Timed Automata [Alur, Dill '94 . .]

Timed words

$$\begin{matrix} a_1 & a_2 & \cdots & a_n \\ t_1 & t_2 & \cdots & t_n \end{matrix}$$

Clocks = C

clock constraint

$$\bigcirc \xrightarrow[R]{a,\ \varphi} \bigcirc$$

reset : $R \subseteq C$

$x \in C$, $q \in \mathbb{Q}$

$x \sim q \mid \neg\varphi \mid \varphi \wedge \varphi$

Run: let t_1 elapse, evaluate φ, execute a, reset R

let $t_2 - t_1$ elapse, . . .

$L(TA) = \{ (w,\sigma) \mid \exists$ an acc. run of TA on .. $\}$

L is a timed regular lang if $\exists TA$ s.t. $L = L(TA)$

Clock valuation : $v : C \to \mathbb{R}_{\geq 0}$

$v + c$: add c to value of every clock

$v[X \leftarrow 0]$: reset clocks in X to 0

Closure properties

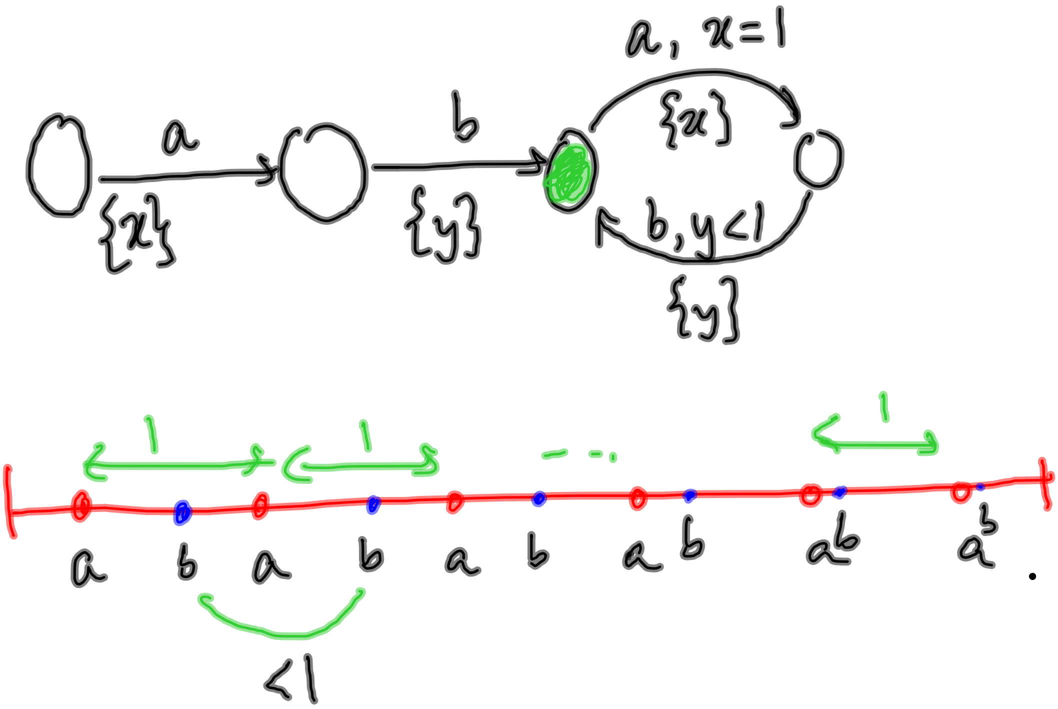
Union ✓ disjoint copies

Intersection ✓ – suitably enhanced product constr

$$(s,t) \xrightarrow[R_1 \cup R_2]{a,\ \varphi_1 \wedge \varphi_2} (s',t')$$

Complementation ✗

Reals vs rationals



If TA accepts any timed word, it accepts a timed word where all actions happen at rational time pts

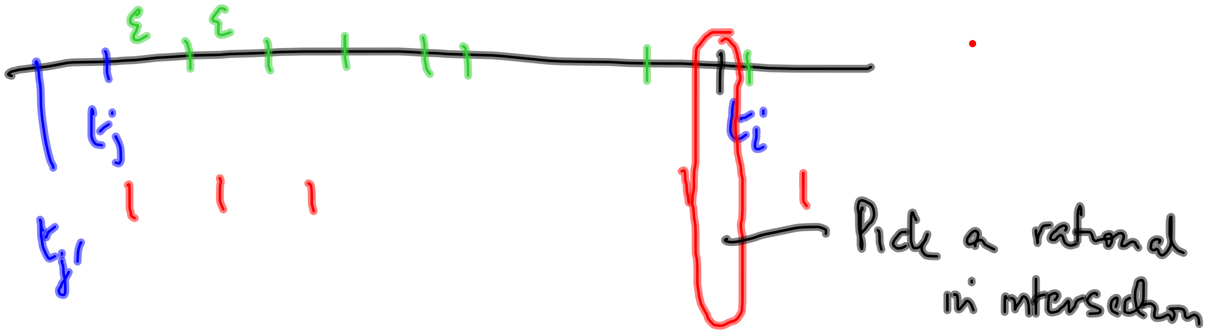
If $(\alpha, \sigma) \in L(TA)$ $\exists$ rational σ' s.t. $(\alpha, \sigma') \in L(TA)$

Fix TA Fix ε s.t all constants used in TA are integral multiples of ε

$$\begin{array}{cccc} (a_1, t_1) & (a_2, t_2) & \ldots & (a_k, t_k). \\ | & | & & \\ t_1' & t_2' & & t_k' \end{array}$$

$t_1, t_2 \ldots$

- If t_i is rational, $t_i' = t_i$
- If $t_i = t_j + l\varepsilon$ for some $j < i$, $t_i' = t_j' + l\varepsilon$



Reason about $L(TA)$

"State space" = configuration (s, v) — uncountable

TA : fix a constant c TA_c = TA with all constants in TA multiplied by c

Run: $(s_0, v_0) \xrightarrow[t_1]{a_1} (s_1, v_1) \xrightarrow[t_2]{a_2} \cdots \xrightarrow[t_k]{a_k} (s_k, v_k)$

$t_1 \cdot c$ $t_2 \cdot c$ $t_k \cdot c$

Take TA, clear the denominators of all constants

↳ all integer constraints

Induce an equivalence relation on clock valuations

s.t. if $v \sim v'$

then any pair of configurations $(s, v), (s, v')$

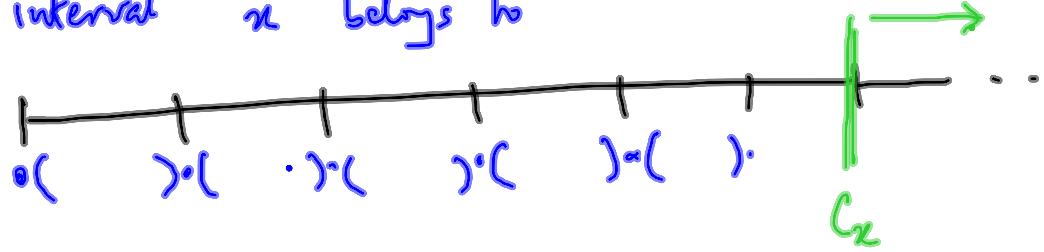
has "same" future

L a timed regular language

$\mathrm{Untime}(L) = \{ w \mid (w,\sigma) \in L \}$

$v \sim v'$? (wrt to fixed TA with all integral constraints)

- Given a clock x, both v & v' agree on which "interval" x belongs to



- For clocks x, y, which of x/y will change its interval first –

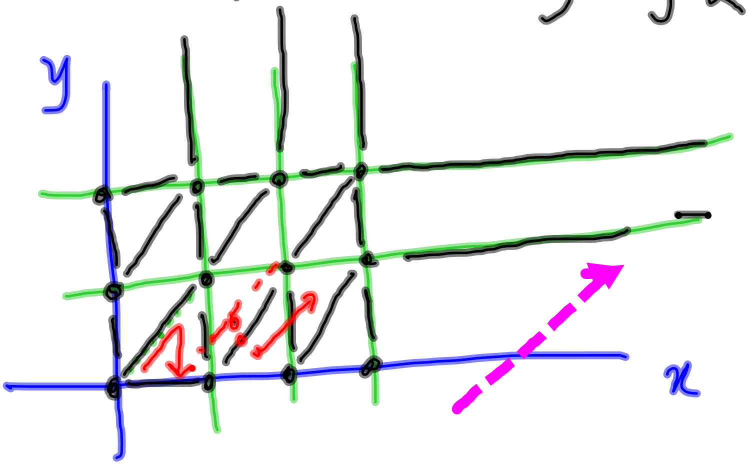
$$\left.\begin{array}{l} \mathrm{fract}(v(x)) \geq \mathrm{fract}(v(y)) \\ \text{iff } \mathrm{fract}(v'(x)) \geq \mathrm{fract}(v'(y)) \end{array}\right\}$$ if both values within c_x, c_y

$v \sim v'$ if
1. $\forall x$ $\lfloor v(x) \rfloor = \lfloor v'(x) \rfloor$ or both values $> c_x$
2. $\forall x \forall y$ $\mathrm{fract}(v(x)) \leq \mathrm{fract}(v(y))$ iff $\mathrm{fract}(v'(x)) \leq \mathrm{fract}(v'(y))$
3. $\forall x$ If $v(x) \leq c_x$, $v(x)$ is an int iff $v'(x)$ is an int

Equivalence reln on clock valuations

Equivalence classes ≡ clock regions

Example: 2 clocks, x, y $\quad c_x = 3 \quad c_y = 2$



\# regions is finite

- $\forall x$, interesting values of $v(x)$ are $2(c_x+1)$

  $\therefore \prod_{x \in C} 2c_x+2$

Replace $|C|$ by # of constraints in TA

- Fix $\underline{X} \subseteq C$ with legal values

X $\forall x,y \in C$ want an ordering on $frac(v(x)), frac(v(y))$

$\Rightarrow$ permutation on $\cdot X$ — fix permutation — subset of =

Total # regions bounded by $|C|!\, 2^{|C|} \cdot \prod_{x \in C} 2c_x+2$

Region automaton

States $(s, [v]) \xrightarrow[\Delta t]{a} (s', [v'])$?

$[v] = \alpha$, $[v'] = \alpha'$

$s \xrightarrow[R]{a, \varphi} s'$ $\exists v \in \alpha$ $v + \Delta t \models \varphi$

$v' = v + \Delta t [R \leftarrow 0] \in \alpha'$

Given a region α, "time successors" of α

regions that can be reached from α by letting time elapse timesucc(α)

- Case 1 $\alpha(x) > c_x \ \forall x$ $\quad$ timesucc(α) = α

- Case 2 $C_0 \subseteq C$ has integer value

  β : all values in C_0 shift to next interval
  all values in $C - C_0$ stay in same interval
  ordering of fractional values does not change

  timesucc(α) = timesucc(β)

  Recall assumption that nonzero time elapses at each move

Case 3

All fractional

let C_{max} = clocks with max fractional value

β = new region in which C_{max} clocks attain next integer

$timesucc(\alpha) = \{\alpha\} \cup \{\beta\} \cup timesucc(\beta)$

<u>Claim</u> α' is a time successor of α if

$\forall v \in \alpha, \exists t$ s.t. $v+t \in \alpha'$

$$s \xrightarrow[R]{a,\varphi} s'$$

$$(s,\alpha) \xrightarrow{a} (s',\alpha')$$

Find $\alpha'' \in \text{timesucc}(\alpha)$ s.t. $\alpha'' \models \varphi$

$\alpha' = \alpha''[R \leftarrow 0]$

Therefore can effectively construct a "region automaton" for TA

Reg(TA)

Thm: $L(\mathrm{Reg}(TA)) = \mathrm{Untime}(L(TA))$

Proof:

TA $(s_0, v_0) \xrightarrow[t_1]{a_1} (s_1, v_1) \xrightarrow[t_2]{a_2} \cdots \xrightarrow[t_k]{a_k} (s_k, v_k)$

"easy"

Reg(TA) $(s_0, [v_0])$ $(s_1, [v_1])$ $(s_k, [v_k])$

$$(s_0, \alpha_0) \xrightarrow{a_1} (s_1, \alpha_1) \xrightarrow{a_2} \cdots \xrightarrow{a_k} (s_k, \alpha_k)$$

t_1' $\quad \alpha_0''$ $\quad t_2'$ $\quad \alpha_1''$

$$(s_0, v_0) \xrightarrow[t_1]{a_1} (s_1, v_1) \xrightarrow[t_2]{a_2} \cdots \xrightarrow[t_k]{a_k} (s_k, v_k)$$

$t_1 + t_2'$

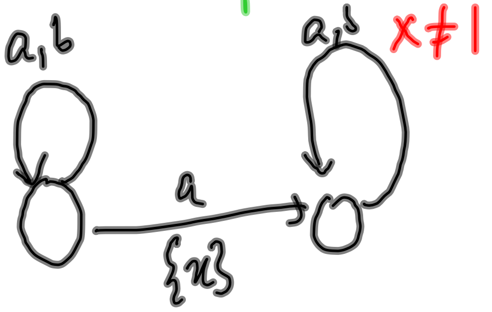
For each timed regular language L, Untime(L) all
is a regular language

Emptiness is decidable

$L(TA) = \phi$ iff $Untime(L(TA)) = \phi$

Closure under complementation?

L



$\exists$ an a s.t. no action happens 1 time unit later

$\bar{L}$: $\forall$ action a, $\exists$ an action one time unit later

L' such that $\text{Untime}(L') = a^*b^*$
& all a's happen strictly before $t = 1$

Consider $L' \cap \bar{L}$ - every a has a matching b

$\text{Untime}(L' \cap \bar{L}) = a^m b^n, \; n \geq m$

not regular

$\therefore L' \cap \bar{L}$ is not regular

$\therefore$ either $L' \cap \bar{L}$ is not regular

$\therefore \bar{L}$ is not regular

Universality : Is every (w, σ) accepted by TA?

2 counter machines

1 : i_1

2 : i_2

3 : ⋮

⋮

m : i_m

Termination is undecidable

each instruction is one of following:

- increment c_1
- decrement c_1 (only works if $c_1 > 0$)
- increment / dec c_2
- if $c_1 = 0$ goto i_k
- if $c_2 = 0$ goto i_k

Code a run of a 2 counter machine using a timed word

$(i_1, 0, 0) \longrightarrow (i_2,$

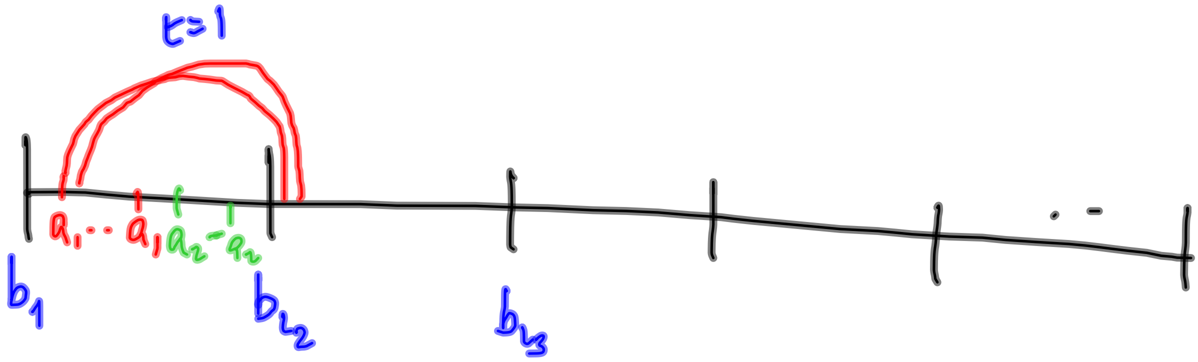
$(i, c_1, c_2) \longrightarrow (i', c_1', c_2')$

Letters $b_1, \ldots, b_m$ to encode $i_1, \ldots, i_m$

a_1 to "count" c_1

a_2 to "count" c_2

$(i_j, c_1, c_2) \rightsquigarrow b_j a_1^{c_1} a_2^{c_2}$



Construct a timed automaton that accepts all non-runs

This automaton is universal iff 2 counter machine has no acc run.

- Non run because
  - not of form $(b_i a_1^* a_2^*)^*$
  - does not start with b_1
  - does not end with b_m
  - b_i's not at integer points

For a_1/a_2 guess that

- Some a_1 in this interval has no a_1 matching in next int.
- Some a_1 in next interval has no a_1 matching now

Use 2 clocks for consec events

$\therefore$ Inclusion & Equivalence of timed reg langs is undecidable

Trivially construct $TA_{univ} = \Sigma^*$, no timing constraints

L universal $\Leftrightarrow$ $L_{univ} \subseteq L$

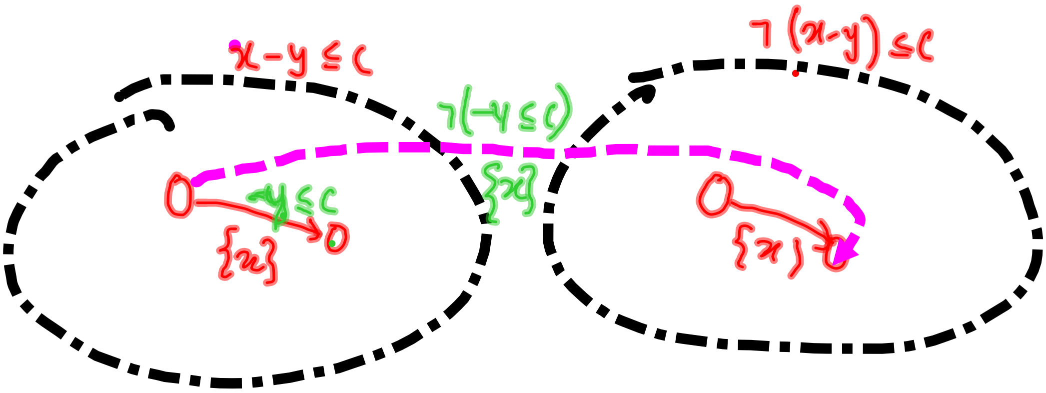
|||ly check $L \stackrel{?}{=} L_{univ}$

Constraints

$x - y \le c$ "Diagonal" constraints

Bouyer Petit

One diagonal constraint : Make two copies of TA



Generalizing resets

$x \leftarrow c$ ✓

$x+y \leq x'+y'$

$x \leftarrow y+7$

$x \mapsto$ $x > y$

y $x < 7$

$y > 5$

non deterministic reset