_2} C_2 \rightarrow \cdots\cdots \xrightarrow{\delta_n, a_n} C_n$

s.t. $C_i \in H_i$

Completeness:

Proposition: For every run $C_0 \xrightarrow{\delta_1, a_1} C_1 \xrightarrow{\delta_2, a_2} C_2 \rightarrow \cdots\cdots \xrightarrow{\delta_n, a_n} C_n$

there exists a path: $H(C_0) \xrightarrow{\tau^*} \xrightarrow{a_1} H(C_1) \xrightarrow{\tau^*} \xrightarrow{a_2} H(C_2) \rightarrow \cdots \xrightarrow{a_n} H(C_n)$

Theorem: Automaton is not universal

iff

a bad node is reached in the encoding tree.

Encoding tree can be constructed on-the-fly.

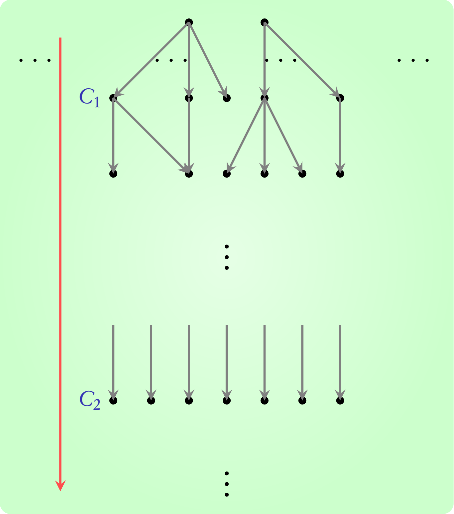
Termination??

A bad node in the tree is an encoding where all the locations are non-accepting.

Step 3:

The domination order

finite **domination** order $\preccurlyeq$



$C_1 \preccurlyeq C_2$ iff:

C_2 goes to a **bad** config $\quad\Rightarrow\quad$ C_1 goes to a **bad** config. too

Look at $H(C_1)$ and $H(C_2)$, the words over Λ^*

$$\Lambda = \mathcal{P}(\ Q \times REG\ )$$

Look at $H(C_1)$ and $H(C_2)$, the words over Λ^*

$$\Lambda = \mathcal{P}(\ Q \times REG\ )$$

Let $\subseteq$ be the **inclusion** (quasi-)order on Λ

Look at $H(C_1)$ and $H(C_2)$, the words over Λ^*

$$\Lambda = \mathcal{P}(\ Q \times REG\ )$$

Let $\subseteq$ be the **inclusion** (quasi-)order on Λ

Consider the induced monotone domination order $\preccurlyeq$ over Λ^*

$H(C_1)$: $\{(q_0, r_0)\}\ \{(q_1, r_0^1), (q_0, r_2^3)\}$

$\preccurlyeq$ $\preccurlyeq$

$H(C_2)$: $\{(q_0, r_0), (q_1, r_1)\}\ \{(q_2, r_2^3)\}\ \{(q_1, r_0^1), (q_0, r_2^3), (q_2, r_1^2)\}$

Look at $H(C_1)$ and $H(C_2)$, the words over Λ^*

$$\Lambda = \mathcal{P}(\ Q \times REG\ )$$

Let $\subseteq$ be the **inclusion** (quasi-)order on Λ

Consider the induced monotone domination order $\preccurlyeq$ over Λ^*

$C_1 \to H(C_1)$ $\{(q_0, r_0)\}\ \{(q_1, r_0^1), (q_0, r_2^3)\}$ $\{(q_0, 0)\}\ \{(q_1, 0.5), (q_0, .5)\}$

$$\preccurlyeq$$

$C_2 \to H(C_2)$: $\{(q_0, r_0), (q_1, r_1)\}\ \{(q_2, r_2^3)\}\ \{(q_1, r_0^1), (q_0, r_2^3), (q_2, r_1^2)\}$

Theorem: If $H(C_1) \preccurlyeq H(C_2)$, then $\exists C_2' \subseteq C_2$ s.t. $C_1 \sim C_2'$

$\{(q_0, 0), (q_1, 1)\}\ \{(q_2, 2.2)\}\ \{(q_1, 0.4), (q_0, 2.4), (q_2, 1.4)\}$

Look at $H(C_1)$ and $H(C_2)$, the words over Λ^*

$$\Lambda = \mathcal{P}(\ Q \times REG\ )$$

Let $\subseteq$ be the **inclusion** (quasi-)order on Λ

Consider the induced monotone domination order $\preccurlyeq$ over Λ^*

$$\{(q_0, r_0)\}\ \ \{(q_1, r_0^1), (q_0, r_2^3)\}$$

$$\preccurlyeq$$

$$\{(q_0, r_0), (q_1, r_1)\}\ \ \{(q_2, r_2^3)\}\ \ \{(q_1, r_0^1), (q_0, r_2^3), (q_2, r_1^2)\}$$

Theorem: If $H(C_1) \preccurlyeq H(C_2)$, then $\exists C_2' \subseteq C_2$ s.t. $C_1 \sim C_2'$

$\subseteq$ is a wqo as Λ is **finite**. Therefore, $\preccurlyeq$ is a **wqo** due to **Higman's lemma**

Final algorithm

- Start from $H(C_0)$, where C_0 is the **initial configuration**
- Successor computation is **effective**
- Termination guaranteed as **domination order is wqo**

— Before each successor computation, check if there is a smaller word. If yes, do not explore this node.

A is **universal** iff the algorithm does **not reach a bad node**

One-clock

Universality is **decidable** for one-clock timed automata

One-clock

Universality is **decidable** for one-clock timed automata

For **two clocks**, we know universality is undecidable

One-clock

Universality is **decidable** for one-clock timed automata

For **two clocks**, we know universality is undecidable

Where does this algorithm go wrong when A has two clocks?

Two clocks

State: (q, u, v)

Configuration: $\{(q_1, u_1, v_1), (q_2, u_2, v_2), \ldots, (q_n, u_n, v_n)\}$

At the **least**, the following should be remembered while abstracting:

- relative ordering between fractional parts of x
- relative ordering between fractional parts of y

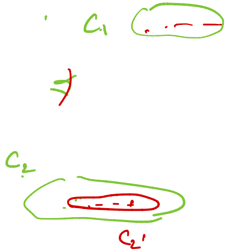
Current encoding can remember **only one** of them

Other encodings possible?

Consider some domination order $\preccurlyeq$

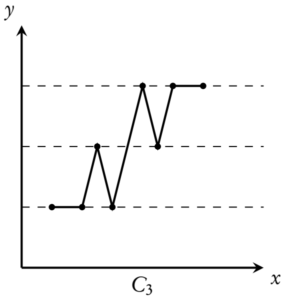
$C_1 \not\preccurlyeq C_2$ if for all $C_2' \subseteq C_2$:

- either relative order of clock x does not match
- or relative order of clock y does not match

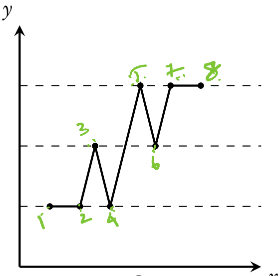


In the next slide: **No wqo** possible!

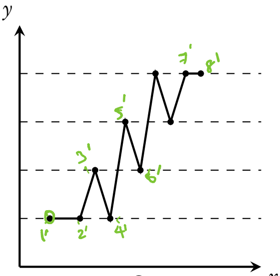
An infinite **non-saturating** sequence $C_1, C_2, C_3, \ldots$



An infinite **non-saturating** sequence $C_1, C_2, C_3, \ldots$

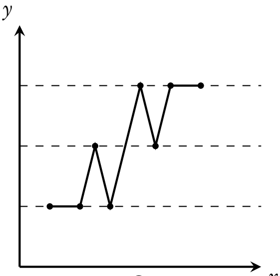


C_3

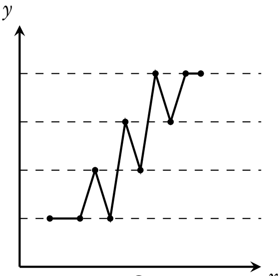


C_4

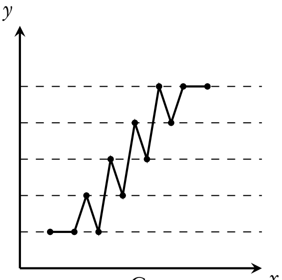
An infinite **non-saturating** sequence $C_1, C_2, C_3, \ldots$



C_3



C_4



C_5

Conclusion

- An algorithm for **universality** when A has one clock
- Can be **extended** for $\mathcal{L}(B) \subseteq \mathcal{L}(A)$ when A has one-clock