Verification

Lecture 17

Bernd Finkbeiner
Peter Faymonville
Michael Gerke

UNIVERSITÄT
DES
SAARLANDES

REVIEW: Timed automaton

A timed automaton is a tuple

$$TA \ = \ \left(Loc, Act, C, \rightsquigarrow, Loc_0, inv, AP, L\right) \quad \text{where:}$$

- Loc is a finite set of locations.
- $Loc_0 \subseteq Loc$ is a set of initial locations
- C is a finite set of clocks
- $L : Loc \rightarrow 2^{AP}$ is a labeling function for the locations
- $\rightsquigarrow \subseteq \ Loc \times CC(C) \times Act \times 2^C \times Loc$ is a transition relation, and
- $inv : Loc \rightarrow CC(C)$ is an invariant-assignment function

REVIEW: Clock constraints

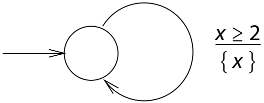
- Clock constraints over set C of clocks are defined by:

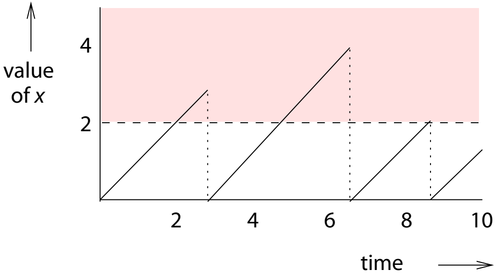
  $$g ::= \text{true} \mid x < c \mid x - y < c \mid x \leq c \mid x - y \leq c \mid \neg g \mid g \wedge g$$

  - where $c \in \mathbb{N}$ and clocks $x, y \in C$
  - rational constants would do; neither reals nor addition of clocks!
  - let $CC(C)$ denote the set of clock constraints over C
  - shorthands: $x \geq c$ denotes $\neg (x < c)$ and $x \in [c_1, c_2)$ or $c_1 \leq x < c_2$ denotes $\neg(x < c_1) \wedge (x < c_2)$
- Atomic clock constraints do not contain true, $\neg$ and $\wedge$
  - let $ACC(C)$ denote the set of atomic clock constraints over C
- Simplification: In the following, we assume constraints are diagonal-free, i.e., do neither contain $x - y \leq c$ nor $x - y < c$.

REVIEW: Guards versus location invariants

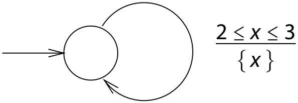
The effect of a lowerbound guard:

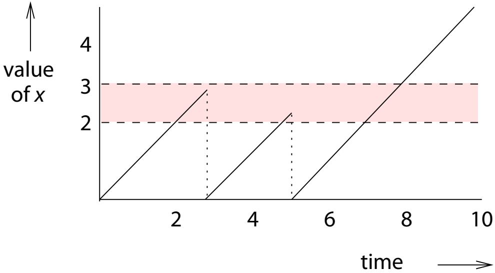




REVIEW: Guards versus location invariants

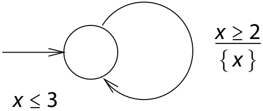
The effect of a lowerbound and upperbound guard:

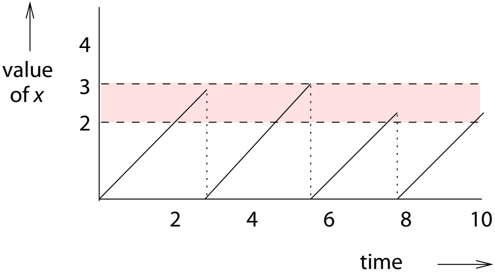




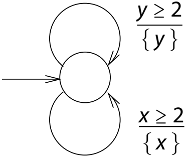
REVIEW: Guards versus location invariants

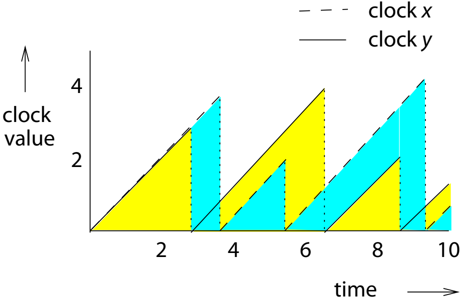
The effect of a guard and an invariant:





Arbitrary clock differences





Composing timed automata

Let $TA_i \ = \ \left(Loc_i, Act_i, C_i, \rightsquigarrow_i, Loc_{0,i}, inv_i, AP, L_i\right)$ and H an action-set

$TA_1 \parallel_H TA_2 = \left(Loc, Act_1 \cup Act_2, C, \rightsquigarrow, Loc_0, inv, AP, L\right)$ where:

- $Loc = Loc_1 \times Loc_2$ and $Loc_0 = Loc_{0,1} \times Loc_{0,2}$ and $C = C_1 \cup C_2$
- $inv(\langle \ell_1, \ell_2 \rangle) = inv_1(\ell_1) \ \wedge \ inv_2(\ell_2)$ and $L(\langle \ell_1, \ell_2 \rangle) = L_1(\ell_1) \cup L_2(\ell_2)$
- $\rightsquigarrow$ is defined by the inference rules:

for $\alpha \in H$
$$\frac{\ell_1 \overset{g_1:\alpha,D_1}{\rightsquigarrow_1} \ell_1' \ \wedge \ \ell_2 \overset{g_2:\alpha,D_2}{\rightsquigarrow_2} \ell_2'}{\langle \ell_1, \ell_2 \rangle \overset{g_1 \wedge g_2:\alpha,D_1 \cup D_2}{\rightsquigarrow} \langle \ell_1', \ell_2' \rangle}$$

for $\alpha \notin H$: $\dfrac{\ell_1 \overset{g:\alpha,D}{\rightsquigarrow_1} \ell_1'}{\langle \ell_1, \ell_2 \rangle \overset{g:\alpha,D}{\rightsquigarrow} \langle \ell_1', \ell_2 \rangle}$ and $\dfrac{\ell_2 \overset{g:\alpha,D}{\rightsquigarrow_2} \ell_2'}{\langle \ell_1, \ell_2 \rangle \overset{g:\alpha,D}{\rightsquigarrow} \langle \ell_1, \ell_2' \rangle}$

Clock valuations

- A clock valuation v for set C of clocks is a function $v : C \longrightarrow \mathbb{R}_{\geq 0}$
  - assigning to each clock $x \in C$ its current value $v(x)$
- Clock valuation $v{+}d$ for $d \in \mathbb{R}_{\geq 0}$ is defined by:
  - $(v{+}d)(x) = v(x) + d$ for all clocks $x \in C$
- Clock valuation reset x in v for clock x is defined by:

$$(\text{reset } x \text{ in } v)(y) = \begin{cases} v(y) & \text{if } y \neq x \\ 0 & \text{if } y = x. \end{cases}$$

  - reset x in (reset y in v) is abbreviated by reset x, y in v

Timed automaton semantics

For timed automaton $TA = \big(Loc, Act, C, \rightsquigarrow, Loc_0, inv, AP, L\big)$:
Transition system $TS(TA) = (S, Act', \rightarrow, I, AP', L')$ where:

- $S = Loc \times val(C)$, state $s = \langle \ell, v \rangle$ for location ℓ and clock valuation v
- $Act' = Act \cup \mathbb{R}_{\geq 0}$, (discrete) actions and time passage actions
- $I = \{ \langle \ell_0, v_0 \rangle \mid \ell_0 \in Loc_0 \ \wedge \ v_0(x) = 0 \text{ for all } x \in C \}$
- $AP' = AP \cup ACC(C)$
- $L'(\langle \ell, v \rangle) = L(\ell) \cup \{ g \in ACC(C) \mid v \models g \}$
- $\rightarrow$ is the transition relation defined on the next slide

Timed automaton semantics

The transition relation → 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. $v{+}d' \models inv(\ell)$

Time divergence

- Let for any $t < d$, for fixed $d \in \mathbb{R}_{>0}$, clock valuation $\eta + t \models inv(\ell)$
- A possible execution fragment starting from the location ℓ is:

$$\langle \ell, \eta \rangle \xrightarrow{d_1} \langle \ell, \eta + d_1 \rangle \xrightarrow{d_2} \langle \ell, \eta + d_1 + d_2 \rangle \xrightarrow{d_3} \langle \ell, \eta + d_1 + d_2 + d_3 \rangle \xrightarrow{d_4} \ldots$$

  - where $d_i > 0$ and the infinite sequence $d_1 + d_2 + \ldots$ converges towards d
  - such path fragments are called time-convergent

  $\Rightarrow$ time advances only up to a certain value

- Time-convergent execution fragments are unrealistic and ignored
  - much like unfair paths (as we will see later on)

Time divergence

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

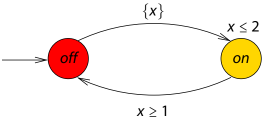
$$ExecTime(\tau) = \begin{cases} 0 & \text{if } \tau \in Act \\ d & \text{if } \tau = d \in \mathbb{R}_{>0} \end{cases}$$

- For infinite execution fragment $\rho = s_0 \xrightarrow{\tau_1} s_1 \xrightarrow{\tau_2} s_2 \ldots$ in $TS(TA)$ let:

$$ExecTime(\rho) = \sum_{i=0}^{\infty} ExecTime(\tau_i)$$

  - for path fragment π in $TS(TA)$ induced by ρ: $ExecTime(\pi) = ExecTime(\rho)$
- For state s in $TS(TA)$: $Paths_{div}(s) = \{ \pi \in Paths(s) \mid \pi \text{ is time-divergent} \}$

Example: light switch



The path π in $TS(Switch)$ in which on- and of-periods of one minute alternate:

$$\pi = \langle off, 0\rangle \langle off, 1\rangle \langle on, 0\rangle \langle on, 1\rangle \langle off, 1\rangle \langle off, 2\rangle \langle on, 0\rangle \langle on, 1\rangle \langle off, 1\rangle \ldots$$

is time-divergent as $ExecTime(\pi) = 1 + 1 + 1 + \ldots = \infty$.
The path:

$$\pi' = \langle off, 0\rangle \langle off, 1/2\rangle \langle off, 3/4\rangle \langle off, 7/8\rangle \langle off, 15/16\rangle \ldots$$

is time-convergent, since $ExecTime(\pi') = \sum\limits_{i \geq 1} \left(\frac{1}{2}\right)^i = 1 < \infty$

Timelock

- State $s \in TS(TA)$ contains a timelock if $Paths_{div}(s) = \varnothing$
  - there is no behavior in s where time can progress ad infinitum
  - clearly: any terminal state contains a timelock (but also non-terminal states may do)
  - terminal location does not necessarily yield a state with timelock (e.g. inv = true)
- TA is timelock-free if no state in $Reach(TS(TA))$ contains a timelock
- Timelocks are considered as modeling flaws that should be avoided

Zenoness

- A *TA* that performs infinitely many actions in finite time is Zeno
- Path π in $TS(TA)$ is Zeno if:
  - it is time-convergent, and
  - infinitely many actions $\alpha \in Act$ are executed along π
- *TA* is non-Zeno if there does not exist an initial Zeno path in $TS(TA)$
  - any π in $TS(TA)$ is time-divergent or
  - is time-convergent with nearly all (i.e., all except for finitely many) transitions being delay transitions
- Zeno paths are considered as modeling flaws that should be avoided

A sufficient criterion for Non-Zenoness

Let *TA* with set C of clocks such that for every control cycle:

$$\ell_0 \overset{g_1:\alpha_1,C_1}{\rightsquigarrow} \ell_1 \overset{g_2:\alpha_2,C_2}{\rightsquigarrow} \ldots \overset{g_n:\alpha_n,C_n}{\rightsquigarrow} \ell_n$$

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

1. $x \in C_i$ for some $0 < i \leq n$, and
2. there exists a constant $c \in \mathbb{N}_{>0}$ such that for all clock evaluations η:

$$\eta(x) < c \text{ implies } (\eta \not\models g_j \text{ or } \eta \not\models inv(\ell_j)), \text{ for some } 0 < j \leq n$$

Then: *TA* is non-Zeno

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.

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

Some abbreviations

- $\Diamond^J \Phi \;=\; \text{true}\,\mathsf{U}^J\,\Phi$
- $\mathsf{E}\,\Box^J\,\Phi \;=\; \neg\mathsf{A}\,\Diamond^J\,\neg\Phi$ and $\mathsf{A}\,\Box^J\,\Phi \;=\; \neg\mathsf{E}\,\Diamond^J\,\neg\Phi$
- $\Diamond\,\Phi = \Diamond^{[0,\infty)}\,\Phi$ and $\Box\,\Phi = \Box^{[0,\infty)}\,\Phi$

Semantics of TCTL

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

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

path quantification over time-divergent paths only

The $\Longrightarrow$ relation

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

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

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

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

where $k_i \in \mathbb{N}$, $d_i \in \mathbb{R}_{\geq 0}$ and $\alpha_i \in Act$ such that $\sum_{j=1}^{k_i} d_i^j = d_i$.
Notation: $s_i{+}d = \langle \ell_i, \eta_i{+}d \rangle$ where $s_i = \langle \ell_i, \eta_i \rangle$.

Semantics of TCTL

For time-divergent path $\pi \in s_0 \xrightarrow{d_0} s_1 \xrightarrow{d_1} \ldots$:

$\pi \vDash \Phi \, \mathsf{U}^{J} \, \Psi$

iff

$\exists\, i \geq 0.\, s_i{+}d \vDash \Psi$ for some $d \in [0, d_i]$ with $\sum_{k=0}^{i-1} d_k + d \in J$
and
$\forall j \leq i.\, s_j{+}d' \vDash \Phi \vee \Psi$ for every $d' \in [0, d_j]$ with $\sum_{j=0}^{j-1} d_k + d' \leq \sum_{k=0}^{i-1} d_k + d$

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 $Sat(\Phi)$ is defined by:

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

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

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

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

Timed CTL versus CTL

- Due to ignoring time-convergent paths in TCTL semantics, possibly:

$$\underbrace{TS(TA) \models_{\text{TCTL}} \mathsf{A}\,\varphi}_{\text{TCTL semantics}} \quad \text{but} \quad \underbrace{TS(TA) \not\models_{\text{CTL}} \mathsf{A}\,\varphi}_{\text{CTL semantics}}$$

    - CTL semantics considers all paths, timed CTL only time-divergent paths

- For $\Phi = \mathsf{A}\,\Box\left(on \longrightarrow \mathsf{A}\,\Diamond\, off\right)$ and the light switch

$$TS(Switch) \models_{\text{TCTL}} \Phi \quad \text{whereas} \quad TS(TA) \not\models_{\text{CTL}} \Phi$$

    - there are time-convergent paths on which location *on* is never left

Characterizing timelock

- TCTL semantics is also well-defined for *TA* with timelock
- A state is timelock-free if and only if it satisfies $\mathrm{E}\ \Box\ \mathrm{true}$
  - some time-divergent path satisfies $\Box$true, i.e., there is ≥ 1 time-divergent path
  - note: for fair CTL, the states in which a fair path starts also satisfy $\mathrm{E}\ \Box$ true
- TA is timelock-free iff $\forall s \in Reach(TS(TA)){:}\ s \models \mathrm{E}\ \Box\ \mathrm{true}$
- Timelocks can thus be checked by model checking

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: $TA \models_{\text{TCTL}} \Phi$ iff $\underbrace{RG(TA)}_{\text{finite state graph}} \models_{\text{CTL}} \widehat{\Phi}$

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
  - $s \vDash \mathsf{E}\, \Diamond^{J}\, \Phi$ iff reset z in $s \vDash \Diamond(z \in J \;\wedge\; \Phi)$
- Formally: for any state s of $TS(TA)$ it holds:

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

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

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

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

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 $\mathit{frac}(d) = d - \lfloor d \rfloor$
- $\eta \models x \leq c$ whenever $\lfloor \eta(x) \rfloor < c$ or $\lfloor \eta(x) \rfloor = c$ and $\mathit{frac}(\eta(x)) = 0$

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

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

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

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

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:

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

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)

Midterm Review

Verification -- Part I

- Transition systems: sequential circuits, concurrent systems, channel systems
- Linear-time properties: safety vs. liveness
- Regular properties: Büchi automata
- LTL: from LTL to Büchi automata, LTL model checking
- CTL*: LTL vs. CTL, fairness, model checking
- Symbolic verification: BDDs, bounded model checking
- Implementation relations: Bisimulation, simulation, stuttering

True or False?

$$\mathrm{AX}\,\mathrm{AG}\,p \equiv \mathrm{AG}\,\mathrm{AX}\,p$$

True or False?

$$\mathrm{EX}\,\mathrm{EG}\,p \equiv \mathrm{EG}\,\mathrm{EX}\,p$$

True or False?

AF AG *p* can be expressed in LTL.

True or False?

If Φ is a CTL formula and ψ is an LTL formula such that $\Phi \equiv \psi$, then $\neg\Phi \equiv \neg\psi$.

True or False?

$s \models \mathrm{EF}\,\mathrm{EG}\,p$ iff
there is a path π from s with $\pi \models \mathrm{F}\,\mathrm{G}\,p$

True or False?

$s \models \mathsf{EG}\,\mathsf{EF}\,p$ iff
there is a path π from s with $\pi \models \mathsf{G}\,\mathsf{F}\,p$

True or False?

Let *TS* be a transition system and Φ a CTL formula.
If *TS* does not satisfy $\neg\Phi$,
then *TS* satisfies Φ.

True or False?

Let s_1, s_2 be states of a transition system and let

$$\Phi = \mathsf{E}\,(a\,\mathsf{U}\,(\mathsf{EX}\,b \wedge \mathsf{EX}\,c)).$$

If $s_1 \vDash \Phi$ and $\underline{\text{not}}$ $s_2 \vDash \Phi$
then $\mathit{Traces}(s_1) \neq \mathit{Traces}(s_2)$.

True or False?

CTL* equivalence is strictly finer than CTL equivalence.

True or False?

LTL equivalence is strictly finer than CTL equivalence.

True or False?

CTL equivalence is strictly finer than LTL equivalence.

True or False?

If $s \vDash \mathrm{AF}\, p$
then $s \vDash_{fair} \mathrm{AF}\, p$

True or False?

If $s \vDash \mathrm{EF}\, p$
then $s \vDash_{fair} \mathrm{EF}\, p$

True or False?

$s \vDash_{fair} \mathsf{E}\,(a\,\mathsf{U}\,b)$ iff
$s \vDash \mathsf{E}\,(a\,\mathsf{U}\,(b \wedge \mathsf{EG}\,\mathit{true}))$

True or False?

$s \models_{fair} \mathrm{E}\,(a\,\mathrm{U}\,b)$ iff
$s \models \mathrm{E}\,(a\,\mathrm{U}\,(b \wedge a_{fair}))$

where a_{fair} is an atomic proposition with
$s \models a_{fair}$ iff $s \models_{fair} \mathrm{EG}\,\mathit{true}$

True or False?

For each Büchi automaton A there is an LTL formula φ such that $\mathrm{Words}(\varphi)$ is the language of A.

True or False?

If two states s_1 an s_2 in a finite transition system satisfy the same $CTL_{\setminus \mathrm{U}}$ formulas, then s_1 and s_2 are bisimilar.

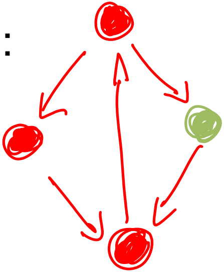
True or False?

Bisimilar transition systems are simulation equivalent.

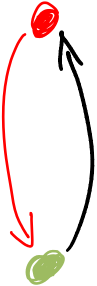
True or False?

The following two transition systems are stutter-trace equivalent.

S_1:



S_2:



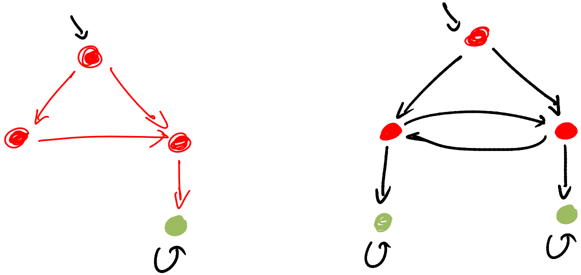
True or False?

Let TS_1 and TS_2 be two stutter-bisimilar transition systems and let φ be an LTL formula without Next

then either both TS_1 and TS_2 satisfy φ
or neither satisfies φ.

True or False?

The following two transition systems are divergence-sensitive stutter-bisimilar.



True or False?

For every boolean function there is a variable ordering such that the size of the ROBDD is polynomial.

True or False?

For every boolean function there is a variable ordering such that the size of the ROBDD is exponential.