Timed CTL Model Checking

Lecture #17 of Advanced Model Checking

Joost-Pieter Katoen

Lehrstuhl 2: Software Modeling & Verification

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

June 25, 2012

Timelock, time-divergence and Zenoness

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

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

- *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

Timed CTL

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

$$\Phi ::= \text{true} \;\Big|\; a \;\Big|\; g \;\Big|\; \Phi \wedge \Phi \;\Big|\; \neg\Phi \;\Big|\; \exists\varphi \;\Big|\; \forall\varphi$$

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

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

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

abbreviate $[c, \infty)$ by $x > c$, $(c_1, c_2]$ by $c_1 < x \leqslant c_2$ etc.

Some abbreviations

"Always" is obtained in the following way:

$$\exists\Box^{J}\,\Phi \;=\; \neg\forall\Diamond^{J}\,\neg\Phi \quad \text{and} \quad \forall\Box^{J}\,\Phi \;=\; \neg\exists\Diamond^{J}\,\neg\Phi$$

$\exists\Box^{J}\,\Phi$ asserts that for some path during the interval J, Φ holds

$\forall\Box^{J}\,\Phi$ requires this to hold for all paths

Standard $\Box$ and $\Diamond$-operator are obtained as follows:

$$\Diamond\,\Phi = \Diamond^{[0,\infty)}\,\Phi \quad \text{and} \quad \Box\,\Phi = \Box^{[0,\infty)}\,\Phi$$

Timed properties in TCTL

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 $(s \models \Psi)$
$s \models \exists\varphi$	iff	$\pi \models \varphi$ for some $\pi \in \mathit{Paths}_{div}(s)$
$s \models \forall\varphi$	iff	$\pi \models \varphi$ for all $\pi \in \mathit{Paths}_{div}(s)$

path quantification over time-divergent paths only

The $\Longrightarrow$ relation

For infinite path fragments in *TS*$($*TA*$)$ performing ∞ many actions let:

$$s_0 \overset{d_0}{\Longrightarrow} s_1 \overset{d_1}{\Longrightarrow} s_2 \overset{d_2}{\Longrightarrow} \ldots \qquad \text{with } d_0, d_1, d_2 \ldots \geqslant 0$$

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

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

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

For $\pi \in s_0 \overset{d_0}{\Longrightarrow} s_1 \overset{d_1}{\Longrightarrow} \ldots$ we have $\mathit{ExecTime}(\pi) = \sum_{i \geqslant 0} d_i$

Semantics of TCTL

For time-divergent path $\pi \in s_0 \overset{d_0}{\Longrightarrow} s_1 \overset{d_1}{\Longrightarrow} \ldots$, we have:

$$\pi \models \Diamond^J \Psi \quad \text{iff} \quad \exists\, i \geqslant 0.\, s_i{+}d \models \Psi \text{ for some } d \in [0, d_i] \text{ with}$$

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

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

TCTL-semantics for timed automata

- Let *TA* be a timed automaton with clocks C and locations *Loc*
- For TCTL-state-formula Φ, the *satisfaction set* *Sat*(Φ) is defined by:

$$\mathit{Sat}(\Phi) \; = \; \{\, s \in \mathit{Loc} \times \mathit{Eval}(C) \mid s \models \Phi \,\}$$

- *TA* satisfies TCTL-formula Φ iff Φ holds in all initial states of *TA*:

$$\mathit{TA} \models \Phi \quad \text{if and only if} \quad \forall \ell_0 \in \mathit{Loc}_0.\, \langle \ell_0, \eta_0 \rangle \models \Phi$$

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

Characterizing timelock

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

TCTL model checking

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

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

- Idea: consider a finite quotient of $TS(TA)$ wrt. a bisimulation
  - $TS(TA)/\cong$ is a *region* transition system and denoted $RTS(TA)$
  - dependence on Φ is ignored for the moment . . .
- Transform TCTL formula Φ into an "equivalent" CTL-formula $\widehat{\Phi}$
- Then: $TA \models_{\text{TCTL}} \Phi$ iff $\underbrace{RTS(TA)}_{\text{finite transition system}} \models_{\text{CTL}} \widehat{\Phi}$

Basic recipe of TCTL model checking

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

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

determine the equivalence classes under $\cong$;

construct the region transition system $TS = RTS(TA)$;

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

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

how does clock equivalence look like?

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*
- Formally: for any state s of $TS(TA)$ it holds:

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

  - 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* $= (\mathit{Loc}, \mathit{Act}, C, \hookrightarrow, \mathit{Loc}_0, \mathit{Inv}, \mathit{AP}, L)$. For clock $z \notin C$, let

$$\mathit{TA} \oplus z \;=\; (\mathit{Loc}, \mathit{Act}, C \cup \{\, z \,\}, \hookrightarrow, \mathit{Loc}_0, \mathit{Inv}, \mathit{AP}, L).$$

For any state s of $\mathit{TS}(\mathit{TA})$ it holds that:

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

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

Clock equivalence $\cong$

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

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

(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 $\eta, \eta' \in \mathit{Eval}(C)$ are *equivalent*, denoted $\eta \cong \eta'$, if either:

- for all $x \in C$: $\eta(x) > c_x$ iff $\eta'(x) > c_x$, or
- for any $x, y \in C$ with $\eta(x) \leqslant c_x$ and $\eta(y) \leqslant c_y$ it holds:
  - $\lfloor \eta(x) \rfloor = \lfloor \eta'(x) \rfloor$ and $\mathit{frac}(\eta(x)) = 0$ iff $\mathit{frac}(\eta'(x)) = 0$, and
  - $\mathit{frac}(\eta(x)) \leqslant \mathit{frac}(\eta(y))$ iff $\mathit{frac}(\eta'(x)) \leqslant \mathit{frac}(\eta'(y))$.

$s \cong s'$ iff $\ell = \ell'$ and $\eta \cong \eta'$

Regions

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

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

- The *state region* of $s = \langle \ell, \eta \rangle \in \mathit{TS}(\mathit{TA})$ 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:

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

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

the number of state regions is $|\textit{Loc}|$ times larger

Proof

Preservation of atomic properties

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

$$\eta \models g \quad \text{if and only if} \quad \eta' \models g \text{ for any } g \in \mathit{ACC}(\mathit{TA} \cup \Phi)$$

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

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

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

Clock equivalence is a bisimulation

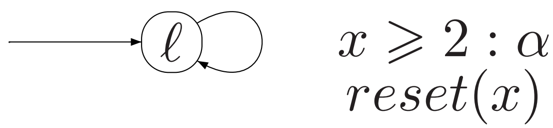
Clock equivalence is a bisimulation equivalence over AP'

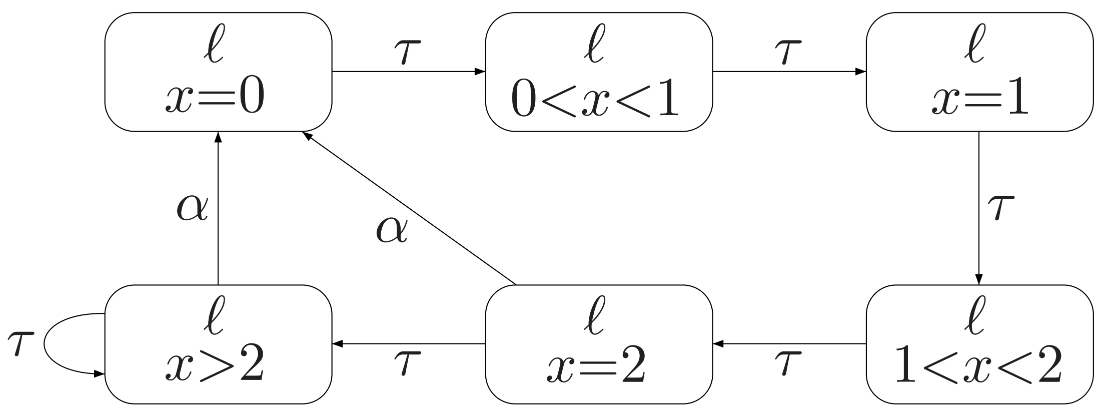
Proof

Region automaton: intuition

- Region automaton = quotient of $\mathit{TS}(\mathit{TA})$ under $\cong$
- State regions are states in quotient transition system under $\cong$
- Transitions in region automaton "mimic" those in $\mathit{TS}(\mathit{TA})$
- 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 $r_\infty = \{ \eta \in \mathit{Eval}(C) \mid \forall x \in C.\, \eta(x) > c_x \}$ is *unbounded*
- r' is the *successor (clock) region* of r, denoted $r' = \mathit{succ}(r)$, if either:
  1. $r = r_\infty$ and $r = r'$, or
  2. $r \neq r_\infty$, $r \neq r'$ and $\forall \eta \in r$:

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

- The *successor region*: $\mathit{succ}(\langle \ell, r \rangle) \ = \ \langle \ell, \mathit{succ}(r) \rangle$
- Note: the location invariants are ignored so far!

Example

Characterizing time convergence

For non-zeno *TA* and $\pi = s_0\, s_1\, s_2 \ldots$ a path in *TS*(*TA*):

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

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

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

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

then π is *time-convergent*

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

Region automaton

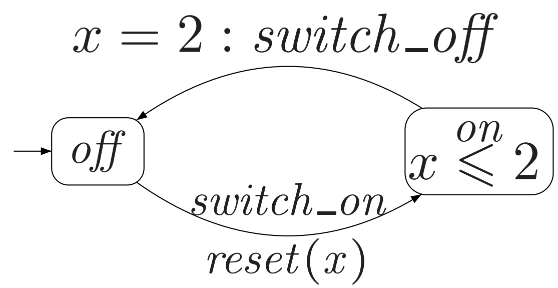
For non-zeno *TA* with $\mathit{TS}(\mathit{TA}) = (S, \mathit{Act}, \rightarrow, I, \mathit{AP}, L)$ let:

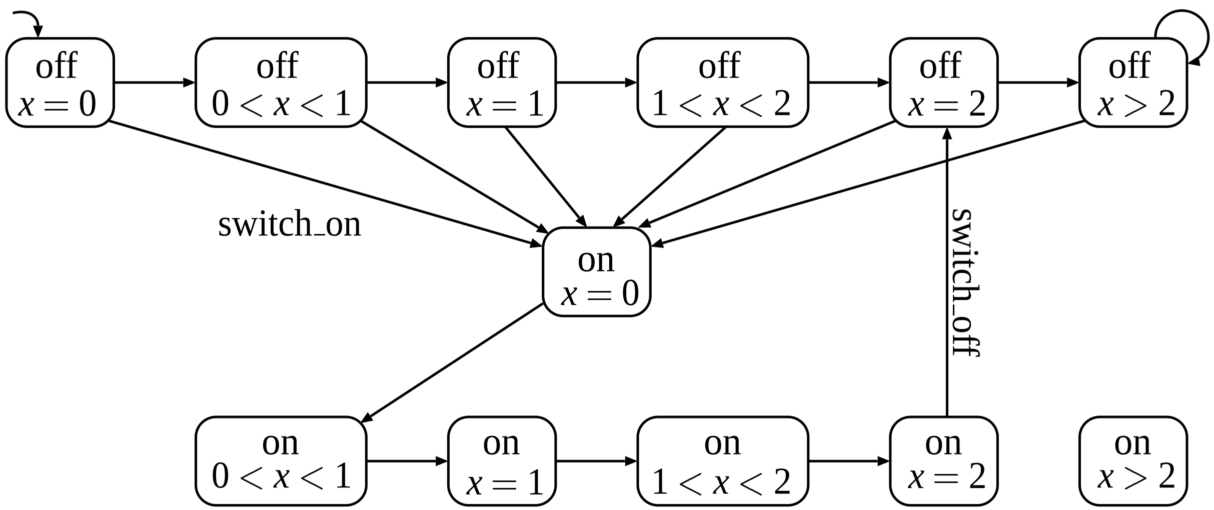
$$\mathit{RTS}(\mathit{TA}, \Phi) = (S', \mathit{Act} \cup \{\, \tau \,\}, \rightarrow', I, \mathit{AP}', L') \quad \text{with}$$

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

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

Example: simple light switch





Correctness theorem [Alur and Dill, 1989]

For non-Zeno timed automaton *TA* and TCTL$_\diamond$ formula Φ:

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

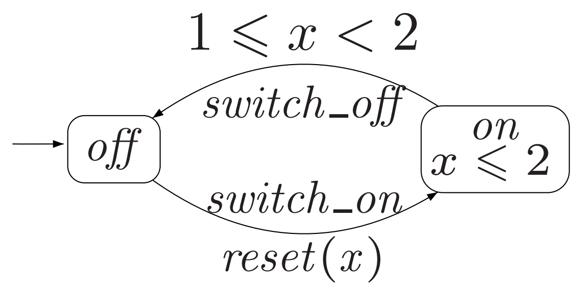
Proof

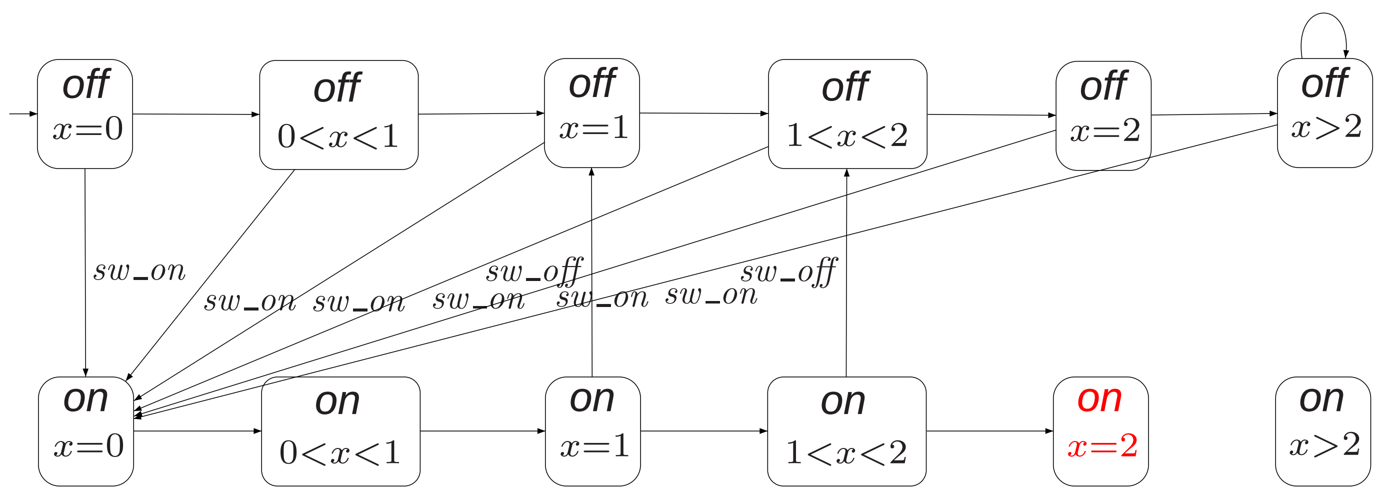
Characterizng timelock freedom

Non-zeno TA is timelock-free iff no reachable state in $RTS(TA)$ is terminal

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

Example


1 ⩽ x < 2
switch_off
off
on
x ⩽ 2
switch_on
reset(x)


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

Time complexity

For timed automaton *TA* and TCTL$_\Diamond$ formula Φ, the model-checking problem

TA $\models \Phi$ can be determined in time $\mathcal{O}\left((N+K) \cdot |\,\Phi\,|\right)$,

where N and K are the number of states and transitions in *RTS*(*TA*, Φ)

Other verification problems

1. The TCTL model-checking problem is PSPACE-complete
2. Model checking safety, reachability, or ω-regular properties in 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 TCTL is undecidable

all facts without proof