TIMED AUTOMATA

LECTURE 19

Diagonal constraints:

$$\xrightarrow{x - y \sim c}$$

- Diagonal constraints can be eliminated.

For d-timed automaton $\mathcal{A}$ (with possibly diag. constr.) $\longrightarrow$ $\exists$ a timed aut. (diagonal-free) $\mathcal{B}$. s.t.

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

Today: There exists a family of languages $\{L_n\}_{n \geq 1}$ s.t.

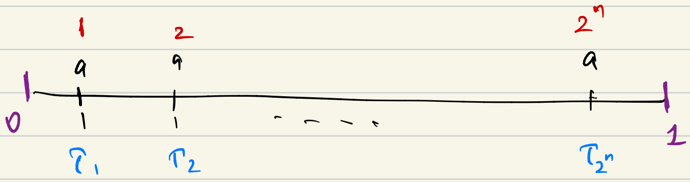
- for each L_n there exists a d-timed automaton of size $O(n)$
- every diagonal free TA that accepts L_n has at least size $O(2^n)$

d-timed automata are exponentially more succinct.

Reference: Bouyer & Chevalier '05

Language L_n:

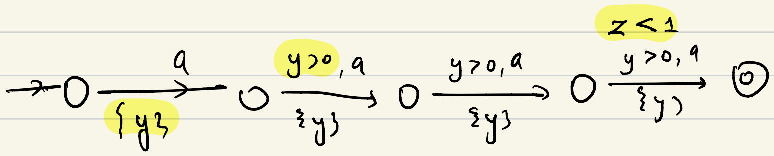
$$L_n \quad \{ (a^{2^n}, \tau) \mid 0 < \tau_1 < \tau_2 < \dots < \tau_{2^n} < 1 \}$$



$$L_1 = \{ (a^2, \tau) \mid 0 < \tau_1 < \tau_2 < 1 \}$$

$$L_2 = \{ (a^4, \tau) \mid 0 < \tau_1 < \dots < \tau_4 < 1 \}$$

Naive T.A.:

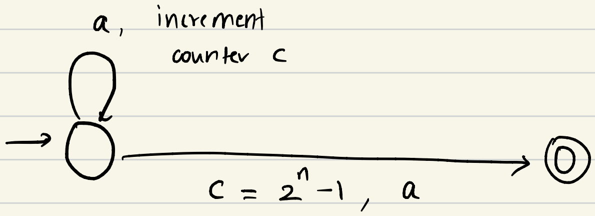


↳ 4 states

For L_n this naïve construction would give 2^n states.

Succinct d-timed automaton for L_n:

Idea:



Add clocks 'y' and 'z' as before to ensure $\tau_i < \tau_{i+1}$ and $\tau_{2^n} < 1$

n-bit counter: $b_1 \; b_2 \; \ldots \; b_n$

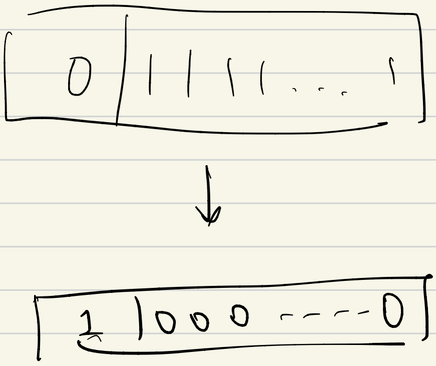
$b_n \; b_{n-1} \; \ldots \; b_1$

most significant — least significant

Ex:

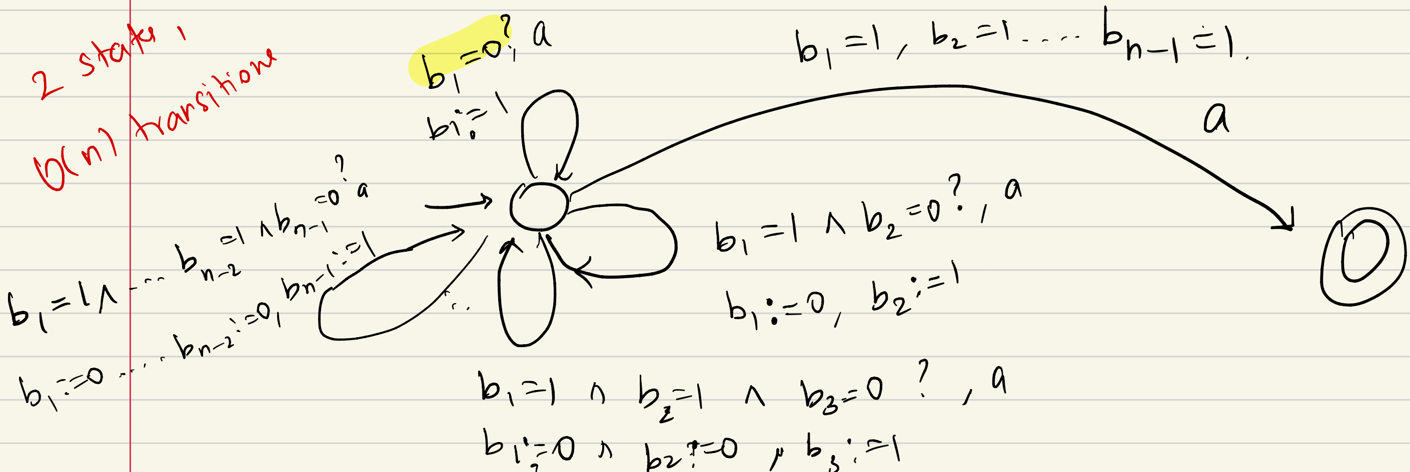
1 1 0 1 0
$b_5 \; b_4 \; b_3 \; b_2 \; b_1$

0	0	0	0	0
0	0	0	0	1
0	0	0	1	0
0	0	0	1	1
0	0	1	0	0
0	0	1	0	1
0	0	1	1	0
⋮				



find first 0 from the right.

2 states, O(n) transitions



Simulating bits b_i using clocks:

b_i : x_i, x_i'

$x_i - x_i' = 0$ encodes $b_i = 0$

$x_i - x_i' > 0$ encodes $b_i = 1$

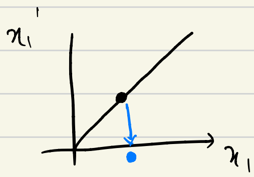
The key observation is that the difference between two clocks is invariant under time elapse.

- So, $x_i - x_i'$ registers a value

- This value is stored until there is a reset of one of the clocks.

How do we change the value of $x_i - x_i'$?

$x_1 - x_1' = 0$



$\downarrow$

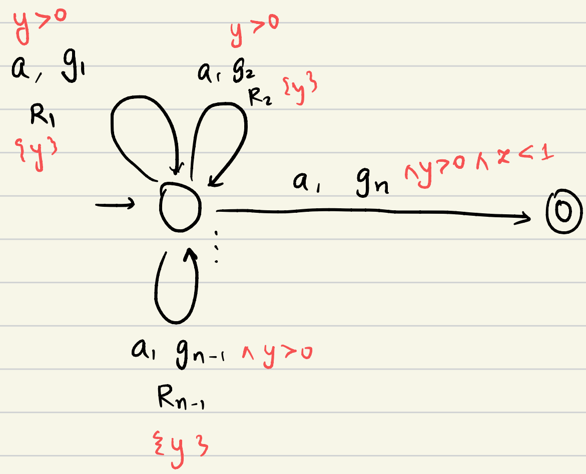
Suppose after a non-zero time elapse, x_1' is reset.

then value of $x_1 - x_1'$ becomes > 0.

A transition:

$$\xrightarrow[b_1 := 0,\ b_2 := 1]{b_1 = 1 \wedge b_2 = 0?} \quad \longrightarrow \quad \xrightarrow[\{x_1,\ x_1'\} \qquad \{x_2'\}]{x_1 - x_1' > 0 \ \wedge \ x_2 - x_2' = 0}$$

d- timed automaton for L_n:



$$g_i = \left(\bigwedge_{j=1}^{i-1} x_j - x_j' > 0 \right) \wedge x_i - x_i' = 0$$

$$R_i = \bigcup_{j=1}^{i-1} \{ x_j, x_j' \} \cup \{ x_i' \}$$

A_n:

↳ accepts L_n.

Size of A_n: States = 2

Transitions = $O(n)$

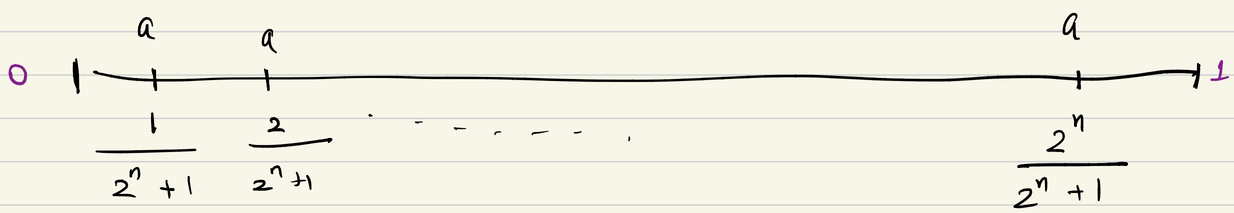
Constants used are 0, 1

Every diagonal-free TA for L_n is big:

$$L_n = \{ (a^{2^n}, \tau) \mid 0 < \tau_1 < \tau_2 < \cdots < \tau_{2^n} < 1 \}$$

Suppose there exists a diagonal-free T.A. B_n with strictly less than 2^n states that accepts L_n.

Consider the word w:



$w \in L_n$ and hence $w \in \mathcal{L}(B_n)$

B_n has an accepting run on w.

$$(q_0, v_0) \xrightarrow{\delta,\ a} (q_1, v_1) \xrightarrow{\delta,\ a} (q_2, v_2) \longrightarrow \cdots \xrightarrow{\delta,\ a} (q_{2^n}, v_{2^n})$$

$$\delta = \frac{1}{2^n+1}$$

Observation:

$$(q_i, v_i) \xrightarrow{\delta,\ a} (q_{i+1}, v_{i+1}) \qquad (q_i, \phi_i, R_i, q_{i+1})$$

In $v_i + \delta$, every clock is $(0, 1)$

This means that the guard ϕ_i in the transition

$(q_i, \phi_i, R_i, q_{i+1})$ allows every valuation in

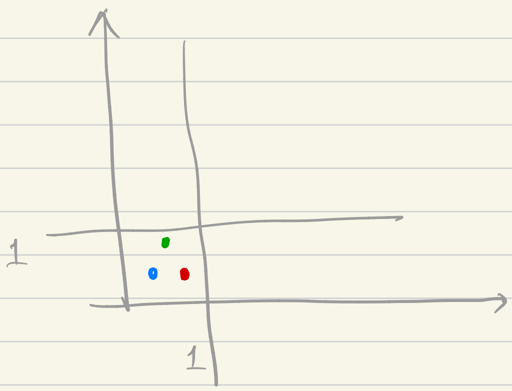
the hypercube $(0,1)^{|X|}$, X is no. of clocks in B_n

$(q_i, \phi_i, R_i, q_{i+1})$

Observation: $(q_i, v_i) \xrightarrow{\delta,\ a} (q_{i+1}, v_{i+1})$

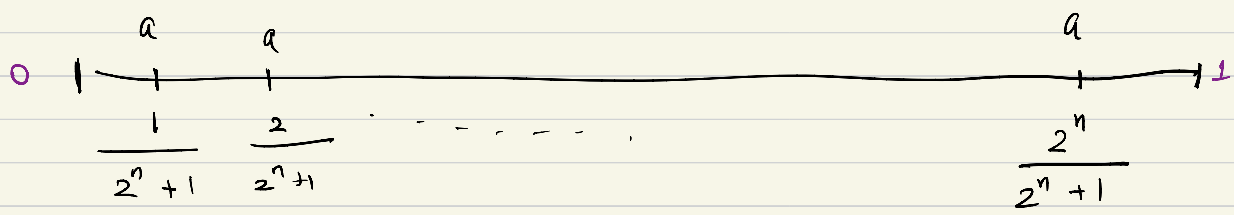
In $v_i + \delta$, every clock is $(0,1)$

This means that the guard ϕ_i in the transition $(q_i, \phi_i, R_i, q_{i+1})$ allows every valuation in the hypercube $(0,1)^{|X|}$, X is no. of clocks in $\mathcal{B}_n$



a diagonal-free guard cannot distinguish between the red, blue and green points.

- as constants used in guards are natural no.s.



$w \in L_n$ and hence $w \in \mathcal{L}(B_n)$

B_n has an accepting run on w.

$$\rho : (q_0, v_0) \xrightarrow{\delta,\ a} (q_1, v_1) \xrightarrow{\delta,\ a} (q_2, v_2) \longrightarrow \cdots \xrightarrow{\delta,\ a} (q_{2^n}, v_{2^n})$$

$$\delta = \frac{1}{2^n+1}$$

Since there are $< 2^n$ states, there are states q_i, q_j in run ρ s.t. $q_i = q_j$

$\bullet \rightsquigarrow (q_i, v_i) \qquad (q_j, v_j) \rightsquigarrow$

Consider the word formed by cutting the part of the run between i and j.

Word w':

$$w' = \left(a, \frac{1}{2^n+1}\right) \dots \left(a, \frac{i}{2^n+1}\right) \left(a, \frac{j+1}{2^n+1}\right) \left(a, \frac{j+2}{2^n+1}\right) \dots \left(a, \frac{2^n}{2^n+1}\right)$$

Claim: $\mathcal{B}_n$ accepts w' too

$$\rho': \underbrace{(q_0, v_0) \rightsquigarrow (q_i, v_i)}_{\text{same as } \rho} \xrightarrow{\delta'} (q_{j+1}, v'_{j+1}) \xrightarrow{\delta, a} (q_{j+2}, v'_{j+2}) \dots (q_{2^n}, v'_{2^n})$$

$\delta' = \frac{(j+1-i)}{2^n+1}$

$v_i + \delta'$

↳ Every clock in $x'_k + \delta \in (0,1)$ for $k \in \{j+1 \dots 2^n-1\}$

- As the guards in the corresponding transitions allow every valuation in $(0,1)^{|X|}$, ρ' is a valid run.

→ Since ρ was an accepting run, q_{2^n} is accepting. Hence ρ' is accepting. $w' \in \mathcal{L}(\mathcal{B}_n)$ → contradiction

- d-timed automata are exponentially more concise.

Summary of diagonal constraints:

- Equal expressive power
- Exponentially more concise.