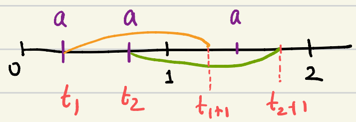
TIMED AUTOMATA

LECTURE 19

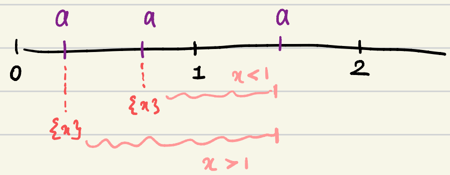
Clarification about the expressive power of 1 - ATA:

Question: Let $L_1 = \{ (aaa, t_1 t_2 t_3) \mid 0 < t_1 < t_2 < 1,\ t_1 + 1 < t_3 < t_2 + 1 \}$



Can you construct a 1- ATA for L_1?

Idea:



1- ATA:

$$(q_0, a, 0<x<1) \longmapsto (p_1, \{x\}) \wedge (s_1, \phi)$$

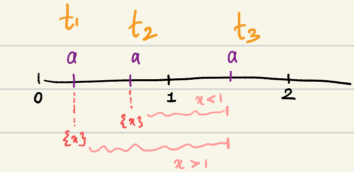
$$(p_1, a, \text{true}) \longmapsto (p_2, \phi)$$

$$(s_1, a, x<1) \longmapsto (s_2, \{x\})$$

$$(p_2, a, x>1) \longmapsto (f, \phi)$$

$$(s_2, a, x<1) \longmapsto (f, \phi)$$

$(f, a, \text{true}) \longmapsto (\text{reject}, \phi)$, $(\text{reject}, a, \text{true}) \longmapsto (\text{reject}, \phi)$



1-ATA: $(q_0, a, 0<x<1) \longmapsto (p_1, \{x\}) \wedge (s_1, \phi)$

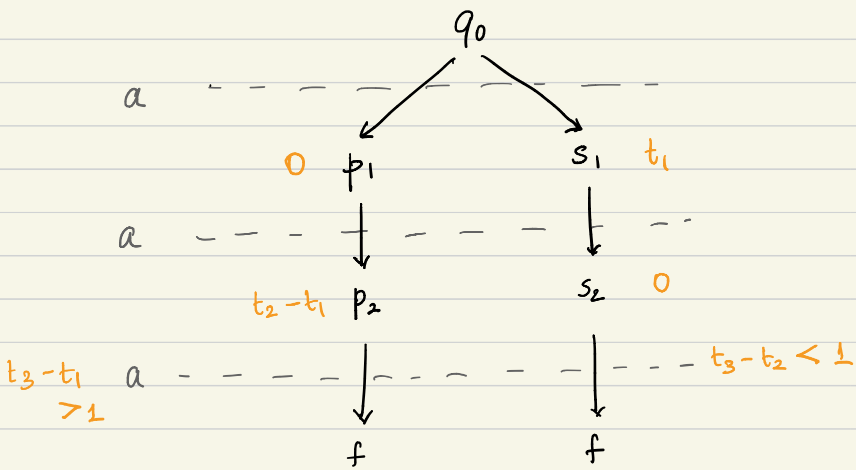
$(p_1, a, true) \longmapsto (p_2, \phi)$

$(s_1, a, x<1) \longmapsto (s_2, \{x\})$

$(p_2, a, x>1) \longmapsto (f, \phi)$

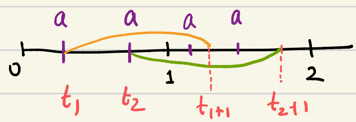
$(s_2, a, x<1) \longmapsto (f, \phi)$

$(f, a, true) \longmapsto (reject, \phi)$, $(reject, a, true) \longmapsto (reject, \phi)$

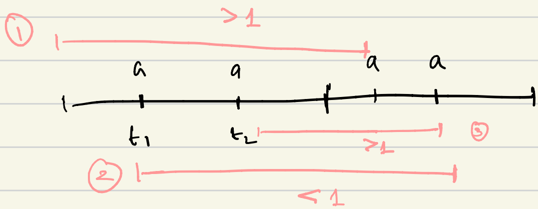


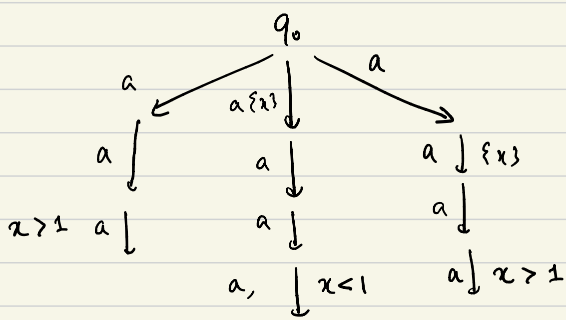
A small modification of the previous example:

Question: Let $L_2 = \{ (aaaa, t_1 t_2 t_3 t_4) \mid 0 < t_1 < t_2 < 1,\ 1 < t_3 < t_4,\ t_1 + 1 < t_4 < t_2 + 1 \}$



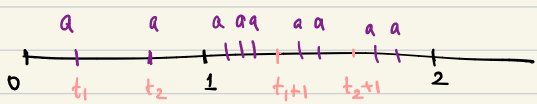
Can you construct a 1-ATA for L_2?



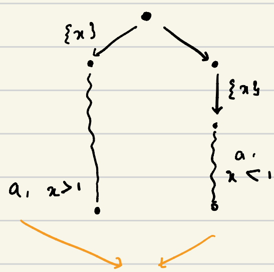


Question:

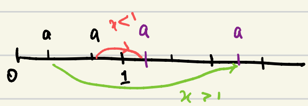
$$\mathcal{L}_3 = \Big\{ (a^k, t_1 t_2 \dots t_k) \mid k \geq 3 : \quad 0 < t_1 < t_2 < 1, \quad \exists j \geq 3 \text{ s.t. } t_1 + 1 < t_j < t_{2+1}, \quad t_3 > 1 \Big\}$$



Problem:



The two a's could be different

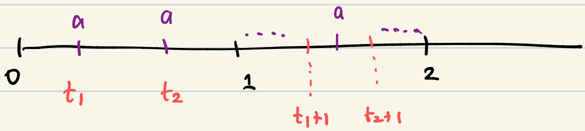


- This is an intuition that $\mathcal{L}_3$ cannot be accepted by a 1-ATA.
- However, proving that a language cannot be accepted by a 1-ATA is difficult.
- We will see another example given in the paper, for which there is a proof that it cannot be accepted by a 1-ATA.

$\mathcal{L} = \{ (a^k, t_1 t_2 \dots t_k) \mid 0 < t_1 < t_2 < 1$

$1 < t_3, \dots, t_k < 2$

there is exactly one a between $t_1 + 1$ and $t_2 + 1$ $\}$



- $\mathcal{L}$ can be accepted by a deterministic T.A with 2 clocks.

Goal: To prove that $\mathcal{L}$ cannot be accepted by a 1-ATA.

Step 1: Understand some property of DFAs

Step 2: How Step 1 translates to untimed alternating finite automata

Step 3: Any 1-ATA accepting $\mathcal{L}$ behaves like an untimed AFA in the interval $(1, 2)$, where clocks are useless.

Step 4: Use Step 1 and 2 in 3 to get a contradiction.

Step 1: Understanding a property of DFAs.

- Consider a unary alphabet $\{a\}$, and DFA $\mathcal{B} = (Q, q_0, \delta, F)$
- For each a^k, the DFA gives rise to a function

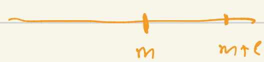
$$f_k^{\mathcal{B}} : Q \longmapsto Q$$

$$\boxed{\begin{array}{l} q_1 \xrightarrow{\;a^k\;} q_1' \\ q_2 \\ \vdots \\ q_n \longrightarrow q_n' \end{array}}$$

- The number of functions from $Q \longmapsto Q$ is finite.
- Therefore, if we look at the sequence:

$$f_1^{B},\ f_2^{B},\ f_3^{B},\ \ldots\ldots.$$

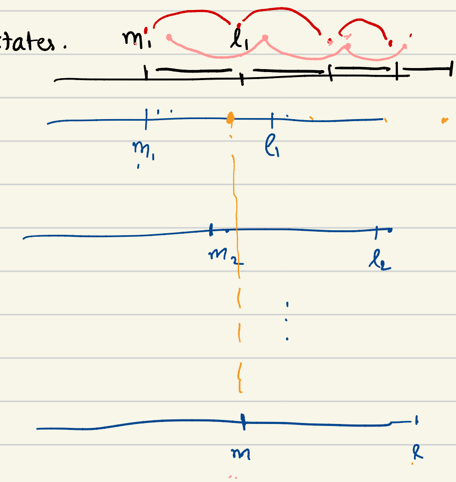
there exist m, ℓ, s.t.



$$\boxed{f_m^{\mathcal{B}} = f_{m+\ell}^{\mathcal{B}}}$$

- Moreover: $f_{m+i}^{B} = f_{m+\ell+i}^{B} \quad \forall i \geq 0$

Consider all DFA with atmost n states.



finitely many

$$\left\{\begin{array}{lll} B_1 & - & m_1, l_1 \\ B_2 & - & m_2, l_2 \\ \vdots & \vdots & \\ B_j & - & m_j, l_j \end{array}\right.$$

- Let $m = \max(m_1, \ldots, m_j)$

$l = l_1 \cdot l_2 \cdot l_3 \cdots l_j$

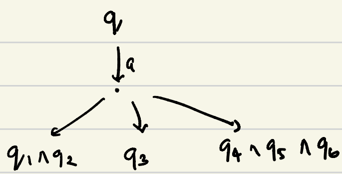
Then for every DFA B with $\leq n$ states, we have:

$$f^{B}_{m+i} = f^{B}_{m+l+i} \qquad \forall i \geq 0$$

Step 2: Translating Step 1 to alternating finite automata.

AFA: (Q, q_0, δ, F)

$$\delta : Q \times \Sigma \longmapsto \mathcal{B}^+(Q)$$



Syntax and semantics similar to ATA: with no guards, no resets

Claim: Every AFA can be converted into an equivalent DFA.

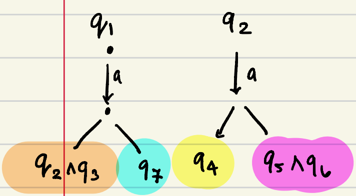
Modified subset construction:

Each node: a set of subsets of Q

$$\{\ \{q_1, q_2\},\ \{q_1, q_3, q_4\},\ \{q_2, q_5\},\ \{q_3\}\ \}$$

$$\big\downarrow a$$

$$?$$



$$\{q_1, q_2\}$$

$$\big\downarrow a$$

$$\{q_2, q_3, q_4\},\ \{q_2, q_3, q_5, q_6\}\ \{q_7, q_4\},\ \{q_7, q_5, q_6\}$$

- Perform the above operation on each set from the set of subsets.

- Node is accepting if there is a subset containing only accepting states.

Theorem: Every AFA with 'n' states can be converted into a DFA with $\leq 2^{2^n}$ states.

Consider unary alphabet $\{a\}$.

An AFA $\mathcal{A}$ with state set Q gives a function :

$$\boxed{f_k^{\mathcal{A}} : 2^{2^Q} \longmapsto 2^{2^Q}}$$

$$\{\{\ \}, \{\ \}, \{\ \} \ldots \{\ \}\} \xrightarrow{a^k} \{\{\ \}, \{\ \}, \{\ \} \ldots \{\ \}\}.$$

$$\vdots$$

$$\{\{\ \}, \{\ \}\ \} \xrightarrow{a^k} \{\{\ \}, \{\ \}, \{\ \}\}.$$

- Similar to the DFA case, let 'm', 'ℓ' be numbers s.t.

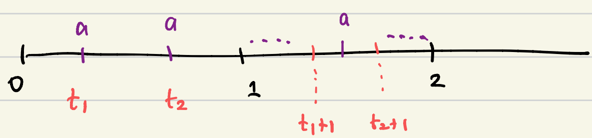
$$f_{m+i}^{\mathcal{A}} = f_{m+\ell+i}^{\mathcal{A}} \quad \forall i \geq 0$$

for all AFA $\mathcal{A}$ with atmost $2n$ states

- Starting from some $\{q\}$, a^{m+i} goes to an accepting node iff $a^{m+\ell+i}$ goes to an accepting node.

Recall:

$$L = \Big\{ (a^k, t_1 t_2 \dots t_k) \;\Big|\; \begin{array}{l} 0 < t_1 < t_2 < 1 \\ 1 < t_3, \dots, t_k < 2 \\ \text{there is exactly one } a \text{ between} \\ \quad t_1 + 1 \text{ and } t_2 + 1 \end{array} \Big\}$$



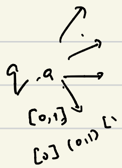
Step 1: Understand some property of DFAs

Step 2: How Step 1 translates to untimed alternating finite automata

Step 3: Any 1-ATA accepting L behaves like an untimed AFA in the interval $(1, 2)$, where clocks are useless.

Step 4: Use Step 1 and 2 in 3 to get a contradiction.

Suppose $\mathcal{A}$ is a 1-ATA with 'n' states accepting 'L'.



- We can assume that every transition is partitioned as:

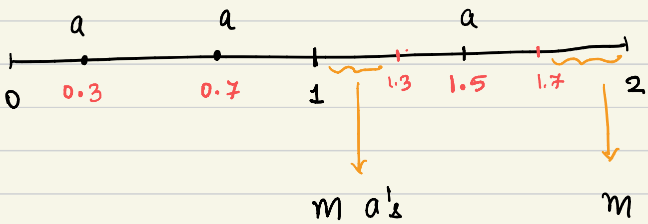
$$x = 0 \;\Big|\; 0 < x < 1 \;\Big|\; x = 1 \;\Big|\; 1 < x < 2 \;\Big|$$ beyond 2 original guard.

- For the moment, let us ignore all transitions with $x = 0$. We will see later why we can do this.

Construct two timed words w_1 and w_2 as follows:

No two 'a's come at the same time-stamp

w_1 :



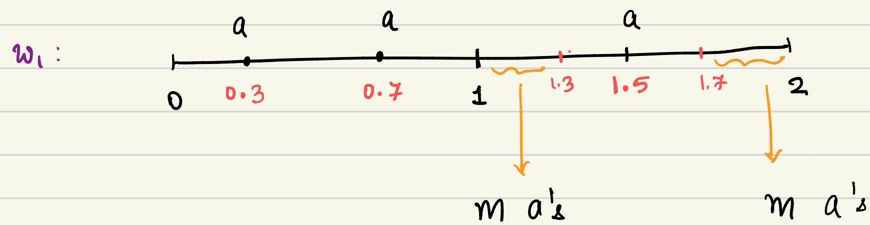
(m, ℓ are as chosen before)

w_2 : On top of w_1, add 'ℓ' a's in the interval $(1.3, 1.7)$, but not at 1.5

$w_1 \in L, \quad w_2 \notin L.$

We will show that if $\mathcal{A}$ accepts w_1, it also accepts w_2 – a contradiction.

w_1 :



(m, l are as chosen before)

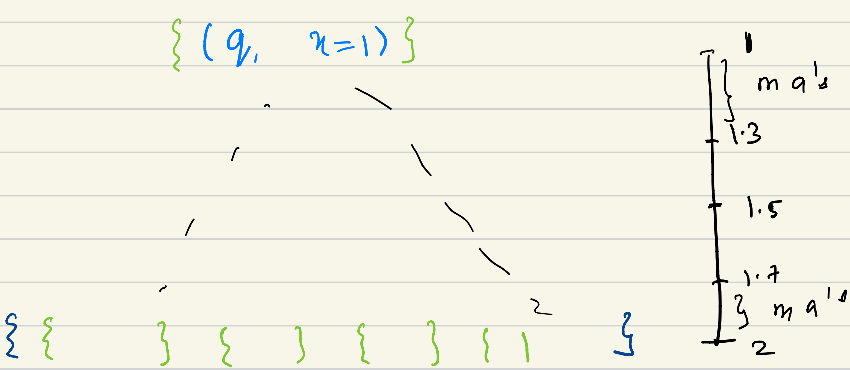
Consider the acceptance game for $\mathcal{A}$ on w_1.

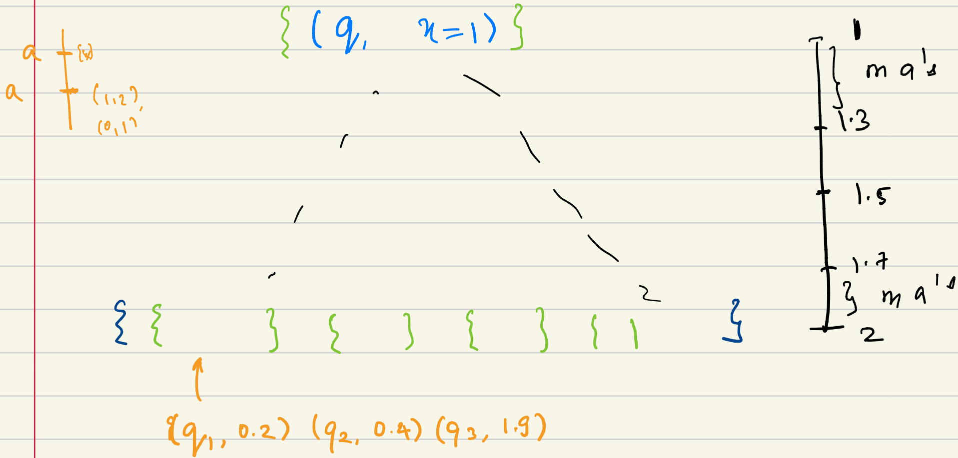
- Let (q, v) be a configuration reached at $t = 1$.

What are the possible values of x at $t = 1$?

$(q_1, x = 1)$ $(q_1, x = 0.7)$ $(q_1, x = 0.3)$

Pick $(q, x = 1)$ and investigate the set of sets of configns. reached from here after reading the entire word.



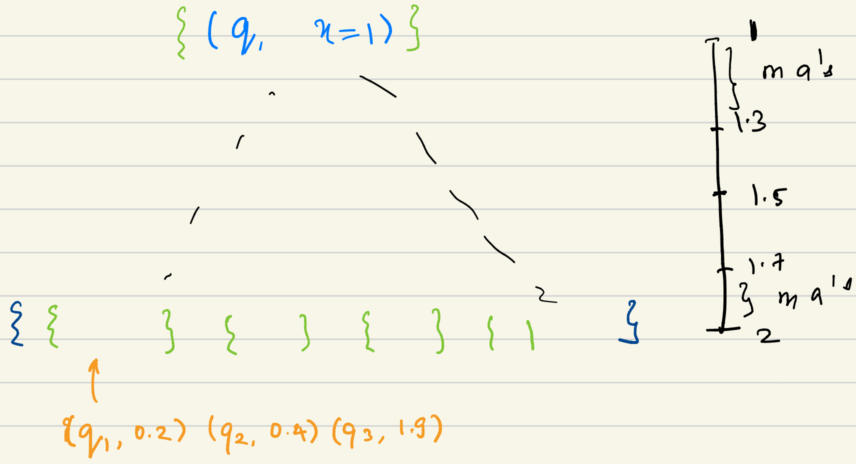


- In $(1, 2)$ transitions with guard $x=0$ are never used.
- In fact, only those transitions with

  either i) $1 < x < 2$

  or ii) $0 < x < 1$

  are used.
- i) is taken until 'x' is reset, (ii) is taken after x is reset.
- Therefore, if we maintain an extra bit $0/1$ in each state to mark whether x has been reset until now, we can recover the behaviour of $\mathcal{A}$ in the interval $(1, 2)$.

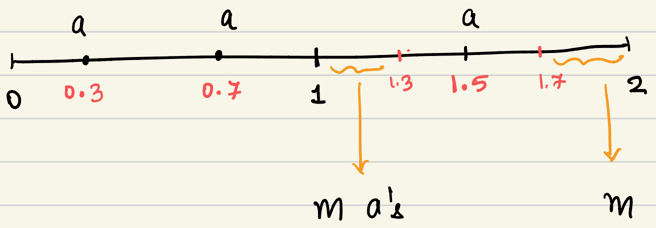


- Therefore, starting from $(q, x=1)$, the rest of the accepting run is identical to the run of an (untimed) AFA with $2n$ states, starting from $(q, 0)$ → to denote not exact.

- From our choice of 'm' and 'ℓ', the same set of sets will be reached by this untimed AFA on the word w_2!

- Hence, from $(q, x=1)$, w_2 will also be accepted.

w_1 :



(m, ℓ are as chosen before)

Let us now focus on $(q, x = 0.7)$ at $t=1$

- Upto $t=1.3$ the word is the same in both w_1 and w_2 and hence the same set of configuration will be reached at $t=1.3$

- Configurations at $t=1.3$ are either $(q, x=1)$ or $(q, x < 0.3)$

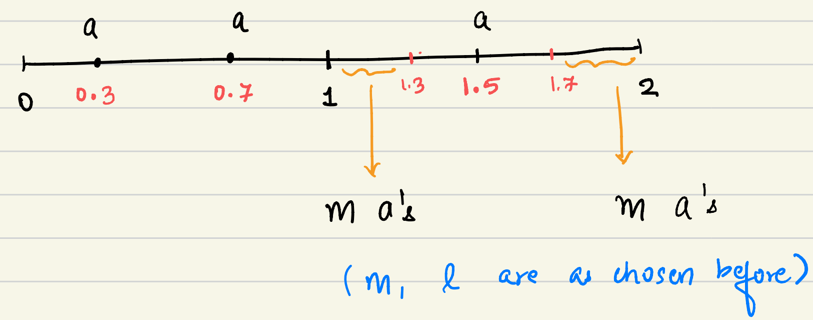
- From $(q, x=1)$, apply same argument as before.

- From $(q, x \leq 0.3)$, only $(0 < x < 1)$ transitions will be taken, so it behaves like an untimed AFA with 'n' states.

- By our choice of 'm' and 'ℓ' the same set of set of states is reached after reading w_1 and w_2.

Hence from $(q, x = 0.7)$ at $t=1$, if w_1 is accepted, w_2 is also accepted.

w_1 :



Finally consider $(q, x = 0.3)$ at $t = 1$.

- upto $t = 1.7$, $\mathcal{A}$ will take only $0 < x < 1$ edges.

- Hence the behaviour is similar to an AFA, and the same set of "state" will be reached for both w_1 and w_2 at $t = 1.7$

  The value of x may be different. However, it will either be $x = 1$ for both words, or some value with $x < 1$ in both.

- . From configurations with $x < 1$ at $t = 1.7$, the actual value remains $0 < x < 1$ for the rest of the word. Hence the true value does not matter.

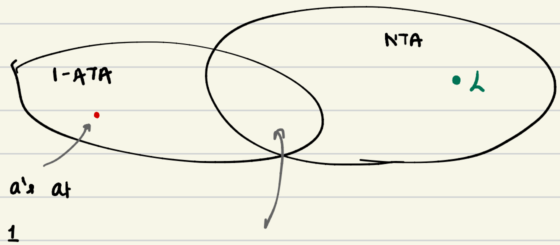
- This shows that the set of set of states reached after both w_1 and w_2 are the same!

If w_1 is accepted by $\mathcal{A}$, w_2 is also accepted by $\mathcal{A}$.

— Contradiction

Summary:

Expressive power of 1-ATA Vs many clock NTA



no two a's at distance 1

Contains 1-clock NTAs

and some 2-clock NTAs too.