

# Decidability of Timed Reachability

## Lecture #18 of Advanced Model Checking

*Joost-Pieter Katoen*

Lehrstuhl 2: Software Modeling & Verification

E-mail: `katoen@cs.rwth-aachen.de`

February 1, 2018

## Timelock, time-divergence and Zenoness

- A path is *time-divergent* if its execution time is infinite

$$ExecTime(s_0 \xrightarrow{d_0} s_1 \xrightarrow{d_1} \dots) = \sum_{i=0} d_i = \infty$$

- *TA* is *timelock-free* if no state in  $Reach(TS(TA))$  contains a timelock  
a state contains a timelock whenever no time-divergent paths emanate from it
- *TA* is *non-Zeno* if there does not exist an initial Zeno path in  $TS(TA)$   
a path is Zeno if it is time-convergent and performs infinitely many actions

## Some abbreviations

“Always” is obtained in the following way:

$$\exists \square^J \Phi = \neg \forall \diamond^J \neg \Phi \quad \text{and} \quad \forall \square^J \Phi = \neg \exists \diamond^J \neg \Phi$$

$\exists \square^J \Phi$  asserts that for some path during the interval  $J$ ,  $\Phi$  holds

$\forall \square^J \Phi$  requires this to hold for all paths

Standard  $\square$  and  $\diamond$ -operator are obtained as follows:

$$\diamond \Phi = \diamond^{[0, \infty)} \Phi \quad \text{and} \quad \square \Phi = \square^{[0, \infty)} \Phi$$

## The $\Longrightarrow$ relation

For infinite path fragments in  $TS(TA)$  performing  $\infty$  many actions let:

$$s_0 \xrightarrow{d_0} s_1 \xrightarrow{d_1} s_2 \xrightarrow{d_2} \dots \quad \text{with } d_0, d_1, d_2 \dots \geq 0$$

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

$$s_0 \underbrace{\xrightarrow{d_0^1} \dots \xrightarrow{d_0^{k_0}}}_{\substack{\text{time passage of} \\ d_0 \text{ time-units}}} s_0 + d_0 \xrightarrow{\alpha_1} s_1 \underbrace{\xrightarrow{d_1^1} \dots \xrightarrow{d_1^{k_1}}}_{\substack{\text{time passage of} \\ d_1 \text{ time-units}}} s_1 + d_1 \xrightarrow{\alpha_2} s_2 \underbrace{\xrightarrow{d_2^1} \dots \xrightarrow{d_2^{k_2}}}_{\substack{\text{time passage of} \\ d_2 \text{ time-units}}} s_2 + d_2 \xrightarrow{\alpha_3} \dots$$

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

For  $\pi \in s_0 \xrightarrow{d_0} s_1 \xrightarrow{d_1} \dots$  we have  $\text{ExecTime}(\pi) = \sum_{i \geq 0} d_i$

## Semantics of timed reachability

For time-divergent path  $\pi \in s_0 \xrightarrow{d_0} s_1 \xrightarrow{d_1} \dots$ , we have:

$\pi \models \diamond^J \Psi$  iff  $\exists i \geq 0. s_i + d \models \Psi$  for some  $d \in [0, d_i]$  with

$$\sum_{k=0}^{i-1} d_k + d \in J \quad \text{and}$$

where for  $s_i = \langle \ell_i, \eta_i \rangle$  and  $d \geq 0$  we have  $s_i + d = \langle \ell_i, \eta_i + d \rangle$

## Timed reachability for timed automata

- Let  $TA$  be a timed automaton with clocks  $C$  and locations  $Loc$
- The *satisfaction set*  $Sat(\forall \diamond^J \Phi)$  is defined by:

$$Sat(\forall \diamond^J \Phi) = \{ s \in Loc \times Eval(C) \mid \forall \pi \in Paths_{div}(s). \pi \models \diamond^J \Phi \}$$

The satisfaction set for  $\exists \diamond^J \Phi$  is defined analogously

- $TA$  satisfies  $\forall \diamond^J \Phi$  iff  $\forall \diamond^J \Phi$  holds in all initial states of  $TA$ :

$$TA \models \forall \diamond^J \Phi \quad \text{if and only if} \quad \forall \ell_0 \in Loc_0. \langle \ell_0, \eta_0 \rangle \models \forall \diamond^J \Phi$$

where  $\eta_0(x) = 0$  for all  $x \in C$

## Characterizing timelock

- TCTL semantics is also well-defined for  $TA$  with timelock
- A state has a timelock if no time-divergent paths emanate from it
- A state is *timelock-free* if and only if it satisfies  $\exists \square \text{true}$ 
  - some time-divergent path satisfies  $\square \text{true}$ , i.e., there is  $\geq 1$  time-divergent path
  - note: for fair CTL, the states in which a fair path starts also satisfy  $\exists \square \text{true}$
- $TA$  is timelock-free iff  $\forall s \in \text{Reach}(TS(TA)): s \models \exists \square \text{true}$
- Timelocks can thus be characterised by a timed reachability property

## Verifying timed reachability

- Timed reachability problem:  $TA \models \forall \diamond^J \Phi$  for non-Zeno  $TA$

$$\underbrace{TA \models \forall \diamond^J \Phi}_{\text{timed automaton}} \quad \text{iff} \quad \underbrace{TS(TA) \models \forall \diamond^J \Phi}_{\text{uncountable transition system}}$$

- Zeno paths are excluded as they could be false alarms
- Idea: take a finite quotient of  $TS(TA)$  wrt. a tailored bisimulation
  - $TS(TA) / \cong$  is a *region* transition system and denoted  $RTS(TA)$
- Transform  $\forall \diamond^J \Phi$  into an “equivalent” reachability property  $\forall \diamond \hat{\Phi}$
- Then:  $TA \models \forall \diamond^J \Phi$  iff  $\underbrace{RTS(TA)}_{\text{finite transition system}} \models \underbrace{\forall \diamond \hat{\Phi}}_{\text{CTL formula}}$



## Eliminating timing parameters

- Eliminate all intervals  $J \neq [0, \infty)$  from timed reachability
  - introduce a fresh clock,  $z$  say, that does not occur in  $TA$
- Formally: for any state  $s$  of  $TS(TA)$  it holds:

$$s \models \exists \diamond^J \Phi \quad \text{iff} \quad \underbrace{s\{z := 0\}}_{\text{state in } TS(TA \oplus z)} \models \exists \diamond((z \in J) \wedge \Phi)$$

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

atomic clock constraints are atomic propositions, i.e., a CTL formula results

## Correctness

Let  $TA = (Loc, Act, C, \hookrightarrow, Loc_0, Inv, AP, L)$ . For clock  $z \notin C$ , let

$$TA \oplus z = (Loc, Act, C \cup \{z\}, \hookrightarrow, Loc_0, Inv, AP, L).$$

For any state  $s$  of  $TS(TA)$  it holds that:

$$1. \quad s \models \exists \diamond^J \Psi \quad \text{iff} \quad \underbrace{s\{z := 0\}}_{\text{state in } TS(TA \oplus z)} \models \exists \diamond ((z \in J) \wedge \Psi)$$

$$2. \quad s \models \forall \diamond^J \Psi \quad \text{iff} \quad \underbrace{s\{z := 0\}}_{\text{state in } TS(TA \oplus z)} \models \forall \diamond ((z \in J) \wedge \Psi)$$

## Constraints on clock equivalence $\cong$

(A) Equivalent clock valuations satisfy the same clock constraints  $g$ :

$$\eta \cong \eta' \Rightarrow (\eta \models g \text{ iff } \eta' \models g)$$

(B) Time-divergent paths of 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

## Clock equivalence

- Correctness criteria (A) and (B) are ensured if equivalent states:
  - agree on the integer parts of all clock values, and
  - agree on the ordering of the fractional parts of all clocks
- ⇒ This yields a denumerable infinite set of equivalence classes
- Observe that:
  - if clocks exceed the maximal constant with which they are compared their precise value is not of interest
- ⇒ The number of equivalence classes is then finite (C)

## Clock equivalence: definition

Clock valuations  $\eta, \eta' \in Eval(C)$  are *equivalent*, denoted  $\eta \cong \eta'$ , if for all  $x, y \in C$  it holds:

1.  $\eta(x) > c_x$  if and only if  $\eta'(x) > c_x$
2.  $\eta(x) \leq c_x$  implies  $\lfloor \eta(x) \rfloor = \lfloor \eta'(x) \rfloor$
3.  $\eta(x) \leq c_x$  implies  $frac(\eta(x)) = 0$  iff  $frac(\eta'(x)) = 0$
4.  $\eta(x) \leq c_x$  and  $\eta(y) \leq c_y$  implies:

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

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

## 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:

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

**Example**  $c_x=2, c_y=1$

## Bounds on the number of regions

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

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

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



# Proof

## Preservation of atomic properties

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

$\eta \models g$  if and only if  $\eta' \models g$  for any  $g \in ACC(TA \cup \Phi)$

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

$s \models a$  if and only if  $s' \models a$  for any  $a \in AP'$

where  $AP'$  includes all propositions in  $TA$  and atomic clock constraints in  $TA$  and  $\Phi$

## Clock equivalence is a bisimulation

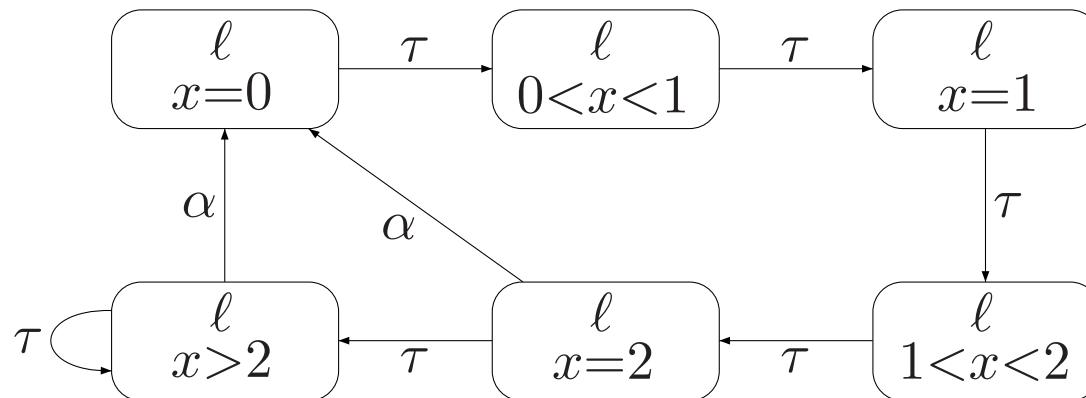
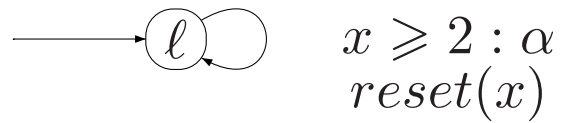
Clock equivalence is a bisimulation equivalence over  $AP'$

# Proof

## Region automaton: intuition

- Region automaton = quotient of  $TS(TA)$  under  $\cong$
- State regions are states in quotient transition system under  $\cong$
- Transitions in region automaton “mimic” those in  $TS(TA)$
- Delays are **abstract**
  - the exact delay is not recorded, only that some delay took place
  - if any clock  $x$  exceeds  $c_x$ , delays are self-loops
- Discrete transitions correspond to **actions**

## A simple example



## 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:  $succ(\langle \ell, r \rangle) = \langle \ell, succ(r) \rangle$
- Note: the location invariants are ignored so far!

## Characterizing time convergence

For non-zero  $TA$  and  $\pi = s_0 s_1 s_2 \dots$  a path in  $TS(TA)$ :

(a)  $\pi$  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  $\pi$  is *time-convergent*

time-convergent paths are paths that only perform delays from some time instant on



## Region automaton

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

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

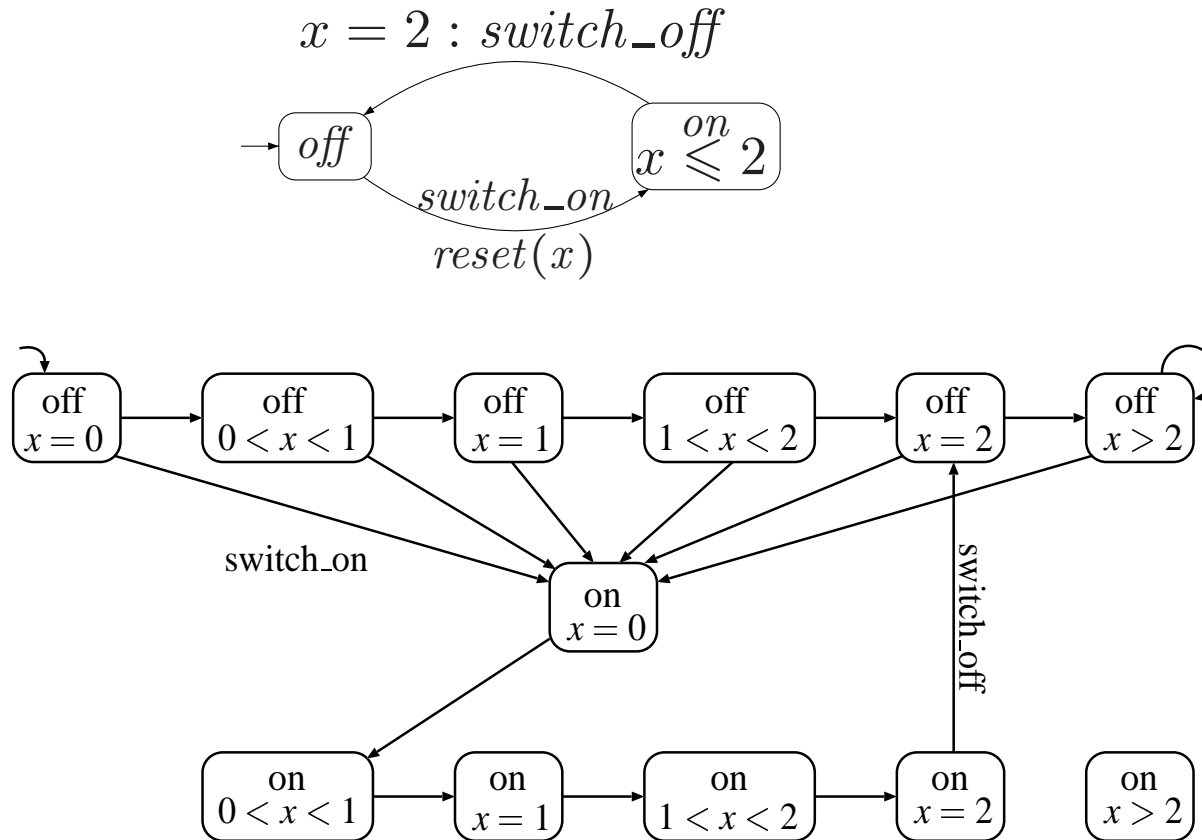
- $S' = S / \cong = \{[s] \mid s \in S\}$  and  $I' = \{[s] \mid s \in I\}$ , the state regions

- $L'(\langle \ell, r \rangle) = L(\ell) \cup \{g \in AP' \setminus AP \mid r \models g\}$

- $\rightarrow'$  is defined by: 
$$\frac{\ell \xrightarrow{g:\alpha,D} \ell' \quad r \models g \quad \text{reset } D \text{ in } r \models Inv(\ell')}{\langle \ell, r \rangle \xrightarrow{\alpha}' \langle \ell', \text{reset } D \text{ in } r \rangle}$$

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



## Correctness theorem [Alur and Dill, 1989]

For non-Zeno timed automaton  $TA$  and timed reachability property  $\forall \diamond^J \Phi$ :

$$TA \models \forall \diamond^J \Phi \quad \text{iff} \quad RTS(TA, \Phi) \models \forall \widehat{\Phi}$$

## Characterizing timelock freedom

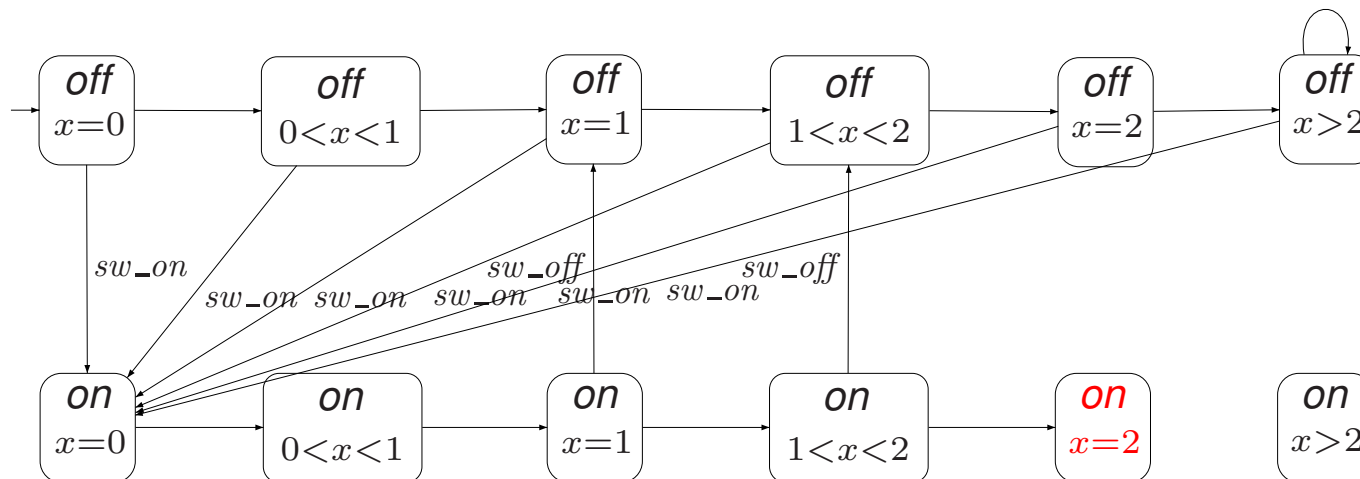
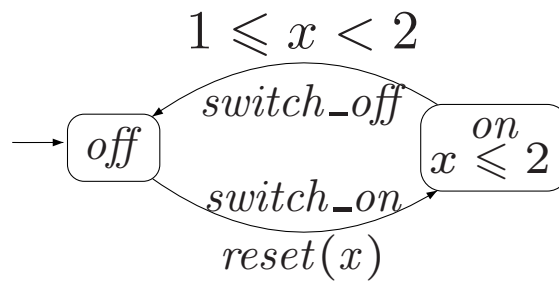
Non-Zeno  $TA$  is timelock-free

iff

$RTS(TA)$  has no reachable terminal states

timelocks can thus be checked by a reachability analysis of  $RTS(TA)$

# Example



# Time complexity

Model checking timed reachability on TA is **PSPACE-complete**

## Other verification problems

1. The timed CTL model-checking problem is **PSPACE-complete**
2. Model checking safety, or  $\omega$ -regular properties on TA is **PSPACE-complete**
3. Model checking LTL and CTL against TA is **PSPACE-complete**
4. The model-checking problem for timed LTL is **undecidable**
5. The satisfaction problem for timed CTL is **undecidable**

*all facts without proof*