Verification

Lecture 18

Bernd Finkbeiner
Peter Faymonville
Michael Gerke

UNIVERSITÄT
DES
SAARLANDES

REVIEW: Timed automaton semantics

The transition relation $\rightarrow$ is defined by the following two rules:

- Discrete transition: $\langle \ell, v \rangle \xrightarrow{d} \langle \ell', v' \rangle$ if all following conditions hold:
    - there is an edge labeled $(g : \alpha, D)$ from location ℓ to ℓ' such that:
    - g is satisfied by v, i.e., $v \models g$
    - $v' = v$ with all clocks in D reset to 0, i.e., $v' =$ reset D in v
    - v' fulfills the invariant of location ℓ', i.e., $v' \models inv(\ell')$
- Delay transition: $\langle \ell, v \rangle \xrightarrow{\alpha} \langle \ell, v{+}d \rangle$ for positive real d
    - if for any $0 \leq d' \leq d$ the invariant of ℓ holds for $v{+}d'$, i.e. $v{+}d' \models inv(\ell)$

REVIEW: Timelock, time-divergence and Zenoness

- A timed automaton is only considered an adequate model of a time-critical system if it is:

  non-Zeno and timelock-free

- Time-convergent paths will be explicitly excluded from the analysis.

REVIEW: Timed CTL

Syntax of TCTL state-formulas over *AP* and set *C*:

$$\Phi ::= \text{true} \;\Big|\; a \;\Big|\; g \;\Big|\; \Phi \wedge \Phi \;\Big|\; \neg\Phi \;\Big|\; \mathsf{E}\,\varphi \;\Big|\; \mathsf{A}\,\varphi$$

where $a \in AP$, $g \in ACC(C)$ and φ is a path-formula defined by:

$$\varphi ::= \Phi\, \mathsf{U}^{J}\, \Phi$$

where $J \subseteq \mathbb{R}_{\geq 0}$ is an interval whose bounds are naturals
Forms of J: $[n, m]$, $(n, m]$, $[n, m)$ or (n, m) for $n, m \in \mathbb{N}$ and $n \leq m$

for right-open intervals, $m = \infty$ is also allowed

REVIEW: Semantics of TCTL

For state $s = \langle \ell, \eta \rangle$ in $TS(TA)$ the satisfaction relation $\models$ is defined by:

$s \models \text{true}$		
$s \models a$	iff	$a \in L(\ell)$
$s \models g$	iff	$\eta \models g$
$s \models \neg \Phi$	iff	not $s \models \Phi$
$s \models \Phi \wedge \Psi$	iff	$(s \models \Phi)$ and $(s \models \Psi)$
$s \models \mathrm{E}\,\varphi$	iff	$\pi \models \varphi$ for some $\pi \in Paths_{div}(s)$
$s \models \mathrm{A}\,\varphi$	iff	$\pi \models \varphi$ for all $\pi \in Paths_{div}(s)$

path quantification over time-divergent paths only

REVIEW: TCTL model checking

- TCTL model-checking problem: $TA \models \Phi$ for non-Zeno TA

$$\underbrace{TA \models \Phi}_{\text{timed automaton}} \quad \text{iff} \quad \underbrace{TS(TA) \models \Phi}_{\text{infinite state graph}}$$

- Idea: consider a finite region graph $RG(TA)$
- Transform TCTL formula Φ into an "equivalent" CTL-formula $\widehat{\Phi}$
- Then: $TA \models_{\text{TCTL}} \Phi \quad \text{iff} \quad \underbrace{RG(TA)}_{\text{finite state graph}} \models_{\text{CTL}} \widehat{\Phi}$

REVIEW: Eliminating timing parameters

- Eliminate all intervals $J \neq [0, \infty)$ from TCTL formulas
  - introduce a fresh clock, z say, that does not occur in TA
  - $s \models \mathrm{E}\,\Diamond^{J}\,\Phi$ iff reset z in $s \models \Diamond(z \in J \;\wedge\; \Phi)$
- Formally: for any state s of $TS(TA)$ it holds:

$$s \models \mathrm{E}\,\Phi\,\mathrm{U}^{J}\,\Psi \quad \text{iff} \quad \underbrace{s\{z := 0\}}_{\text{state in } TS(TA \oplus z)} \models \mathrm{E}\Big((\Phi \vee \Psi)\,\mathrm{U}\,(z \in J) \wedge \Psi\Big)$$

$$s \models \mathrm{A}\,\Phi\,\mathrm{U}^{J}\,\Psi \quad \text{iff} \quad \underbrace{s\{z := 0\}}_{\text{state in } TS(TA \oplus z)} \models \mathrm{A}\Big((\Phi \vee \Psi)\,\mathrm{U}\,(z \in J) \wedge \Psi\Big)$$

  - where $TA \oplus z$ is TA (over C) extended with $z \notin C$

REVIEW: Clock equivalence

Impose an equivalence, denoted $\cong$, on the clock valuations such that:

(A) Equivalent clock valuations satisfy the same clock constraints g in *TA* and Φ:

$$\eta \cong \eta' \;\Rightarrow\; \left(\eta \vDash g \quad \text{iff} \quad \eta' \vDash g\right)$$

- no diagonal clock constraints are considered
- all the constraints in *TA* and Φ are thus either of the form $x \leq c$ or $x < c$

(B) Time-divergent paths emanating from equivalent states are equivalent

- this property guarantees that equivalent states satisfy the same path formulas

(C) The number of equivalence classes under $\cong$ is finite

REVIEW: First observation

- $\eta \models x < c$ whenever $\eta(x) < c$, or equivalently, $\lfloor \eta(x) \rfloor < c$
  - $\lfloor d \rfloor = \max\{ c \in \mathbb{N} \mid c \leq d \}$ and $frac(d) = d - \lfloor d \rfloor$
- $\eta \models x \leq c$ whenever $\lfloor \eta(x) \rfloor < c$ or $\lfloor \eta(x) \rfloor = c$ and $frac(\eta(x)) = 0$

$\Rightarrow$ $\eta \models g$ only depends on $\lfloor \eta(x) \rfloor$, and whether $frac(\eta(x)) = 0$

- Initial suggestion: clock valuations η and η' are equivalent if:

$$\lfloor \eta(x) \rfloor = \lfloor \eta'(x) \rfloor \quad \text{and} \quad frac(\eta(x)) = 0 \text{ iff } frac(\eta'(x)) = 0$$

- Note: it is crucial that in $x < c$ and $x \leq c$, c is a natural

REVIEW: Second observation

- Consider location ℓ with $inv(\ell)$ = true and only outgoing transitions:
  - one guarded with $x \geq 2$ (action α) and $y > 1$ (action β)
- Let state $s = \langle \ell, \eta \rangle$ with $1 < \eta(x) < 2$ and $0 < \eta(y) < 1$
  - α and β are disabled, only time may elapse
- Transition that is enabled next depends on $x < y$ or $x \geq y$
  - e.g., if $frac(\eta(x)) \geq frac(\eta(y))$, action α is enabled first
- Suggestion for strengthening of initial proposal for all $x, y \in C$ by:

$$frac(\eta(x)) \leq frac(\eta(y)) \quad \text{if and only if} \quad frac(\eta'(x)) \leq frac(\eta'(y))$$

REVIEW: Final observation

- So far, clock equivalence yield a denumerable though not finite quotient
- For $TA \models \Phi$ only the clock constraints in TA and Φ are relevant
  - let $c_x \in \mathbb{N}$ the largest constant with which x is compared in TA or Φ

⇒ If $\eta(x) > c_x$ then the actual value of x is irrelevant

  - constraints on $\cong$ so far are only relevant for clock values of x (y) up to c_x (c_y)

Clock equivalence

Clock valuations $\eta, \eta' \in Eval(C)$ are <u>equivalent</u>, denoted $\eta \cong \eta'$, if:

(1) for any $x \in C$: $(\eta(x) > c_x) \wedge (\eta'(x) > c_x)$ or $(\eta(x) \leq c_x) \wedge (\eta'(x) \leq c_x)$

(2) for any $x \in C$: if $\eta(x), \eta'(x) \leq c_x$ then:

$$\lfloor \eta(x) \rfloor = \lfloor \eta'(x) \rfloor \quad \text{and} \quad frac(\eta(x)) = 0 \text{ iff } frac(\eta_2(x)) = 0$$

(3) for any $x, y \in C$: if $\eta(x), \eta'(x) \leq c_x$ and $\eta(y), \eta'(y) \leq c_y$, then:

$$frac(\eta(x)) \leq frac(\eta(y)) \quad \text{iff} \quad frac(\eta'(x)) \leq frac(\eta'(y)).$$

$$s \cong s' \quad \text{iff} \quad \ell = \ell' \quad \text{and} \quad \eta \cong \eta'$$

Clock equivalence is a bisimulation

Clock equivalence is a bisimulation equivalence over AP'

Regions

- The clock region of $\eta \in Eval(C)$, denoted $[\eta]$, is defined by:

$$[\eta] \;=\; \{\, \eta' \in Eval(C) \mid \eta \cong \eta' \,\}$$

- The state region of $s = \langle \ell, \eta \rangle \in TS(TA)$ is defined by:

$$[s] \;=\; \langle \ell, [\eta] \rangle \;=\; \{\, \langle s, \eta' \rangle \mid \eta' \in [\eta] \,\}$$

Number of regions

The number of clock regions is bounded from below and above by:

$$|C|! * \prod_{x \in C} c_x \;\leq\; \Big| \underbrace{Eval(C)/\cong}_{\text{number of regions}} \Big| \;\leq\; |C|! * 2^{|C|-1} * \prod_{x \in C} (2c_x + 2)$$

where for the upper bound it is assumed that $c_x \geq 1$ for any $x \in C$

the number of state regions is $|Loc|$ times larger

Preservation of atomic properties

1. For $\eta, \eta' \in Eval(C)$ such that $\eta \cong \eta'$:

$$\eta \vDash g \quad \text{if and only if} \quad \eta' \vDash g \text{ for any } g \in AP' \setminus AP$$

2. For $s, s' \in TS(TA)$ such that $s \cong s'$:

$$s \vDash a \quad \text{if and only if} \quad s' \vDash a \text{ for any } a \in AP'$$

where AP' includes all atomic propositions and atomic clock constraints in TA and Φ.

Unbounded and successor regions

- Clock region $r_\infty = \{ \eta \in Eval(C) \mid \forall x \in C.\, \eta(x) > c_x \}$ is unbounded
- r' is the successor (clock) region of r, denoted $r' = succ(r)$, if either:
  1. $r = r_\infty$ and $r = r'$, or
  2. $r \neq r_\infty$, $r \neq r'$ and $\forall \eta \in r$:

$$\exists d \in \mathbb{R}_{>0}.\ (\eta + d \in r' \quad \text{and} \quad \forall 0 \leq d' \leq d.\, \eta + d' \in r \cup r')$$

- The successor region: $succ(\langle \ell, r \rangle) = \langle \ell, succ(r) \rangle$

Region automaton

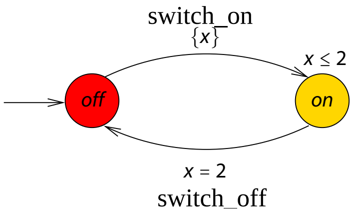
For non-Zeno TA with $TS(TA) = (S, Act, \rightarrow, I, AP, L)$ let:

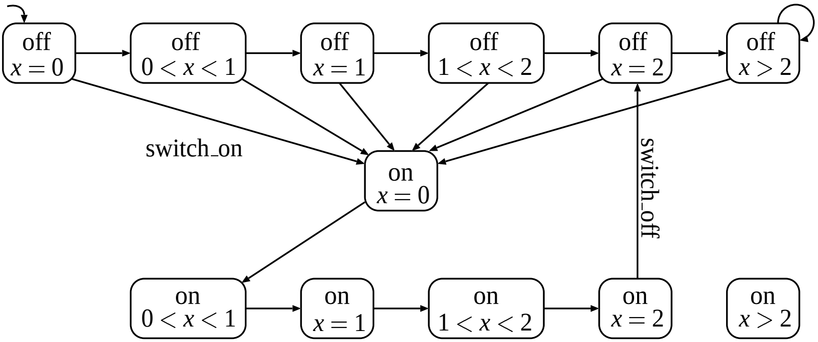
$$RG(TA, \Phi) = (S', Act \cup \{ \tau \}, \rightarrow', I, AP', L') \quad \text{with}$$

- $S' = S/\cong = \{ [s] \mid s \in S \}$ and $I' = \{ [s] \mid s \in I \}$, the state regions
- $L'(\langle \ell, r \rangle) = L(\ell) \cup \{ g \in AP' \setminus AP \mid r \models g \}$
- $\rightarrow'$ is defined by: $\dfrac{\ell \overset{g:\alpha,D}{\rightsquigarrow} \ell' \quad r \models g \quad \text{reset } D \text{ in } r \models inv(\ell')}{\langle \ell, r \rangle \xrightarrow{\alpha}' \langle \ell', \text{reset } D \text{ in } r \rangle}$ and

$$\frac{r \models inv(\ell) \quad succ(r) \models inv(\ell)}{\langle \ell, r \rangle \xrightarrow{\tau}' \langle \ell, succ(r) \rangle}$$

Example: simple light switch





Time convergence

For non-Zeno TA and $\pi = s_0\, s_1\, s_2 \ldots$ an initial, infinite path in $TS(TA)$:

(a) π is time-convergent $\Rightarrow$ $\exists$ state region $\langle \ell, r \rangle$ such that for some j:

$$s_i \in \langle \ell, r \rangle \text{ for all } i \geq j$$

(b) If $\exists$ state region $\langle \ell, r \rangle$ with $r \neq r_\infty$ and an index j such that:

$$s_i \in \langle \ell, r \rangle \text{ for all } i \geq j$$

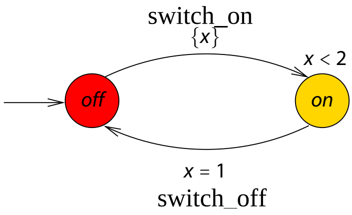
then π is time-convergent

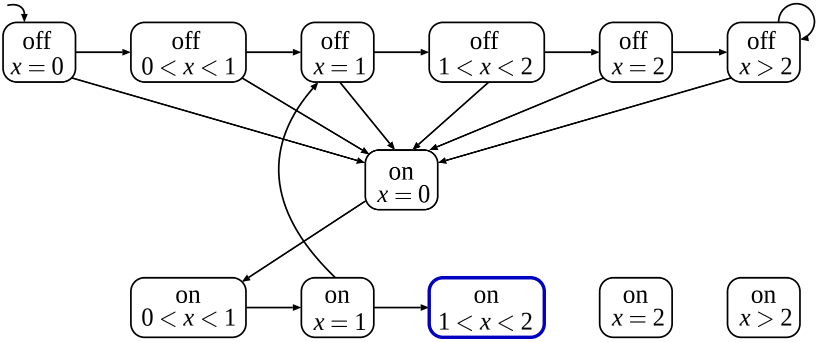
Timelock freedom

For non-Zeno TA:

TA is timelock-free iff no reachable state in $RG(TA)$ is terminal

Example


switch_on
{x}
x < 2
off
on
x = 1
switch_off


off
x = 0
off
0 < x < 1
off
x = 1
off
1 < x < 2
off
x = 2
off
x > 2
on
x = 0
on
0 < x < 1
on
x = 1
on
1 < x < 2
on
x = 2
on
x > 2

Correctness theorem

Let *TA* be a non-Zeno timed automaton and Φ a $\text{TCTL}_\diamond$ formula.
Then:

$$\underbrace{TA \models \Phi}_{\text{TCTL semantics}} \quad \text{iff} \quad \underbrace{RG(TA, \Phi) \models \Phi}_{\text{CTL semantics}}$$

Overview TCTL model checking

Require: timed automaton *TA* and TCTL formula Φ (both over *AP* and *C*)
Ensure: $TA \models \Phi$

$\widehat{\Phi}$:= eliminate the timing parameters from Φ;

determine the equivalence classes under $\cong$;

construct the region graph $TS = RG(TA)$;

apply the CTL model-checking algorithm to check $TS \models \widehat{\Phi}$;

$TA \models \Phi$ if and only if $TS \models \widehat{\Phi}$

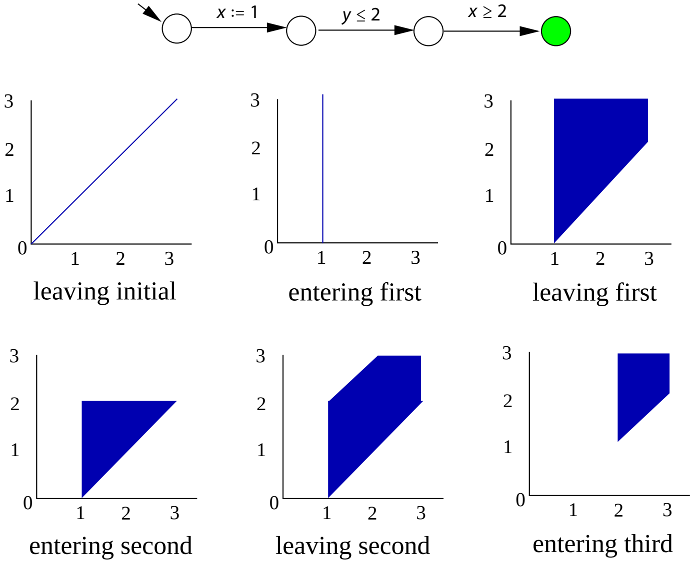
Other verification problems

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

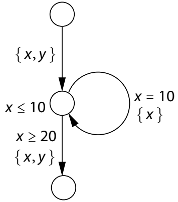
Zones

- Clock constraints are conjunctions of atomic constraints
  - $x \prec c$ and $x - y \prec c$ for $\prec \in \{<, \leq, =, \geq, >\}$
  - restrict to *TA* with only conjunctive clock constraints
  - and (as before) assume no difference clock constraints
- A clock zone is the set of clock valuations that satisfy a clock constraint
  - a clock zone for g is the maximal set of clock valuations satisfying g
- Clock zone of g: $[\![g]\!] = \{ \eta \in Eval(C) \mid \eta \models g \}$
  - use z, z' and so on to range over zones
- The state zone of $s = \langle \ell, \eta \rangle \in TS(TA)$ is $\langle \ell, z \rangle$ with $\eta \in z$

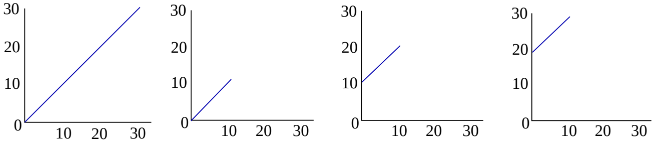
Zone automaton: intuition



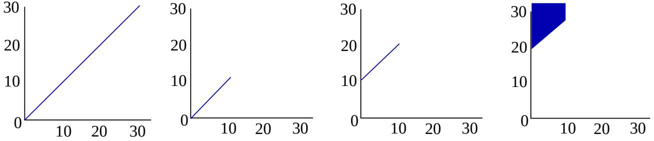
Normalization: intuition



symbolic semantics has infinitely many zones:



normalization yields a finite zone graph:



Successor and reset zones

- z' is the successor (clock) zone of z, denoted $z' = z^{\uparrow}$, if:
  - $z^{\uparrow} = \{ \eta + d \mid \eta \in z, d \in \mathbb{R}_{>0} \}$
- z' is the zone obtained from z by resetting clocks D:
  - $\text{reset } D \text{ in } z = \{ \text{reset } D \text{ in } \eta \mid \eta \in z \}$

Zone graph

For non-Zeno TA let:

$$ZG(TA, \Phi) = (Q, Q_0, E, L') \quad \text{with}$$

- $Q = Loc \times Zone(C)$ and $Q_0 = \{ \langle \ell, z_0 \rangle \mid \ell \in Loc_0 \}$
- $L(\langle \ell, z \rangle) = L(\ell) \cup \{ g \mid g \in z \}$
- E consists of two types of edges:
  - Discrete transitions: $\langle \ell, z \rangle \xrightarrow{\alpha} \langle \ell', \text{reset } D \text{ in } (z \wedge g) \wedge inv(\ell') \rangle$ if $\ell \overset{g:\alpha,D}{\rightsquigarrow} \ell'$, and
  - Delay transitions: $\langle \ell, z \rangle \xrightarrow{\tau} \langle \ell, z^{\uparrow} \wedge inv(\ell) \rangle$.

Correctness (1)

For timed automaton *TA* and any initial state $\langle \ell, \eta_0 \rangle$:

- Soundness:

$$\underbrace{\langle \ell, \underbrace{\{ \eta_0 \}}_{z_0} \rangle \to^* \langle \ell', z' \rangle}_{\text{in } ZG(TA)} \quad \text{implies} \quad \underbrace{\langle \ell, \eta_0 \rangle \to^* \langle \ell', \eta' \rangle}_{\text{in } TS(TA)} \text{ for all } \eta' \in z'$$

- Completeness:

$$\underbrace{\langle \ell, \eta_0 \rangle \to^* \langle \ell', \eta' \rangle}_{\text{in } TS(TA)} \text{ implies } \underbrace{\langle \ell, \{ \eta_0 \} \rangle \to^* \langle \ell', z' \rangle}_{\text{in } ZG(TA)} \text{ for some } z' \text{ with } \eta' \in z'$$

Zone normalization

- To obtain a finite representation, zone normalization is employed
- For zone z, $norm(z) = \{ \eta \mid \eta \cong \eta', \eta' \in z \}$
  - where $\cong$ is the clock equivalence
- There can only be finitely many normalized zones
- $\langle \ell, z \rangle \rightarrow_{norm} \langle \ell', norm(z') \rangle$ if $\langle \ell, z \rangle \rightarrow \langle \ell', z' \rangle$

Correctness (2)

For timed automaton *TA* and any initial state $\langle \ell, \eta \rangle$:

- Soundness:

$$\langle \ell, \{ \eta_0 \} \rangle \to^*_{norm} \langle \ell', z' \rangle \quad \text{implies} \quad \langle \ell, \eta_0 \rangle \to^* \langle \ell', \eta' \rangle$$

  - for all $\eta' \in z'$ such that $\forall x.\, \eta'(x) \leq c_x$

- Completeness:

$$\langle \ell, \eta_0 \rangle \to^* \langle \ell', \eta' \rangle \text{ with } \forall x.\, \eta'(x) \leq c_x \text{ implies } \langle \ell, \{ \eta_0 \} \rangle \to^*_{norm} \langle \ell', z' \rangle$$

  - for some z' such that $\eta' \in z'$

- Finiteness: the transition relation $\to_{norm}$ is finite

Forward reachability algorithm

```
Passed := ∅;                                          // explored states so far
Wait := { (ℓ₀, z₀) };                                 // states to be explored
while  Wait ≠ ∅                                       // still states to go
do select and remove (ℓ, z) from Wait;
     if (ℓ = goal ∧ z ∩ z_goal ≠ ∅) then return "reachable"! fi ;
     if ¬(∃(ℓ, z') ∈ Passed. z ⊆ z') // no "super"state explored yet
     then add (ℓ, z) to Passed                        // (ℓ, z) is a new state
          foreach (ℓ', z') with (ℓ, z) →_norm (ℓ', z')
          do add (ℓ', z') to Wait;                    // add symbolic successors
     fi
od
return "not reachable"!
```