TIMED AUTOMATA

LECTURE 22

TODAY'S LECTURE

- Reachability problem for Updatable Timed Automata

Semantics of UTA:

When does UTA $\mathcal{A}$ accept a timed word $(a_1, t_1)(a_2, t_2) \cdots (a_n, t_n)$?

Valuations: $v : X \longmapsto \mathbb{R}_{\geq 0}$

Operations on valuations: $v + \delta$

$up(v)$ ← New operation

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

up is:

$x := x + 2$
$y := z - 5$
$z := x$

$$v_1 = \begin{array}{c} x \\ y \\ z \end{array} \begin{bmatrix} 12 \\ 2 \\ 7.2 \end{bmatrix} \qquad up(v_1) = \begin{bmatrix} 14 \\ 2.2 \\ 12 \end{bmatrix}$$

$$v_2 = \begin{bmatrix} 0 \\ 3 \\ 2.5 \end{bmatrix}$$

$up(v_2)$ not defined since $y := z - 5$ results in a negative value

---

$$up(v)(x) = \begin{cases} v(y) + d & \text{if } \quad x := y + d \text{ and } v(y) + d \geq 0 \\ c & \text{if } \quad x := c \,, \; c \geq 0 \end{cases}$$

$up(v)$ is undefined otherwise

Run of a UTA on a timed word:

$w: \quad (a_1, t_1) \; (a_2, t_2) \; \ldots \; (a_n, t_n)$

Run: $\quad (q_0, v_0) \longrightarrow (q_1, v_1) \longrightarrow \ldots \longrightarrow (q_n, t_n)$

if: $\quad \exists \quad q_i \xrightarrow[up_i]{a_i, g_i} q_{i+1}$

s.t. $\quad v_i + (\overbrace{t_{i+1} - t_i}^{\delta_i}) \models g_i$

$up_i(v_i + \delta_i)$ is defined

$v_{i+1} = up_i(v_i + \delta_i)$

Accepting run: q_n is accepting

Emptiness problem:

- Given UTA $\mathcal{A}$, is language of $\mathcal{A}$ empty?

Theorem: Emptiness problem is undecidable for UTA.

Proof of undecidability:

Reducing emptiness problem of 2-counter machines.

2-Counter Machines

$$(Q, Q_0, \Sigma, \{c, d\}, \Delta)$$

Counters

Operations on counters:
1) increment $c{+}{+}$, $d{+}{+}$
2) decrement $c{-}{-}$, $d{-}{-}$
3) zero test $c==0$, $d==0$?

Ex. of a transition: $q \xrightarrow[c{+}{+}]{a,\ d==0} q'$

- Counter values are always ≥ 0

- A transition with a decrement $c{-}{-}$ can be taken only when $c \geq 1$

Simulating a 2-counter machine using a UTA:

Run of the counter machine:

$(q_0, 0, 0) \longrightarrow (q_1, 1, 0) \longrightarrow \cdots \rightarrow (q_i, c_i, d_i) \rightarrow \cdots$

2-Counter machine $\mathcal{A}$ $\longrightarrow$ 3-clock UTA $\mathcal{B}$

clocks $\{x, y, z\}$

$q \xrightarrow[c++]{} q' \quad \rightarrow \quad q \xrightarrow[x := x+1]{z = 0\,?} q'$

$q \xrightarrow[d--]{} q' \quad \rightarrow \quad q \xrightarrow[y := y-1]{z = 0?} q'$

$q \xrightarrow{c == 0} q' \quad \rightarrow \quad q \xrightarrow{z = 0 \wedge x = 0\,?} q'$

- There is zero time elapse in UTA $\mathcal{B}$, ensured by $z = 0$.

  Clock x gives the value of counter c,
  Clock y counter d

- For every run of 2-counter machine; there is a zero-time run of UTA:

$(q_0, x=0, y=0, z=0) \longrightarrow (q_1, x=c_1, y=d_1, z=0) \rightarrow \cdots\cdots$

Decidable subclasses:

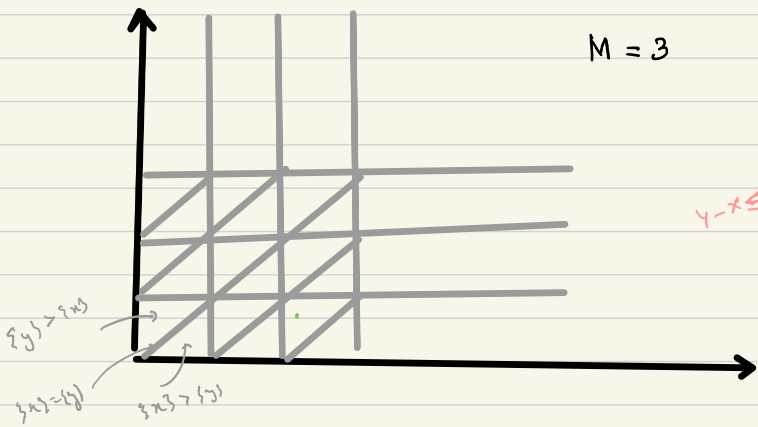
General idea to show decidability : Region automaton

Recall region equivalence: for diagonal-free

$v \equiv_M v'$ if 1) $\lfloor v(x) \rfloor = \lfloor v'(x) \rfloor$ or $v(x),\ v'(x) > M$

2) $\{v(x)\} = 0$ iff $\{v'(x)\} = 0$ $\quad \forall x : v(x) \leq M$

3) $\{v(x)\} \leq \{v(y)\} \iff \{v'(x)\} \leq \{v'(y)\}$ $\quad \forall x, y : v(x), v(y) \leq M$



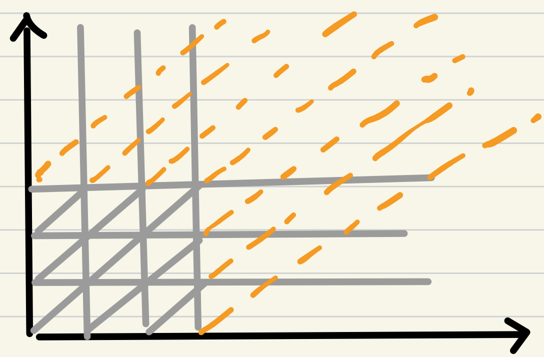
$M = 3$

$y - x \leq 2$

$v \equiv_M v'$ does not work in the presence of diagonal constraints in guards. Add the following conditions in the presence of diagonals:

$$\boxed{\begin{array}{ll} 4) & \lfloor v(x) - v(y) \rfloor = \lfloor v'(x) - v'(y) \rfloor \text{ or} \\ & \quad v(x) - v(y),\ v'(x) - v'(y) \text{ are both } > M \\ 5) & \{v(x) - v(y)\} = 0 \iff \{v'(x) - v(y)\} = 0 \\ & \quad \forall x, y \text{ s.t. } v(x) - v(y) \leq M \end{array}}$$

$M = 3$



orange lines in the presence of diagonals

- Call the equivalence given by the 5 conditions as $v \equiv_M^d v'$

The region equivalences $v \equiv_M^d v'$ and $v \equiv_M v'$ satisfy the following conditions:

Lemma 1: $v \equiv v' \implies \forall \delta \geq 0 \ \exists \delta' \geq 0$ s.t. $v + \delta \equiv v' + \delta'$

Lemma 2: $v \equiv_M v' \implies v$ and v' satisfy the same set of diagonal-free guards having constant $\leq M$

$v \equiv_M^d v' \implies v$ and v' satisfy the same set of diagonal-free and diagonal guards having constant $\leq M$

Lemma 3: $v \equiv v' \implies [R]v \equiv [R]v'$

Reset of R

We now want the region-equivalence to work when resets are replaced with updates.

Goal: For what subclasses of updates will the region equivalence work?

Subclass 1: $x := c$, $x := y$, diagonal-free guards

Let M be max constant occurring among all guards in the automaton.

Problem: Show that region-equivalence $\equiv_M$ satisfies Lemma 3 with resets replaced with updates of the above form.

$$v \equiv_M v' \implies up(v) \equiv_M up(v')$$

for all updates of the form $x := c$, $x := y$ $x, y \in X$, $c \geq 0$

Proof: $v \equiv_M v'$ if
1) $\lfloor v(x) \rfloor = \lfloor v'(x) \rfloor$ or $v(x), v'(x) > M$
2) $\{v(x)\} = 0 \iff \{v'(x)\} = 0$ $\forall x : v(x) \leq M$
3) $\{v(x)\} \leq \{v(y)\} \iff \{v'(x)\} \leq \{v'(y)\}$ $\forall x, y$ $v(x), v(y) \leq M$

$up(v) \equiv_M up(v')$

Condition 1: $up(v)(x)$: if x was not updated, then (1) is true since $v \equiv_M v'$

Otherwise $up(v)(x) = up(v')(x) = c$ (done)

$up(v)(x) = v(y)$ $up(v'(x)) = v'(y)$

↳ due to $v \equiv_M v'$, (1) is satisfied.

Conditions 2 and 3: similar.

Subclass 2: $x := x+1$, diagonal-free guards

Let M be max constant occurring among all guards in the automaton.

Problem: Show that region-equivalence $\equiv_M$ satisfies Lemma 3 with resets replaced with updates of the above form.

To show: $v \equiv_M v' \Rightarrow up(v) \equiv_M up(v')$

- Same as before.

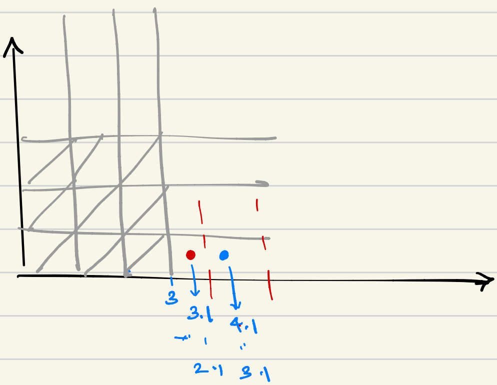
Problem: Consider subclass with $x := x-1$, diagonal-free guards.

Is there an 'M', in general, for which $\equiv_M$ satisfies Lemma 3?

Can we give an M s.t. $v \equiv_M v' \Rightarrow up(v) \equiv_M up(v')$

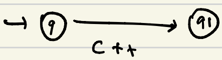
for decrement update.

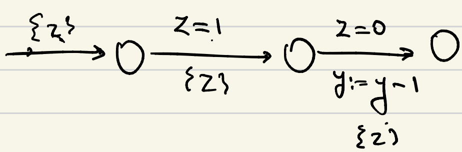
No.

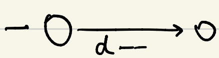


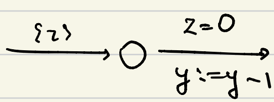
Idea of undecidability of this subclass:

x, y, z









When $z=0$, value of x gives counter 'c'

y ——— 'd'

Summary:

- Emptiness problem for UTA
- Undecidable
- Some sublasses with decidability. Proof based on regions
- More decidable classes in the paper:

  Bouyer et al: Updatable Timed Automata.