Algorithms for Model Checking (2IW55)

Lecture 13

Timed Verification: Timed Automata

Chapter 16, 17

Tim Willemse
(timw@win.tue.nl)
http://www.win.tue.nl/~timw
HG 6.81

Outline

Timed Automata

Let $\mathcal{T} = \langle L, l_0, Act, C, \longrightarrow, \iota, AP, \ell \rangle$ be a Timed Automaton.

- Time divergent paths have infinite execution time:

$$\Delta(s_0 \overset{d_0}{\rightsquigarrow} s_1 \overset{d_1}{\rightsquigarrow} \ldots) = \sum_{i=0}^{\infty} d_i = \infty$$

- A Timed Automaton is timelock-free if no reachable state contains a timelock
  - a state contains a timelock if it has no time-divergent path
- A Timed Automaton is non-Zeno if there is no initial Zeno path in $[\mathcal{T}]$
  - a path is Zeno if it is time-convergent and performs infinitely many actions

Timed CTL

Syntax of TCTL state-formulae over AP and set of clocks C:

$$\mathcal{S} ::= \mathsf{true} \mid AP \mid \mathcal{S} \wedge \mathcal{S} \mid \neg\mathcal{S} \mid \mathsf{E}\,\mathsf{F}_J\,\mathcal{S} \mid \mathsf{A}\,\mathsf{F}_J\,\mathcal{S}$$

where $J \subseteq \mathbb{R}_{\geq 0}$ is an interval whose bounds are naturals

- J can have the following forms: $[n,m]$, $(n,m]$, $[n,m)$ or (n,m) for $n,m \in \mathbb{N}$ and $n \leq m$
- For right-open intervals, $m = \infty$ is also allowed

Timed CTL

Let $\mathcal{T} = \langle L, L_0, Act, C, \longrightarrow, \iota, AP, \ell \rangle$ be a Timed Automaton.

- Satisfaction of a TCTL formula ϕ is defined as $(l, v) \models \phi$
- For TCTL state-formulae ϕ, the satisfaction set $\mathsf{sat}(\phi)$ is defined by:

$$\mathsf{sat}(\phi) = \{s \in L \times (C \to \mathbb{R}_{\geq 0}) \mid s \models \phi\}$$

- $\mathcal{T}$ satisfies TCTL-formula ϕ iff ϕ holds in all initial states of $\mathcal{T}$:

$$\mathcal{T} \models \phi \qquad \text{if and only if} \qquad \forall l \in L_0 (l, v_0) \models \phi$$

where $v_0(x) = 0$ for all clocks $x \in C$.

TCTL

- A timelock points at a modelling problem
- A timelock-free state has at least one time-divergent path
- Absence of timelock for a state s holds iff $s \models \mathsf{E}\,\mathsf{G}\,\mathsf{true}$
- Absence of timelock in a Timed Automaton $\mathcal{T}$ holds iff for all reachable state $s \in [\mathcal{T}]$, $s \models \mathsf{E}\,\mathsf{G}\,\mathsf{true}$ holds
- Hence, timelocks can be found by means of model checking

Outline

Clock Equivalence

Let $\mathcal{T} = \langle L, L_0, Act, C, \longrightarrow, \iota, AP, \ell \rangle$ be a non-Zeno Timed Automaton.

Definition

Let ϕ be a TCTL formula. Then

$$\mathcal{T} \models \phi \quad \text{iff} \quad [\mathcal{T}] \models \phi$$

Problem: $[\mathcal{T}]$ is infinite state, so it cannot be explored exhaustively. Therefore:

1. Map TCTL formulae ϕ onto proper CTL formulae $\hat{\phi}$
2. Consider a finite quotient of $[\mathcal{T}]$ with respect to a bisimulation relation $\sim$

such that: $\mathcal{T} \models_{TCTL} \phi \quad \text{iff} \quad \mathcal{T} / \sim \models_{CTL} \hat{\phi}$

Clock Equivalence

Let ϕ be a TCTL formula and $\mathcal{T} = \langle L, L_0, Act, C, \longrightarrow, \iota, AP, \ell \rangle$ a Timed Automaton

- Assume $J \neq [0, \infty)$ occurs in ϕ
- Let $\mathcal{T} \oplus z = \langle L, L_0, Act, C \cup \{z\}, \longrightarrow, \iota, AP, \ell \rangle$ for $z \notin C$
- $(l, v) \models \mathsf{E}\,\mathsf{F}_J\,\phi$ iff $(l, [v]_{\{z\}}) \models (z \in J) \wedge \phi$
- Likewise, $\mathsf{E}\,\mathsf{G}_J$, $\mathsf{A}\,\mathsf{F}_J$ and $\mathsf{A}\,\mathsf{G}_J$

Formally

for any state $(l, v) \in \mathcal{T} \oplus z$:

$$(l, v) \models \mathsf{E}\,\mathsf{F}_J\,\phi \qquad \text{iff} \qquad (l, [v]_{\{z\}}) \models \mathsf{E}\,\mathsf{F}\,((z \in J) \wedge \phi)$$

- Note: atomic clock constraints are atomic propositions in $[\mathcal{T} \oplus z]$
- So, the transformation yields a CTL formula
- For instance, $\mathsf{E}\,\mathsf{G}_{[0,2]}\,\phi$ yields $\mathsf{E}\,\mathsf{G}\,((0 \leq z \wedge z \leq 2) \rightarrow \phi)$

Clock Equivalence

Observations:

- A Timed Automaton $\mathcal{T}$ has a finite number of locations
- $[\mathcal{T}]$ has an infinite number of states due to clock valuations only

Impose an equivalence $\sim$ on clock valuations such that $(C \to \mathbb{R}_{\geq 0})/\sim$ is finite.
Moreover:

1. Equivalent clock valuations satisfy the same clock constraints:

$$\nu \sim \nu' \qquad \text{implies} \qquad (\nu \models \phi \quad \text{iff} \quad \nu' \models \phi)$$

2. Time-divergent paths starting from equivalent states are equivalent

Clock Equivalence

Major result from [1]:

- Criteria 1 and 2 are satisfied if equivalent clock valuations:
  - Agree on the integer parts of all clock values, and
  - Agree on the ordering of the fractional parts of all clocks
- This gives rise to a countably infinite set of equivalence classes
- Finiteness is obtained by considering the maximal constants to which clocks are compared:
- If a clock grows beyond the maximal constant to which it is compared, its exact value is no longer of importance.

[1] R. Alur and D.L. Dill, *A theory of timed automata*, in **Theoretical Computer Science 126(2):183–235, 1994**

Clock Equivalence

Clock Equivalence (1)

- $\nu \models x < c$ whenever $\nu(x) < c$
- Equivalently: $\lfloor \nu(x) \rfloor < c$ (i.e. the greatest integer at most $\nu(x)$

- $\nu \models x \leq c$ whenever $\nu(x) < c$ or $\nu(x) = c$
- Equivalently: $\lfloor \nu(x) \rfloor < c$ or $\lfloor \nu(x) \rfloor = c$ and $\mathsf{frac}(\nu(x)) = 0$

First proposal

Two clock valuations ν and ν' are equivalent, denoted $\nu \sim \nu'$ iff

1. for any $x \in C$:
   $\lfloor \nu(x) \rfloor = \lfloor \nu'(x) \rfloor$ and $\mathsf{frac}(\nu(x)) = 0$ iff $\mathsf{frac}(\nu'(x)) = 0$

- Decidability of $\sim$ is guaranteed because clocks are compared to natural numbers.

Clock Equivalence

Clock Equivalence (2)

- Assume a location l with invariant **true** and two outgoing switches:
  - action a, guarded by $x \geq 2$; action b, guarded by $y > 1$
- Assume $1 < \nu(x) < 2$ and $0 < \nu(y) < 1$
  - then $(l,\nu) \overset{a}{\not\rightarrow}$ and $(l,\nu) \overset{b}{\not\rightarrow}$
  - invariant l is **true**, so time may elapse
- The transition that is first enabled depends on $x < y$ or $x \geq y$
  - action a is enabled first if $\mathsf{frac}(\nu(x)) \geq \mathsf{frac}(\nu(y))$

Second proposal

Two clock valuations ν and ν' are equivalent, denoted $\nu \sim \nu'$ iff

1. for any $x \in C$:
   $\lfloor \nu(x) \rfloor = \lfloor \nu'(x) \rfloor$, and $\mathsf{frac}(\nu(x)) = 0$ iff $\mathsf{frac}(\nu'(x)) = 0$
2. for all $x, y \in C$: $\mathsf{frac}(\nu(x)) \leq \mathsf{frac}(\nu(y))$ iff $\mathsf{frac}(\nu'(x)) \leq \mathsf{frac}(\nu'(y))$

Clock Equivalence

Clock Equivalence (3)

- Problem second proposal: countable, but still infinite
- Solution: for $\mathcal{T} \models \phi$, only the clock constraints in $\mathcal{T}$ and ϕ are relevant.
- Let $c_x \in \mathbb{N}$ be the largest constant to which x is compared in $\mathcal{T}$ or ϕ
- If $\nu(x) > c_x$, then the exact value of x is of no importance (x only grows)

Final proposal

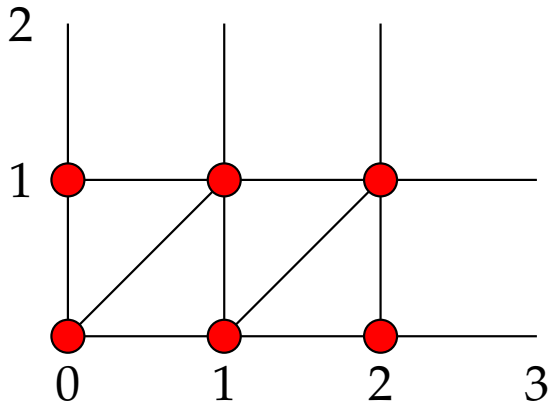
Two clock valuations ν and ν' are equivalent, denoted $\nu \sim \nu'$ iff

1. for any $x \in C$: $\nu(x), \nu'(x) > c_x$ or $\nu(x), \nu'(x) \leq c_x$
2. for any $x \in C$: if $\nu(x), \nu'(x) \leq c_x$ then:
   $\lfloor \nu(x) \rfloor = \lfloor \nu'(x) \rfloor$ and $\mathsf{frac}(\nu(x)) = 0$ iff $\mathsf{frac}(\nu'(x)) = 0$
3. for any $x, y \in C$: if $\nu(x), \nu'(x) \leq c_x$ and $\nu(y), \nu'(y) \leq c_y$, then:
   $\mathsf{frac}(\nu(x)) \leq \mathsf{frac}(\nu(y))$ iff $\mathsf{frac}(\nu'(x)) \leq \mathsf{frac}(\nu'(y))$

Clock Equivalence

Example

Consider a Timed Automaton with clocks x and y, with $c_x = 2$ and $c_y = 1$. The clock regions are shown below:



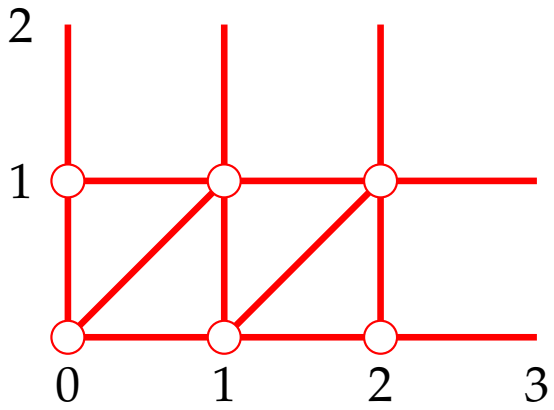
Regions:

6 Corner points, e.g. $[(0,0)]$

Clock Equivalence

Example

Consider a Timed Automaton with clocks x and y, with $c_x = 2$ and $c_y = 1$. The clock regions are shown below:



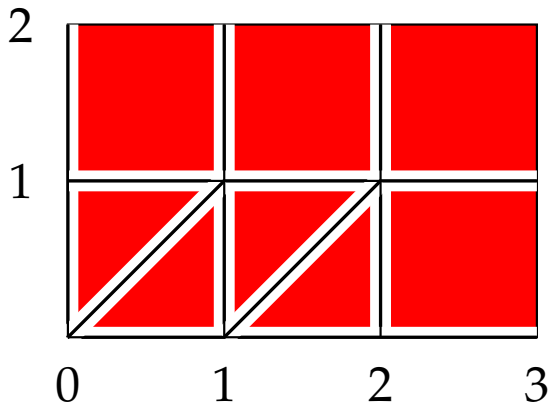
Regions:

14 Open line segments: e.g. $[0 < x = y < 1]$

Clock Equivalence

Example

Consider a Timed Automaton with clocks x and y, with $c_x = 2$ and $c_y = 1$. The clock regions are shown below:



Regions:

8 Open regions: e.g. $[0 < x < y < 1]$

Clock Equivalence

- The clock region of $\nu \in [C \to \mathbb{R}_{\geq 0}]$, denoted $[\nu]$ is defined by:

  $$[\nu] := \{\nu' : C \to \mathbb{R}_{\geq 0} \mid \nu \sim \nu'\}$$

- The state region of a state (l, ν) in $[\mathcal{T}]$ is defined by:

  $$[(l, \nu)] := (l, [\nu])$$

- The number of clock regions is bounded from below by:

  $$\text{if for all } x \in C: \ c_x \geq 1 \text{ then } R_l := |C|! \times \prod_{x \in C} c_x$$

- The number of clock regions is bounded from above by:

  $$\text{if for all } x \in C: \ c_x \geq 1 \text{ then } R_u := |C|! \times 2^{|C|-1} \times \prod_{x \in C} (2(c_x + 1))$$

- The number of state regions in $[\mathcal{T}]/\sim$ is finite:

  $$|L| \times R_l \ \leq \ S/\sim \ \leq \ |L| \times R_u$$

Clock Equivalence

Let $\mathcal{T} = \langle L, l_0, Act, C, \longrightarrow, \iota, AP, \ell \rangle$ be a Timed Automaton. Let $[\mathcal{T}] = \langle S, S_0, Act, \rightarrow, \mapsto, AP', \ell' \rangle$

- Let $\phi \in \mathcal{C}_a(C)$. For $\nu, \nu' : C \rightarrow \mathbb{R}_{\geq 0}$ such that $[\nu] = [\nu']$

$$\nu \models \phi \qquad \text{iff} \qquad \nu' \models \phi$$

- Let $p \in AP'$. For any $s, s' \in S$ such that $s \sim s'$:

$$s \models p \qquad \text{iff} \qquad s' \models p$$

Clock Equivalence

Theorem

Clock equivalence is a (time abstract) bisimulation equivalence over AP'

Property ϕ: reachability of the location q.

Time abstract bisimulation: two states (l, ν) and (l, ν') have the same behaviour (w.r.t. ϕ) when:

1. Any action transition enabled from ν is also enabled from ν'; and the target states have the same behaviour
2. For any delay transition d from ν, there is a delay transition d', such that $(l, \nu + d)$ and $(l, \nu' + d')$ have the same behaviour

. . .(and vice versa)

Time abstract bisimulation: $(l, v)\ B\ (l, v')$ when

1. For any $l \xrightarrow{g\ a\ R} l'$, we have
   $(l, v) \xrightarrow{a} (l', [v]_R)$ implies
   there is $(l, v') \xrightarrow{a} (l', [v']_R)$ and $(l, [v]_R)\ B\ (l, [v']_R)$ (and vice versa)
2. For any $(l, v) \xmapsto{d} (l, v + d)$
   there is d' such that
   $(l, v') \xmapsto{d'} (l, v' + d')$ and $(l, v + d)\ B\ (l, v' + d')$ (and vice versa)

Outline

Region Automata

Model Checking TCTL over $\mathcal{T} = \langle L, L_0, Act, C, \longrightarrow, \iota, AP, \ell \rangle$

Main Procedure

Reduce the verification of TCTL formulae over Timed Automata to a model checking problem over the Region Automaton for a CTL formula

- For a TCTL formula ϕ introduce a new clock z_J for every interval J ($\neq [0, \infty)$) occurring in ϕ; let c_{z_J} be the maximal integer to which z_J is compared
- Let z_ϕ be the set of all clocks introduced by ϕ
- Compute the clock regions for the clock valuations $C \cup z_\phi \rightarrow \mathbb{R}_{\geq 0}$
- Then $\mathcal{T} \models_{TCTL} \phi$ iff $R(\mathcal{T} \oplus z_\phi) \models_{CTL} \hat{\phi}$

Region Automata

Let $\mathcal{T} = \langle L, l_0, Act, C, \longrightarrow, \iota, AP, \ell \rangle$ be a Timed Automaton.

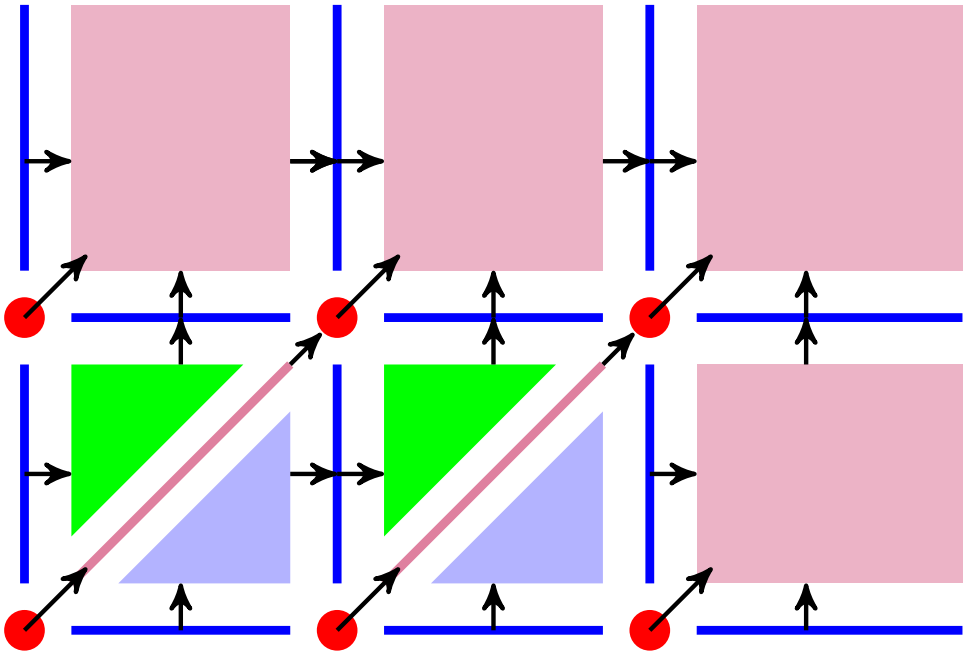
- Clock region $r_\infty = \{\nu \in [C \to \mathbb{R}_{\geq 0}] \mid \forall x \in C : \ \nu(x) > c_x\}$ is unbounded
- r' is the successor clock region of r, denoted $r' = \mathsf{succ}(r)$, if either:
  1. $r = r_\infty$ and $r = r'$, or
  2. $r \neq r_\infty, r \neq r'$ and for all $\nu \in r$:

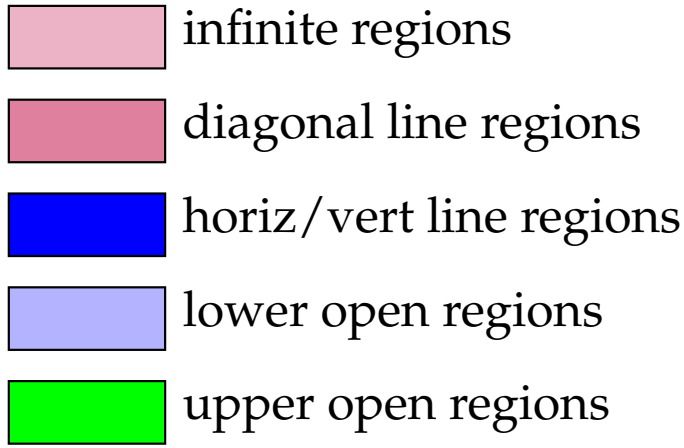
$$\exists d \in \mathbb{R}_{\geq 0} : \ (\nu + d \in r' \qquad \text{and} \qquad \forall 0 \leq d' \leq d : \ \nu + d' \in r \cup r')$$

- The successor region: $\mathsf{succ}((l, \nu)) := (l, \mathsf{succ}(\nu))$
- Resetting a region: $r[R := 0] := \{\nu \in [C \to \mathbb{R}_{\geq 0}] \mid \exists \nu' \in r : \nu = [\nu']_R\}$

Region Automata

Clock regions and their successor regions.





Region Automata

The Region Automaton $R(\mathcal{T})$ of a non-Zeno $\mathcal{T} = \langle L, L_0, Act, C, \longrightarrow, \iota, AP, \ell \rangle$ is defined as:

$$R(\mathcal{T}) = \langle S, S_0, Act \cup \{\tau\}, \rightarrow', AP', \ell' \rangle$$

where the state regions are defined as:

- $S = (L \times (C \to \mathbb{R}_{\geq 0}))/\sim = \{[s] \mid s \in S_{[\mathcal{T}]}\}$
- $S_0 = \{[s] \mid s \in S_{0\,[\mathcal{T}]}\}$
- $\ell'((l,r)) = \ell(l) \cup \{\phi \in \mathcal{C}_a(C) \mid r \models \phi\}$
- $\rightarrow' \subseteq S \times Act \cup \{\tau\} \times S$ is defined as:

$$\frac{l \xrightarrow{g\ a\ R} l' \qquad r \models g \qquad r[R := 0] \models \iota(l')}{(l,r) \xrightarrow{a}{}' (l', r[R := 0])}$$

$$\frac{r \models \iota(l) \qquad \mathsf{succ}(r) \models \iota(l)}{(l,r) \xrightarrow{\tau}{}' (l', \mathsf{succ}(r))}$$

Outline

Wrap-up

Other verification problems:

1. The TCTL model checking problem is PSPACE-complete
2. The model checking problem for timed LTL (and TCTL*) is undecidable
3. The satisfaction problem for TCTL is undecidable

Some open questions:

- Adding clock constraints $x + y < c$:
  - for two clocks, decidable,
  - for four clocks, undecidable,
  - for three clocks, unknown.