TIMED AUTOMATA

LECTURE 5

Neighbourhood equivalence:

$v \simeq_{nbd} v'$ if

1. $\lfloor v(x) \rfloor = \lfloor v'(x) \rfloor \quad \forall x$
2. $\{v(x)\} = 0$ iff $\{v'(x)\} = 0 \quad \forall x$
3. $\{v(x)\} \leq \{v(y)\}$ iff $\{v'(x)\} \leq \{v'(y)\} \quad \forall x, y$

Region equivalence:

$M \in \mathbb{N}$, $v \simeq_M v'$ if

1. $Bounded(v) = Bounded(v')$
2. $v\restriction_{Bounded(v)} \simeq_{nbd} v'\restriction_{Bounded(v')}$

$Bounded(v) = \{ x \mid v(x) \leq M \}$

Direct formulation of region equivalence:

1. $v(x) > M \iff v'(x) > M \quad \forall x$

2. For all x, y s.t. $v(x) \leq M$, $v(y) \leq M$:

   2.1. $\lfloor v(x) \rfloor = \lfloor v'(x) \rfloor$

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

   2.3. $\{v(x)\} \leq \{v(y)\}$ iff $\{v'(x)\} \leq \{v'(y)\}$

USING A DIFFERENT CONSTANT FOR EACH CLOCK:

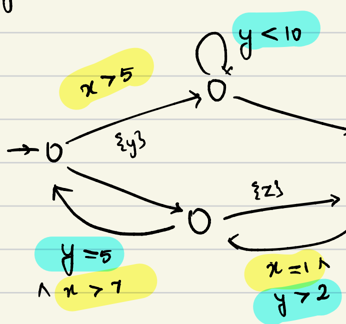
$M : X \longrightarrow \mathbb{N}$ a bounds function

- associates a constant to each clock

Eg: Suppose $X = \{x, y, z\}$

$M(x) = 5$ $M(y) = 2$ $M(z) = 10$ is a bounds function

Intuition: Associate to each clock x the largest constant appearing in a guard of the form $x \sim c$ in the given automaton.



$M(x) = 7$
$M(y) = 10$
$M(z) = 20$

Modified region equivalence:

$Bounded(v) = \{x \mid v(x) \leq M\}$ → $Bounded(v) = \{x \mid v(x) \leq M(x)\}$

$Bounded(v') = \{x \mid v'(x) \leq M\}$ → $Bounded(v') = \{x \mid v'(x) \leq M(x)\}$

$v \simeq_M v'$ if

1. $Bounded(v) = Bounded(v')$ → same
2. $v\big|_{Bounded(v)} \simeq_{nbd} v'\big|_{Bounded(v')}$ → definition of region automaton same.

Direct formulation of region equivalence:

1. $v(x) > M \iff v'(x) > M$

2. For all x, y s.t. $v(x) \leq M$, $v(y) \leq M$:

   2.1. $\lfloor v(x) \rfloor = \lfloor v'(x) \rfloor$

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

   2.3. $\{v(x)\} \leq \{v(y)\}$ iff $\{v'(x)\} \leq \{v'(y)\}$

Direct formulation of region equivalence with M-bounds:

1. $v(x) > M_x \iff v'(x) > M_x$

2. For all x, y s.t. $v(x) \leq M_x$, $v(y) \leq M_y$:

   2.1. $\lfloor v(x) \rfloor = \lfloor v'(x) \rfloor$

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

   2.3. $\{v(x)\} \leq \{v(y)\}$ iff $\{v'(x)\} \leq \{v'(y)\}$

↳ Alur, Dill '94

A theory of timed automata.

Size of region automaton:

A region is specified as follows:

- 1. for each clock x, One constraint from the set:

$$\{x = c \mid c \in \{0,1,\ldots M_x\}\} \cup \{c-1 < x < c \mid c \in \{1,2,\ldots M_x\}\} \cup \{x > M_x\}$$

- 2. for every pair of clocks x, y s.t. the constraints for x, y are open unit intervals, say $c-1 < x < c$ and $d-1 < y < d$,

there is one constraint from:

$$\{x\} < \{y\}, \quad \{x\} = \{y\}, \quad \{x\} > \{y\}$$

$\therefore$ No. of regions is $\leq \prod_{x \in X} (2M_x + 1) \cdot 2^{|X|} \cdot |X|!$

- $\prod_{x \in X} (2M_x + 1)$ → constant
- $2^{|X|}$ → clocks that have fractional value 0
- $|X|!$ → ordering bet. bdd. and non-zero fractional clocks

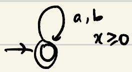
Next:

Solutions to Problem Sheets 1 & 3

3/10/2021

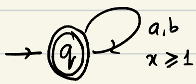
Solutions to Problem Sheet 1:

1. A timed automaton that accepts all words.



contains a trivial guard that is always true.

2.



rejects (a, 0.5)

---

3. (0.6, a) (1.3, a)

4.

a) Yes. $q_0, \langle 0,0,0\rangle \xrightarrow{0.5,\,a} q_1, \langle 0, 0.5, 0.5\rangle \xrightarrow{0.5,\,b} q_2, \langle 0.5, 0, 1\rangle \xrightarrow{0.5,\,c} q_0, \langle 1, 0.5, 0\rangle$

$\xrightarrow{0.3,\,a} q_1, \langle 0, 0.8, 0.3\rangle \xrightarrow{0.1,\,b} q_2, \langle 0.1, 0, 0.4\rangle \xrightarrow{1.1,\,c} q_0, \langle 1.2, 1.1, 0\rangle$

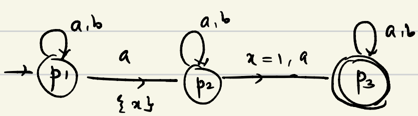
b) Assume $k \geq 1$.

Consider:

	1			2				k − 1		
	a	b	c	a	b	c	. . . . .	a	b	c
	$\frac{1}{3k}$	$\frac{2}{3k}$	$\frac{1}{k}$	$\frac{4}{3k}$	$\frac{5}{3k}$	$\frac{1}{2k}$		$\frac{3(k-2)+1}{3k}$	$\frac{3(k-2)+2}{3k}$	$\frac{3(k-1)}{3k}$

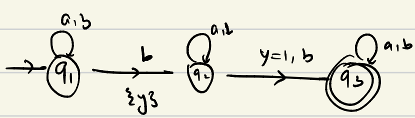
---

5. L_a =



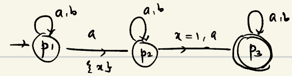
∃ 2 a's at distance 1 apart.

L_b =

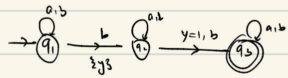


∃ 2 a's at distance 1 apart

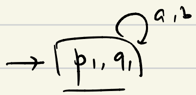
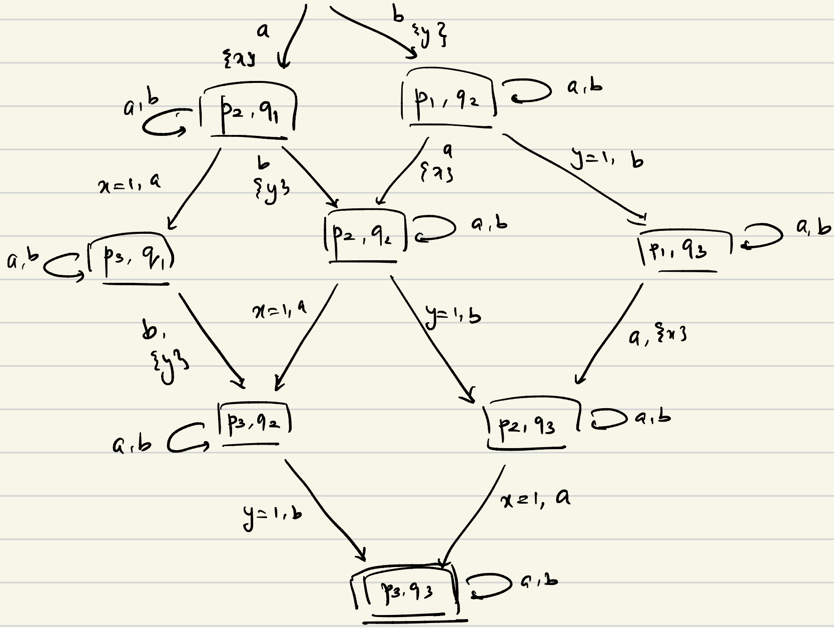
$L_a =$



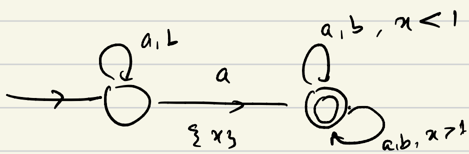
$L_b =$

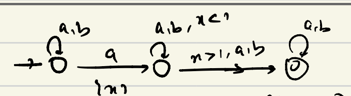


$L_a \cap L_b$:

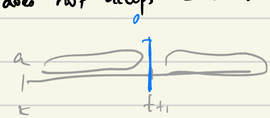



6. a)

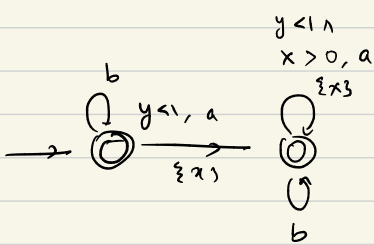




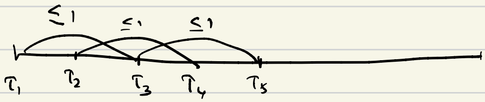
does not accept (a, 0.5)

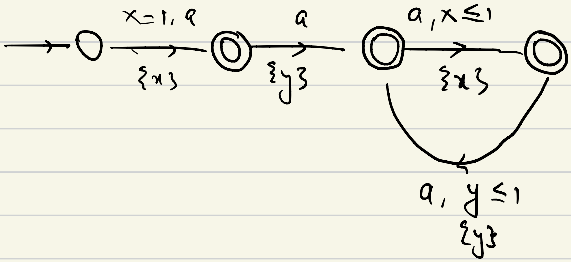


b)



c)



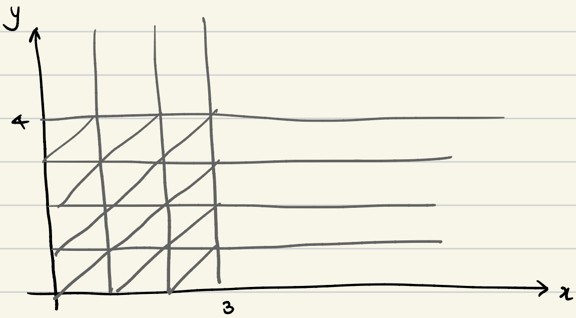


3/10/2021

Solutions to Problem Sheet 3:

1. $v_1 \simeq_M v_4 \simeq_M v_3$ $\qquad$ $v_2 \simeq_M v_6 \simeq_M v_5$

2.



3.

$x=0,$	$y=0,$	$z=0$	$x=0,$	$0<y<1,$	$z=0$
	"	$0<z<1$	"		$0<z<1,$ $\{y\}<\{z\}$
	"	$z=1$	"		$,$ $\{y\}=\{z\}$
	"	$1<z<2$	"		$,$ $\{y\}>\{z\}$
	"	$z=2$	"		$z=1$
	"	$z>2$	"		$1<z<1,$ $\{y\}<\{z\}$
			"		$\{y\}=\{z\}$
			"		$\{y\}>\{z\}$
			"		$z=2,$
			"		$z>2$

... and so on!!

4. A region is specified as follows:

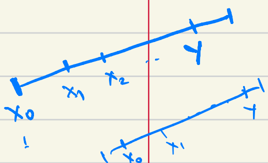
— 1. for each clock x, one constraint from the set:

$$\{x = c \mid c \in \{0, 1, \ldots M_x\}\} \cup \{c-1 < x < c \mid c \in \{1, 2, \ldots M_x\}\} \cup \{x > M_x\}$$

— 2. for every pair of clocks x, y s.t. the constraints for x, y are open unit intervals, say $c-1 < x < c$ and $d-1 < y < d$,

there is one constraint from:

$$\{x\} < \{y\}, \quad \{x\} = \{y\}, \quad \{x\} > \{y\}$$



$x = c$

$c < x < c+1$

$x = M$

$x > M$

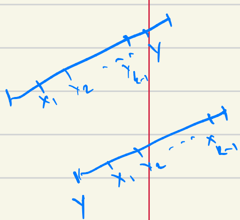
Computing the immediate time successor of a region R:

Let $X_0 = \{x \mid x = c,\ 0 \le c \le M_x$ (for some) is the constraint in R$\}$

X_0 is the set of bounded clocks whose fractional part is 0.

Let $Y = \{x \mid c-1 < x < c$ for some $1 \le c \le M$ is the constraint in R and for every y that is bounded we have $\{y\} \le \{x\}\}$.

Y is the set of bounded clocks that have the highest non-zero fractional part.



Let R' be the immediate time successor of R:

Algorithm: 1. If X_0 is non-empty

R' associate:

$c < x < c+1$	if	$x = c$ in R, and $c < M$
$x > M$	if	$x = M$ in R
$c-1 < x < c$	if	$c-1 < x < c$ in R
$x > M$	if	$x > M$ in R
$\{x\} = \{y\}$	if	$x = c$, $y = d$ in R, $c < M$, $d < M$
$\{x\} < \{y\}$	if	$x = c$, $d-1 < y < d$ in R
$\{x\} \sim \{y\}$	where	$\sim$ is the same comparison operator as in R when $c-1 < x < c$, $d-1 < y < d$ in R.

2. If X_0 is empty, Y is non-empty

R' associate: $x = c$ if $c-1 < x < c$ in R, and $x \in Y$, $c \leq M$

~~$\{x\} < \{y\}$ if $x \in Y$, $d-1 < y < d$ in R and $y \notin Y$~~

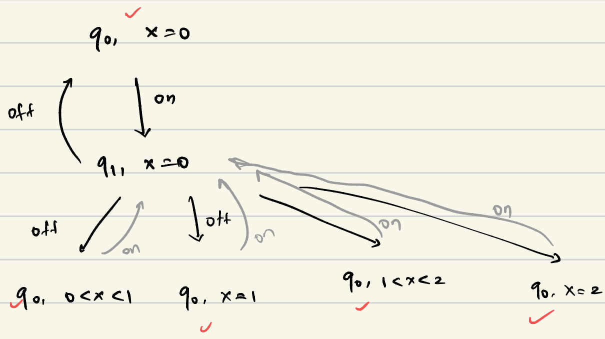
same constraints as in R for all other clocks outside Y.

3. If X_0 is empty and Y is empty

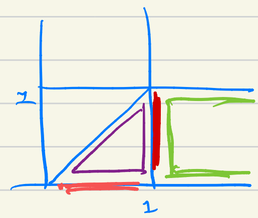
this means all clocks are unbounded, $\therefore R' = R$

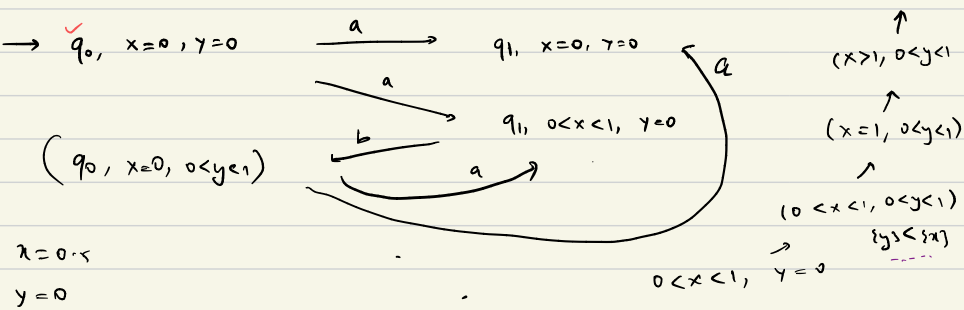
To find all time successors, one can iteratively apply the above algorithms to get the immediate successors, until the unbounded region $x > M_x \ \forall x$ is reached.

5.



✓ : acc. states.





$x = 0.5$

$y = 0$

---

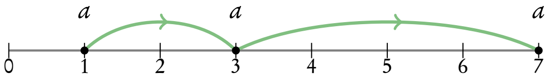
6. Yes. It is a region since the region equivalence is satisfied for $\{x, y\}$.

Next:

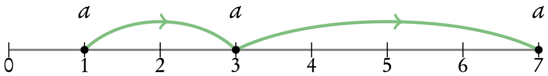
New problems

Is the following language timed regular?

$$L_6 := \{\ (\ a^k, \tau\ )\ |\ \tau_i \text{ is some integer for each } i\}$$



$$L_6 := \{ (a^k, \tau) \mid \tau_i \text{ is some integer for each } i\}$$



Claim: **No timed automaton** can accept L_6

Step 1: *Suppose* $L_6 = \mathcal{L}(A)$

Let c_{max} be the maximum constant appearing in a guard of A

Step 1: *Suppose* $L_6 = \mathcal{L}(A)$

We consider only integral constants.

Let c_{max} be the maximum constant appearing in a guard of A

Step 2: For a clock x,

$x = c_{max} + 1$ and $x = c_{max} + 1.1$

satisfy the same guards

Step 1: *Suppose* $L_6 = \mathcal{L}(A)$

Let c_{max} be the maximum constant appearing in a guard of A

Step 2: For a clock x,

$$x = \lceil c_{max} \rceil + 1 \;\text{ and }\; x = \lceil c_{max} \rceil + 1.1$$

satisfy the same guards

Step 3: $(a; \lceil c_{max} \rceil + 1) \in L_6$ and so A has an accepting run

$$(q_0, v_0) \xrightarrow{\delta = \lceil c_{max} \rceil + 1} (q_0, v_0 + \delta) \xrightarrow{a} (q_F, v_F)$$

Step 1: *Suppose* $L_6 = \mathcal{L}(A)$

Let c_{max} be the maximum constant appearing in a guard of A

Step 2: For a clock x,

$$x = c_{max} + 1 \text{ and } x = c_{max} + 1.1$$

satisfy the same guards

Step 3: $(a; c_{max} + 1) \in L_6$ and so A has an accepting run

$$(q_0, v_0) \xrightarrow{\delta = c_{max} + 1} (q_0, v_0 + \delta) \xrightarrow{a} (q_F, v_F)$$

Step 4: By Step 2, the following is an accepting run

$$(q_0, v_0) \xrightarrow{\delta' = c_{max} + 1.1} (q_0, v_0 + \delta') \xrightarrow{a} (q_F, v'_F)$$

Step 1: *Suppose* $L_6 = \mathcal{L}(A)$

Let c_{max} be the maximum constant appearing in a guard of A

Step 2: For a clock x,

$$x = c_{max} + 1 \text{ and } x = c_{max} + 1.1$$

satisfy the same guards

Step 3: $(a;\ c_{max} + 1) \in L_6$ and so A has an accepting run

$$(q_0, v_0) \xrightarrow{\delta = c_{max} + 1} (q_0, v_0 + \delta) \xrightarrow{a} (q_F, v_F)$$

Step 4: By Step 2, the following is an accepting run

$$(q_0, v_0) \xrightarrow{\delta' = c_{max} + 1.1} (q_0, v_0 + \delta') \xrightarrow{a} (q_F, v'_F)$$

Hence $(a;\ c_{max} + 1.1) \in \mathcal{L}(A) \neq L_6$

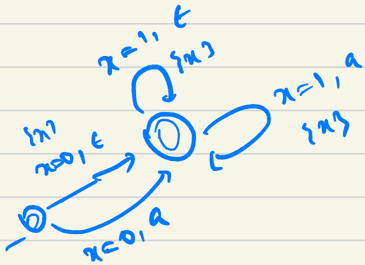
Therefore **no timed automaton** can accept L_6 □

- L_6 can be accepted when ε-transitions are allowed.

<u>Thm.</u> ε-transitions add more expressivity if they contain clock resets.

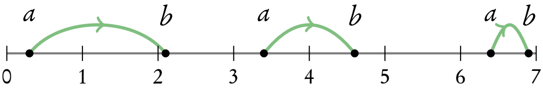
- ε-transitions that have no resets can be eliminated

  ↳ Bérard et al., 1998



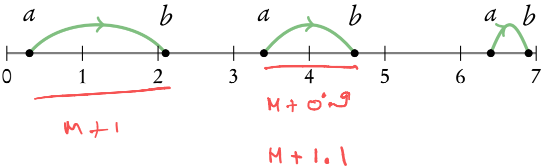
$$L_7 = \{ \; (\; (ab)^k, \tau \;) \mid \tau_{2i+2} - \tau_{2i+1} < \tau_{2i} - \tau_{2i-1} \text{ for each } i \geq 1 \}$$

Converging *ab* distances



$$L_7 = \{ \, (\, (ab)^k, \tau \,) \mid \tau_{2i+2} - \tau_{2i+1} < \tau_{2i} - \tau_{2i-1} \text{ for each } i \geq 1 \}$$

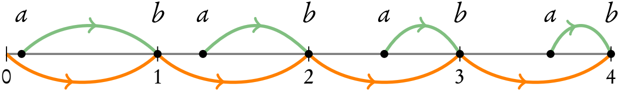
Converging *ab* distances



Exercise: Prove that **no timed automaton** can accept L_7

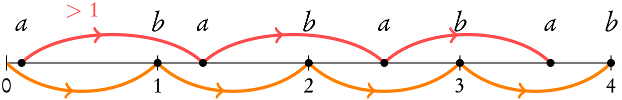
$$L_7 = \{ \ (\ (ab)^k, \tau \ ) \mid \tau_{2i} = i \ \text{ and } \ \tau_{2i+2} - \tau_{2i+1} < \tau_{2i} - \tau_{2i-1} \}$$

Pivoted converging *ab* distances



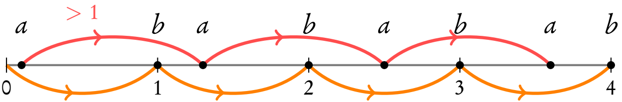
$$L_7 = \{ \, (\, (ab)^k, \tau \,) \mid \tau_{2i} = i \;\text{ and }\; \tau_{2i+2} - \tau_{2i+1} < \tau_{2i} - \tau_{2i-1} \}$$

Pivoted converging *ab* distances



$$L_7 = \{ \ (\ (ab)^k, \tau \ ) \mid \tau_{2i} = i \ \text{ and } \ \tau_{2i+2} - \tau_{2i+1} < \tau_{2i} - \tau_{2i-1} \}$$

Pivoted converging *ab* distances

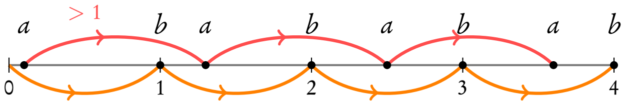


$$\tau_{2i+2} - \tau_{2i+1} < \tau_{2i} - \tau_{2i-1} \iff \tau_{2i+2} - \tau_{2i} < \tau_{2i+1} - \tau_{2i-1}$$

$$\iff 1 < \tau_{2i+1} - \tau_{2i-1}$$

$$L_7 = \{\ (\ (ab)^k, \tau\ )\ |\ \tau_{2i} = i \ \text{ and } \ \tau_{2i+2} - \tau_{2i+1} < \tau_{2i} - \tau_{2i-1}\}$$

Pivoted converging *ab* distances



$$\tau_{2i+2} - \tau_{2i+1} < \tau_{2i} - \tau_{2i-1} \iff \tau_{2i+2} - \tau_{2i} < \tau_{2i+1} - \tau_{2i-1}$$
$$\iff 1 < \tau_{2i+1} - \tau_{2i-1}$$