Algorithms for Model Checking (2IW55)

Lecture 12

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

Analysing Semantics

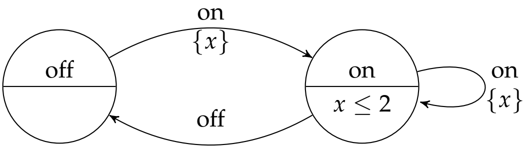
Timed CTL

Exercise

Timed Automata

A timed automaton with propositions AP is a tuple $\mathcal{T} = \langle L, L_0, Act, C, \longrightarrow, \iota, AP, \ell \rangle$

- L is a finite set of locations; $L_0 \subseteq L$ is a non-empty set of initial locations
- Act is the set of actions
- C is a finite set of clock variables
- $\longrightarrow \subseteq L \times \mathcal{C}(C) \times Act \times 2^C \times L$ is the set of switches
- $\iota : L \to \mathcal{C}(C)$ is the invariant assignment function
- $\ell : L \to 2^{AP}$ is the labelling function



Timed Automata

Recalling intuition:

- A switch $l \xrightarrow{g\ a\ R} l'$ means that:
  - action a is enabled whenever guard g evaluates to true.
  - upon executing the switch, we move from location l to location l' and reset all clocks in R to zero.
  - only locations l' that can be reached with clock values that satisfy the location invariant.
- an invariant $\iota(l)$ limits the time that can be spent in location l.
  - staying in location l only is allowed as long as the invariant evaluates to true.
  - before the invariant becomes invalid location l must be left.
  - if no switch is enabled when the invariant becomes invalid no further progress is possible: timed deadlock.

Timed Automata

Recalling notation:

- A clock valuation ν for a set C of clocks is a function $\nu : C \rightarrow \mathbb{R}_{\geq 0}$
- We write $\nu \models \phi$ iff $[\phi]_\nu = \mathsf{true}$.
- Clock valuation update: $\nu + d$ is defined as: $(\nu + d)(x) = \nu(x) + d$ for all $d \in \mathbb{R}_{\geq 0}$.
- Clock valuation reset: $[\nu]_R$ is defined as: $[\nu]_R(x) = 0$ if $x \in R$, else $\nu(x)$.

Additional notation:

- Let $\mathcal{C}(C)$ be the set of clock constraints over C.
- The atomic clock constraints $\mathcal{C}_a(C)$ over C is the subset of $\mathcal{C}(C)$ not containing true and $\wedge$.

Timed Automata

Let $\mathcal{T} = \langle L, L_0, Act, C, \longrightarrow, \iota, AP, \ell \rangle$ be a Timed Automaton.
Its semantics is defined as a timed transition system: $[\mathcal{T}] = \langle S, S_0, Act, \rightarrow, \mapsto, AP', \ell' \rangle$

- $S = \{(l, \nu) \in L \times (C \rightarrow \mathbb{R}_{\geq 0}) \mid \nu \models \iota(l)\}$, i.e. all combinations of locations and clock valuations that do not violate the location invariant.
- $S_0 = \{(l, \nu) \in L_0 \times (C \rightarrow \mathbb{R}_{\geq 0}) \mid \nu \models \iota(l) \wedge \forall x \in C : \nu(x) = 0\}$.
- $\longrightarrow \subseteq S \times Act \times S$ is defined as follows:

$$\frac{l \xrightarrow{g\ a\ R} l' \qquad \nu \models g \wedge \iota(l) \qquad \nu' = [\nu]_R \qquad \nu' \models \iota(l')}{(l, \nu) \xrightarrow{a} (l', \nu')}$$

- $\mapsto \subseteq S \times \mathbb{R}_{\geq 0} \times S$ is defined as follows:

$$\frac{\nu \models \iota(l) \qquad \forall 0 \leq d' \leq d : \nu + d' \models \iota(l)}{(l, \nu) \overset{d}{\mapsto} (l, \nu + d)}$$

- $AP' = AP \cup \mathcal{C}_a(C)$; the labelling function: $\ell'((l, \nu)) = \ell(l) \cup \{\phi \in \mathcal{C}_a(C) \mid \nu \models \phi\}$

Timed Automata

Lemma

Let $\iota(l)$ be a negation-free location invariant. Then for all $d \in \mathbb{R}_{\geq 0}$ and all ν:

$$\nu \models \iota(l) \text{ and } \nu + d \models \iota(l) \text{ implies } \forall 0 \leq d' \leq d : \ \nu + d' \models \iota(l)$$

- The proof follows by a structural induction on $\iota(l)$.
- This means that for negation-free location invariants, we can simplify the rule for timed transition relations:

$$\frac{\nu \models \iota(l) \qquad \nu + d \models \iota(l)}{(l, \nu) \overset{d}{\mapsto} (l, \nu + d)}$$

Outline

Timed Automata

Analysing Semantics

Timed CTL

Exercise

Analysing Semantics

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

- Assume $\forall 0 \leq d' \leq d : \; \nu + d' \models \iota(l)$ for fixed $d \in \mathbb{R}_{\geq 0}$
- A possible execution fragment starting from the location l is:

$$(l, \nu) \overset{d_1}{\mapsto} (l, \nu + d_1) \overset{d_2}{\mapsto} (l, \nu + d_1 + d_2) \overset{d_3}{\mapsto} (l, \nu + d_1 + d_2 + d_3) \overset{d_4}{\mapsto} \ldots$$

  - where $d_i > 0$ and the infinite sequence $d_1 + d_2 + \ldots$ converges towards d
  - such path fragments are called time-convergent, i.e. time advances only up to a certain value.
- Time-convergent execution fragments are unrealistic and ignored
  - compare to unrealistic executions in Kripke Structures and fairness constraints that eliminate these

Analysing Semantics

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

- Infinite path π is time-divergent if $\Delta(\pi) = \infty$
- The function $\Delta : Act \cup \mathbb{R}_{\geq 0} \to \mathbb{R}_{\geq 0}$ is defined as follows:

$$\Delta(\tau) = \begin{cases} 0 & \text{if } \tau \in Act \\ d & \text{if } \tau = \tau \in \mathbb{R}_{\geq 0} \end{cases}$$

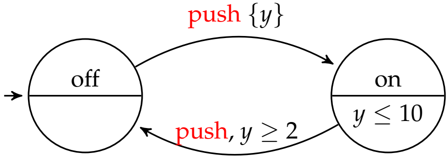
- For infinite execution fragments $\sigma = s_0 \xrightarrow{\tau_1} s_1 \xrightarrow{\tau_2} s_2 \ldots$ in $[\mathcal{T}]$ let:

$$\Delta(\sigma) = \sum_{i=0}^{\infty} \Delta(\tau_i)$$

  - for path fragment π in $[\mathcal{T}]$ induced by σ: $\Delta(\pi) = \Delta(\sigma)$
- For a state $s \in [\mathcal{T}]$: $\mathsf{Path}_{\mathsf{div}}(s) = \{\pi \in \mathsf{path}(s) \mid \pi \text{ is time-divergent}\}$

Analysing Semantics

Light automaton:



- The path $\pi \in [\![\text{Light}]\!]$ in which on-and off-periods of one/two time units alternate:

$$\pi = (\text{off}, 0)\ (\text{off}, 1)\ (\text{on}, 0)\ (\text{on}, 1)(\text{on}, 2)\ (\text{off}, 2)\ (\text{off}, 3)\ (\text{on}, 0)\ (\text{on}, 1) \ldots$$

  is time-divergent as $\Delta(\pi) = 1 + 2 + 1 + 2 + \ldots = \infty$

- The path:

$$\pi' = (\text{off}, 0)\ (\text{off}, \frac{1}{2})\ (\text{off}, \frac{3}{4})\ (\text{off}, \frac{7}{8})\ (\text{off}, \frac{15}{16})\ \ldots$$

  is time-convergent, since $\Delta(\pi') = \sum_{i \geq 1} (\frac{1}{2})^i = 1 < \infty$

Analysing Semantics

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

- State $s \in [\mathcal{T}]$ contains a timelock if $\mathsf{Path}_{\mathsf{div}}(s) = \emptyset$
  - there is no behaviour in s where time can progress *ad infinitum*
- $\mathcal{T}$ is timelock-free if no reachable state in $[\mathcal{T}]$ contains a timelock
- Timelocks are usually modelling flaws that should be avoided
  - like deadlocks, we need mechanisms to check their presence

Analysing Semantics

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

- If $\mathcal{T}$ can perform infinitely many actions in finite time it is Zeno
- A path π in $[\mathcal{T}]$ is Zeno if:
  - it is time-convergent, and
  - infinitely many actions $a \in Act$ are executed along π
- $\mathcal{T}$ is non-Zeno if there does not exist an initial Zeno path in $[\mathcal{T}]$
  - a path $\pi \in \mathsf{path}([\mathcal{T}])$ is time-divergent or
  - π is time-convergent, with nearly all (except for finitely many) transitions being delay transitions
- Zeno paths are considered modelling flaws that should be avoided
  - like deadlocks and timelocks, we need mechanisms to check for Zenoness

Analysing Semantics

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

Non-Zenoness can be checked directly on the Timed Automaton:

Suppose that for every control cycle:

$$l_0 \xrightarrow{g_1 \; a_1 \; R_1} l_1 \xrightarrow{g_2 \; a_2 \; R_2} \ldots \xrightarrow{g_n \; a_n \; R_n} l_n$$

there exists a clock $x \in C$ such that:

1. $x \in R_i$ for some $0 < i \leq n$, and
2. for all clock evaluations ν:

$$\nu(x) < 1 \text{ implies } (\nu \not\models g_j \text{ or } \nu \not\models \iota(l_j)) \text{ for some } 0 < j \leq n$$

Then $\mathcal{T}$ is non-Zeno

Outline

Timed CTL

A really temporal logic:

- CTL is a qualitative branching temporal logic
  - If p holds then in the future q holds: $p \rightarrow \mathsf{A}\,\mathsf{F}\, q$
- TCTL is a quantitative branching temporal logic:
  - If p holds then q holds within 10 time units: $p \rightarrow \mathsf{A}\,\mathsf{F}_{[0,10)}\, q$
- Full TCTL is described in:
  [1] T.A. Henzinger, X. Nicollin, J. Sifakis and S. Yovine, *Symbolic Model Checking for Real-Time Systems*, in **Information and Computation 111:193-244, 1994**
- We consider a subclass that is inspired by the Uppaal model checker (http://www.uppaal.com)
  - \- until-operator
  - \+ timed future operator and derived invariance operator

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

- $\mathsf{F}_J\,\phi$ asserts that a ϕ-state is reached at time instant $t \in J$
- 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
- Note: no $\mathsf{E}\,\mathsf{X}\,\phi$ (what does *next* mean in real-time?)

Timed CTL

- Always is obtained as follows.
- $\mathsf{E}\,\mathsf{G}_J\,\phi = \neg\mathsf{A}\,\mathsf{F}_J\,\neg\phi$
  $\mathsf{E}\,\mathsf{G}_J\,\phi$ asserts that for some path during the interval J, ϕ holds
- $\mathsf{A}\,\mathsf{G}_J\,\phi = \neg\mathsf{E}\,\mathsf{F}_J\,\neg\phi$
  $\mathsf{A}\,\mathsf{G}_J\,\phi$ requires ϕ to hold for all paths during the interval J
- Standard future-operator of CTL: $\mathsf{F}\,\phi = \mathsf{F}_{[0,\infty)}\,\phi$
- Standard global-operator of CTL: $\mathsf{G}\,\phi = \mathsf{G}_{[0,\infty)}\,\phi$

Timed CTL

Let $\mathcal{T} = \langle L, L_0, Act, C, \longrightarrow, \iota, AP, \ell \rangle$ be a Timed Automaton. Let (l, ν) be a state in $[\mathcal{T}]$

Satisfaction of a formula ϕ is defined as:

- $(l, \nu) \models \mathsf{true}$
- $(l, \nu) \models p$ iff $p \in \ell(l)$
- $(l, \nu) \models \neg\phi$ iff $(l, \nu) \not\models \phi$
- $(l, \nu) \models \phi_1 \wedge \phi_2$ iff $(l, \nu) \models \phi_1$ and $(l, \nu) \models \phi_2$
- $(l, \nu) \models \mathsf{E}\,\mathsf{F}_J\,\phi$ iff for some $\pi \in \mathsf{Path}_{\mathsf{div}}((l, \nu))$ we have $\pi \models \mathsf{F}_J\,\phi$
- $(l, \nu) \models \mathsf{A}\,\mathsf{F}_J\,\phi$ iff for all $\pi \in \mathsf{Path}_{\mathsf{div}}((l, \nu))$ we have $\pi \models \mathsf{F}_J\,\phi$

Note: path quantifiers are over time-divergent paths only.

Timed CTL

For infinite path fragments in $[\mathcal{T}]$ performing an infinite number of actions let:

$$(l_0, \nu_0) \overset{d_0}{\rightsquigarrow} (l_1, \nu_1) \overset{d_1}{\rightsquigarrow} (l_2, \nu_2) \overset{d_2}{\rightsquigarrow} \ldots \qquad \text{with } d_0, d_1, d_2, \ldots \geq 0$$

denote the equivalence class containing all infinite path fragments induced by execution fragments of the form:

$$\begin{array}{c} (l_0, \nu_0) \overset{d_0^1}{\mapsto} \ldots \overset{d_0^{k_0}}{\mapsto} (l_0, \nu + 0 + d_0) \xrightarrow{a_1} \\ (l_1, \nu_1) \overset{d_1^1}{\mapsto} \ldots \overset{d_1^{k_1}}{\mapsto} (l_1, \nu + 0 + d_1) \xrightarrow{a_2} \\ (l_2, \nu_2) \overset{d_2^2}{\mapsto} \ldots \overset{d_2^{k_2}}{\mapsto} (l_1, \nu + 0 + d_2) \xrightarrow{a_3} \ldots \end{array}$$

where $k_i \in \mathbb{N}, d_i \in \mathbb{R}_{\geq 0}$ and $a_i \in Act$ such that $\sum_{j=1}^{k_i} d_i^j = d_i$

For $\pi \in (l_0, \nu_0) \overset{d_0}{\rightsquigarrow} (l_1, \nu_1) \overset{d_1}{\rightsquigarrow} \ldots$ we have $\Delta(\pi) = \sum_{i \geq 0} d_i$

Timed CTL

For time-divergent paths $\pi \in (l_0, \nu_0) \overset{d_0}{\rightsquigarrow} (l_1, \nu_1) \overset{d_1}{\rightsquigarrow} \ldots$:

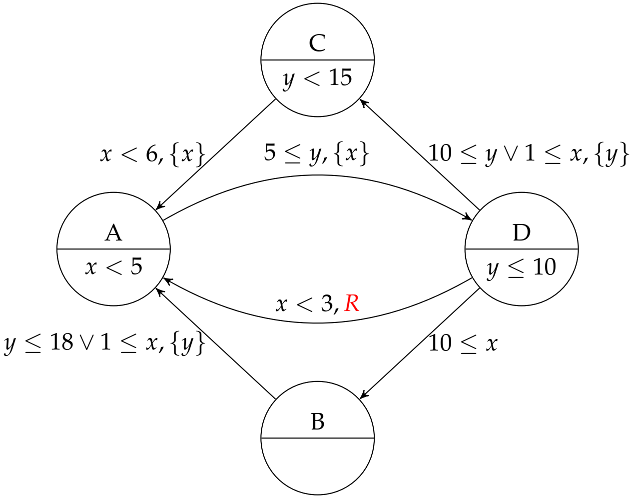
$$\pi \models \mathsf{F}_J\, \phi \text{ iff } \exists i \geq 0 : \ (l_i, \nu_i + d) \models \phi \text{ for some } d \in [0, d_i] \text{ with } \sum_{j=0}^{i-1} d_j + d \in J$$

Timed CTL

- TCTL semantics is also well-defined for Timed Automata suffering from timelocks
- A state is timelock-free if and only if it satisfies $\mathsf{E}\,\mathsf{G}\,\mathsf{true}$
  - some time-divergent path satisfies $\mathsf{G}\,\mathsf{true}$, i.e. there is at least one time-divergent path
  - note: for fair CTL, the states in which a fair path starts also satisfy $\mathsf{E}\,\mathsf{G}\,\mathsf{true}$
- A Timed Automaton $\mathcal{T}$ is timelock-free iff for all reachable $s \in [\mathcal{T}]$, $s \models \mathsf{E}\,\mathsf{G}\,\mathsf{true}$
- Timelocks can thus be checked by means of model checking

Outline

Exercise



Is the Timed Automaton Non-Zeno when:

- $R = \{x\}$
- $R = \{y\}$
- $R = \{x, y\}$

Is the Timed Automaton Timelock-free when:

- $R = \{x\}$
- $R = \{y\}$
- $R = \{x, y\}$

Explain and motivate your answers.