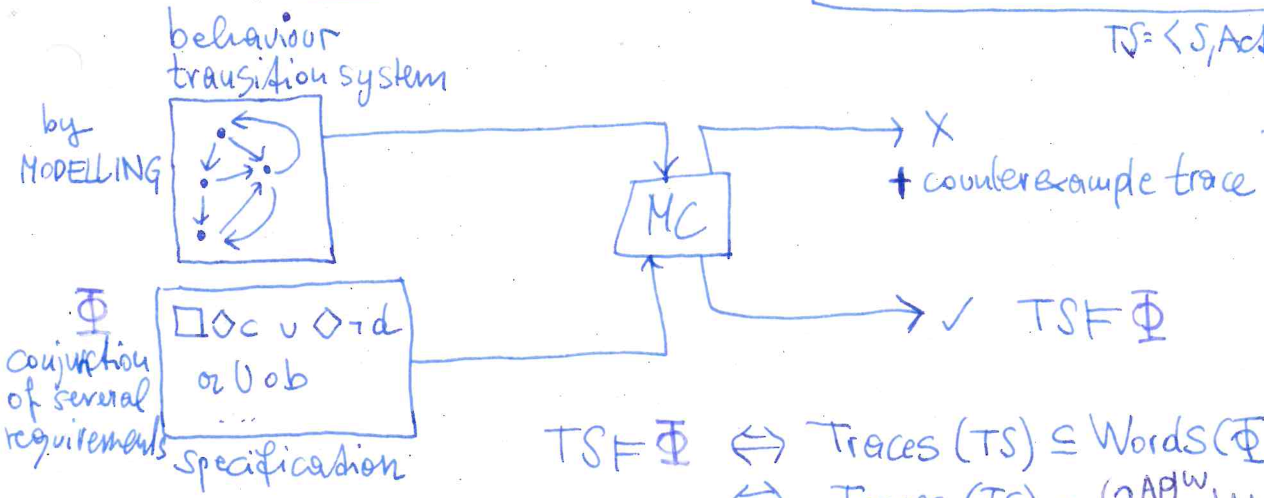
Lecture 6

10-02-2025
10:30 – 12.30

LTL-Model checking algorithm
CTL



$TS = \langle S, Act, \rightarrow, I, AP, L \rangle$

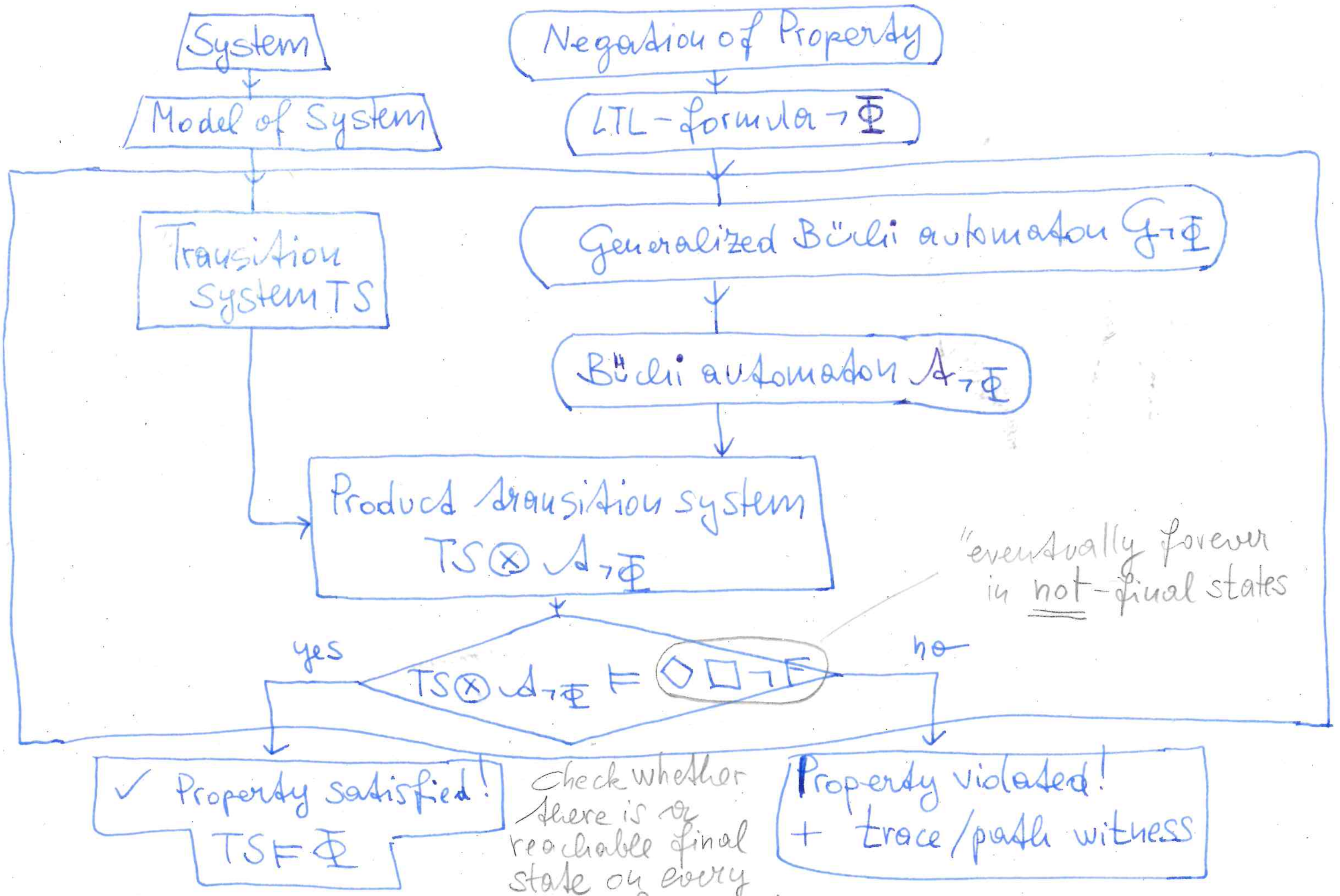
$TS \not\models \Phi$

$I \ni s \not\models \Phi$

$\pi \not\models \Phi$ path

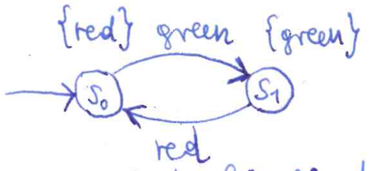
$$TS \models \Phi \iff Traces(TS) \subseteq Words(\Phi)$$
$$\iff Traces(TS) \cap ((2^{AP})^{\omega} \setminus Words(\Phi)) = \emptyset$$
$$\iff Traces(TS) \cap Words(\neg\Phi) = \emptyset$$
$$\iff Traces(TS) \cap \overset{\mathcal{L}_\omega(\mathcal{G}_{\neg\Phi})}{\mathcal{L}_\omega(\mathcal{A}_{\neg\Phi})} = \emptyset$$
$$\iff \mathcal{L}_\omega(TS \otimes \mathcal{A}_{\neg\Phi}) = \emptyset$$
$$\iff TS \otimes \mathcal{A}_{\neg\Phi} \models \Diamond\Box\neg F$$

Basic LTL-model checking algorithm (Vardi, Wolper 1986)



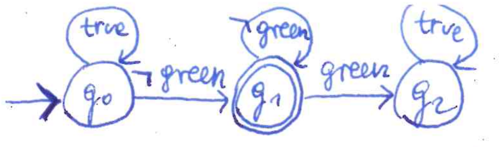
Check whether there is no reachable final state on every cycle of $TS \otimes \mathcal{A}_{\neg\Phi}$

Complexity: $O(|TS| \cdot 2^{|\Phi|})$ PSPACE-Complete



simple traffic light

PedTrLight

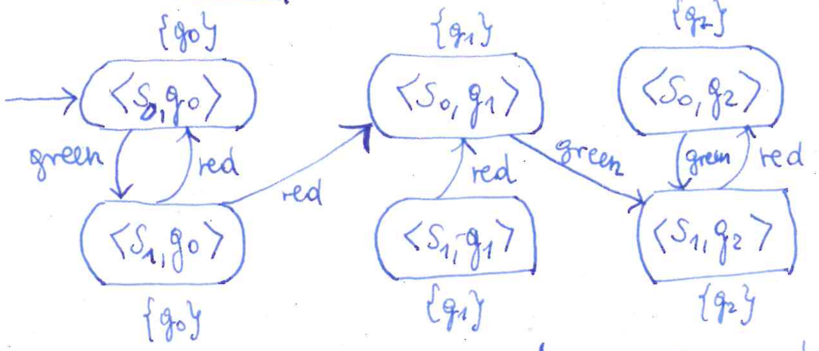


$\mathcal{A}_{\bar{P}_{pers}}$

p.202, p.223
regular properties

NBA for $\bar{P}_{pers} = (2^{AP})^{\omega} \setminus P_{pers}$ (only finitely often green)

where P_{pers}: "infinitely often green"


$\{q_0\}$
$\langle s_0, q_0 \rangle$
green
red
$\langle s_1, q_0 \rangle$
$\{q_0\}$
red
$\{q_1\}$
$\langle s_0, q_1 \rangle$
red
$\langle s_1, q_1 \rangle$
$\{q_1\}$
green
$\{q_2\}$
$\langle s_0, q_2 \rangle$
green
red
$\langle s_1, q_2 \rangle$
$\{q_2\}$

PedTrLight $\otimes$ $\mathcal{A}_{\bar{P}_{pers}}$

$TS = \langle S, Act, \rightarrow, I, AP, L \rangle$ transition system

$\mathcal{A} = \langle Q, 2^{AP}, \delta, Q_0, F \rangle$ NBA

Then $TS \otimes \mathcal{A} = \langle S \times Q, Act, \rightarrow', I', AP', L' \rangle$

where $\rightarrow'$ is defined via: $$\frac{s \xrightarrow{\alpha} t \qquad q \xrightarrow{L(t)} p}{\langle s, q \rangle \xrightarrow{\alpha} \langle t, p \rangle}$$

$$I' = \{ \langle s_0, q \rangle \mid s_0 \in I \text{ and } \exists q_0 \in Q_0 . (q_0 \xrightarrow{L(s_0)} q) \}$$

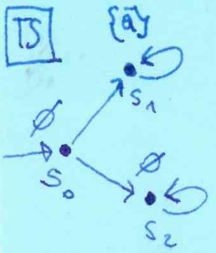
$$AP' = Q$$

$$L'(\langle s, q \rangle) = \{q\}$$

PedTrLight $\otimes \mathcal{A} \models \Diamond \Box \neg q_1$ "eventually forever" $\neg q_1$

$\Downarrow$

PedTrLight $\models$ "infinitely often green"



$TS \not\models \Diamond a$

$s_0 s_2^\omega \not\models \Diamond a$

$TS \not\models \underbrace{\neg \Diamond a}_{\equiv \; \Box \neg a}$

$s_0 s_1^\omega \models \Diamond a$

$s_0 s_1^\omega \not\models \neg \Diamond a$

$\exists \Diamond a$ CTL-formula

$TS \models \exists \Diamond a$

Computation Tree Logic CTL

Motivation LTL-formulas quantify universally over paths

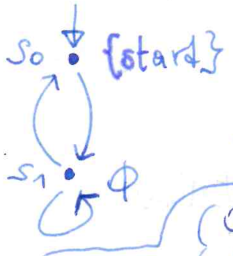
LTL: $s \models \varphi \iff \forall \pi \in Paths(s) : \pi \models \varphi$

thus LTL permits do quantify over all paths, but not directly over some.

OK: path existence can be modeled by checking $\neg\varphi$:

$$s \not\models \neg\varphi \iff \text{not } \forall \pi \in Paths(s) : \pi \models \neg\varphi$$
$$\iff \exists \pi \in Paths(s) : \text{not } \pi \not\models \varphi$$
$$\iff \exists \pi \in Paths(s) : \pi \models \varphi$$

Yet more complicated statements like "it is always possible to return to start"



cannot be specified in LTL

in particular: $s_0 \not\models \Box\Diamond start$ (LTL) since $s_0 s_1^\omega \not\models \Box\Diamond start$

$s_0 \models \forall\Box\exists\Diamond\, start$ (CTL)

Syntax CTL (Queille and Sifakis, 1982) (Clarke & Emerson 1986)

for some path / for all paths

(CTL-formulas) STATE formulas $\Phi ::= true \mid \overset{AP}{\overset{\in}{a}} \mid \neg\Phi \mid \Phi \wedge \Phi \mid \exists\varphi \mid \forall\varphi$

PATH formulas $\varphi ::= \bigcirc\Phi \mid \Phi \cup \Phi$

Defined operators (path formula)

eventually:

potentially

$\exists\Diamond\,\Phi := \exists(true \cup \Phi)$

inevitably

$\forall\Diamond\,\Phi := \forall(true \cup \Phi)$

always:

potentially invariantly

$\exists\Box\,\Phi := \neg\forall\Diamond\neg\Phi$ $= \neg\forall(true \cup \neg\Phi)$

invariantly

$\forall\Box\,\Phi := \neg\exists\Diamond\neg\Phi$ $= \neg\exists(true \cup \neg\Phi)$

LTL

$\Diamond\Phi := true \cup \Phi$

$\Box\Phi := \neg\Diamond\neg\Phi$

Examples of formulas over AP = $\{x=1, x<2, x\geq 3\}$

$\exists\bigcirc(x=1)$, $\forall\bigcirc(x=1)$, $x<2 \vee x=1$, $\exists((x<2)\cup(x\geq 3))$, $\forall(true \cup (x<2))$

Non-examples: $\underbrace{\exists(\underbrace{x=1 \wedge \underbrace{\forall\bigcirc(x\geq 3)}_{\text{state formula}})}_{\text{state, but not path formula}}}_{\text{incorrect as CTL-formula}}$, $\underbrace{\exists\bigcirc(\underbrace{true \cup (x=1)}_{\text{path, but not a state formula}})}_{\text{incorrect as CTL-formula}}$

Examples:

Safety $\forall\Box(\overbrace{\neg c_1 \vee \neg c_2}^{\neg(c_1 \wedge c_2)})$ $\forall\Box\left(\bigwedge_{1\leq i<j\leq n} \overbrace{\neg c_i \vee \neg c_j}^{\neg(c_i \wedge c_j)}\right)$ mutual exclusion

Liveness $\bigwedge_{1\leq i\leq n} \forall\Box\,\forall\Diamond c_i$

$\forall\Box(req \rightarrow \forall\Diamond res)$

Semantics

$$\boxed{TS \models \Phi \;:\Leftrightarrow\; \forall s_0 \in I : s_0 \models \Phi}$$

$Sat(\Phi) = \{s \in S \mid s \models \Phi\}$

For $TS = \langle S, A, \rightarrow, I, AP, L \rangle$, all $s \in S$, state formulas Φ, Ψ and paths π and path formulas φ:

state formulas	path formulas
$s \models true$	$\underbrace{\pi}_{\text{path}} \models \bigcirc \Phi \;:\Leftrightarrow\; \underbrace{\pi[1]}_{\text{state}} \models \Phi$
$s \models a \;:\Leftrightarrow\; a \in L(s)$	$\underbrace{\pi}_{\text{path}} \models \Phi \, U \, \Psi \;:\Leftrightarrow$
$s \models \neg \Phi \;:\Leftrightarrow\;$ not $s \models \Phi$	$\Leftrightarrow \exists j \geq 0 : \underbrace{\pi[j]}_{\text{state}} \models \Psi$ and
$s \models \Phi \wedge \Psi \;:\Leftrightarrow\; s \models \Phi$ and $s \models \Psi$	and $\forall 0 \leq i < j . \underbrace{\pi[i]}_{\text{state}} \models \Phi$
$s \models \exists \varphi \;:\Leftrightarrow\; \exists \pi \in Paths(s) : \pi \models \varphi$	
$s \models \forall \varphi \;:\Leftrightarrow\; \forall \pi \in Paths(s) : \pi \models \varphi$	

For the defined path formula $\lozenge \Phi$ it follows, for all paths π:

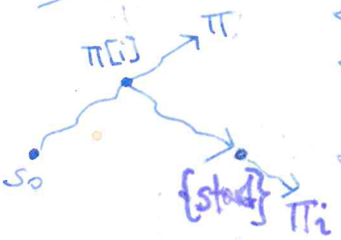
$\lozenge \Phi := T \, U \, \Phi$

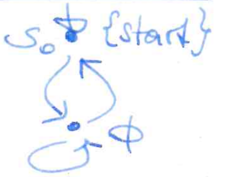
$\pi \models \lozenge \Phi \Leftrightarrow \exists j \geq 0 : \pi[j] \models \Phi$,

$\square \Phi$ is not a defined path formula, but $\forall \square \Phi$ and $\exists \square \Phi$ are defined State formulas

$s \models \overset{\exists}{\forall} \square \Phi \Leftrightarrow \overset{\exists}{\forall} \pi \in Paths(s) \, \pi[j] \models \Phi \; \forall j \geq 0$

Example: $s_0 \models \forall \square \exists \lozenge start$





$\Leftrightarrow \forall \pi \in Paths(s_0) : \pi \models \square \exists \lozenge start$

$\Leftrightarrow \forall \pi \in Paths(s_0) \; \forall i \geq 0 : \pi[i] \models \exists \lozenge start$

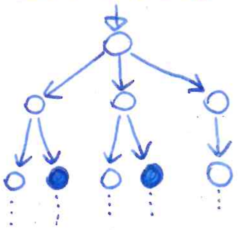
$\Leftrightarrow \forall \pi \in Paths(s_0) \; \forall i \geq 0 \; \exists \pi_i \in Paths(\pi[i]) : \pi_i \models \lozenge start$

$\Leftrightarrow \forall \pi \in Paths(s_0) \; \forall i \geq 0 \; \exists \pi_i \in Paths(\pi[i]) \; \exists j \geq 0 : \pi_i[j] \models start$

for every state $\pi[i]$ on a path from s_0 there is a path π_i that reaches a state in which start holds

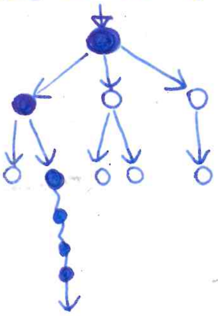
potentially

$\exists \lozenge$ black



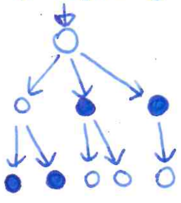
potentially invariantly

$\exists \square$ black



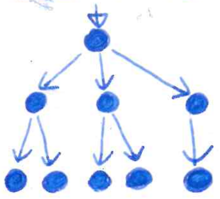
inevitably

$\forall \lozenge$ black

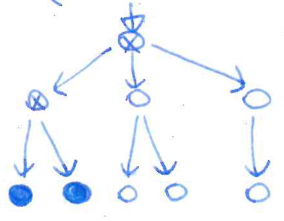


invariantly

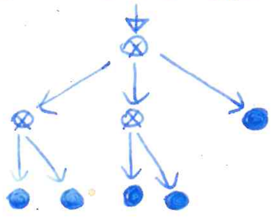
$\forall \square$ black



$\exists$(crossed U black)



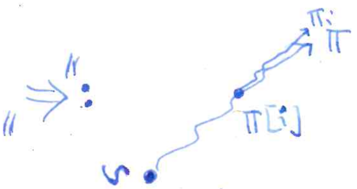
$\forall$ (crossed U black)



Example: Infinitely Often

$s \models \forall\Box\forall\Diamond a \iff \forall \pi \in Paths(s): \pi[i] \models a$ for infinitely many i.

"$\Rightarrow$":

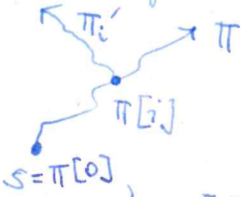


we consider an arbitrary path π from s

Let $i \geq 0$. We have to show (it suffices!) that $j \geq i$ exists with $L(\pi[j]) \ni a$.

Since $s \models \forall\Box\forall\Diamond a$, we have $\pi \models \forall\Box\forall\Diamond a$, which implies $\pi[i] \models \forall\Diamond a$. Then for $\pi_i := \pi[i]\pi[i+1]\ldots$ it holds: $\pi_i \models \Diamond a$, which implies $\underbrace{\pi[j] \models a}_{a \in L(\pi[j])}$ for some $j \geq i$.

"$\Leftarrow$": To show $s \models \forall\Box\forall\Diamond a$, we have to show: for all paths π from s, and for all paths π_i' from $\pi[i]$, for $i \geq 0$, there is some $j \geq 0$ such that $a \in L(\pi_i'[j])$



But then $\pi' := \pi[0]\ldots\pi[i] \cdot \pi_i'$ also is a path from s, on which by assumption a is true infinitely often. Consequently a holds at least once on π_i' (and in fact infinitely often).

s "decides" formula φ : $\iff$ $\iff s \models \varphi$ or $s \models \neg\varphi$

(LTL : $s \models \Box\Diamond a$) $\iff \forall \pi \in Paths(s).\ \pi[i] \models a$ for infinitely many i)

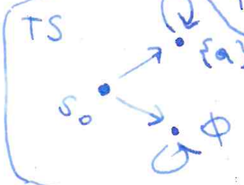
(LTL: traces decide formulas, but states and transition systems do not

$\sigma \models \varphi \iff \sigma \not\models \neg\varphi$

$\sigma \not\models \varphi \iff \sigma \models \neg\varphi$

$TS \models \varphi \;\overset{\not\Leftarrow}{\Rightarrow}\; TS \not\models \neg\varphi$

$TS \not\models \varphi \;\underset{\not\Rightarrow}{\Leftarrow}\; TS \models \neg\varphi$



$s_0 \not\models \Diamond a$ $\quad$ $TS \not\models \Diamond a$

$s_0 \not\models \neg\Diamond a$ $\quad$ $TS \not\models \neg\Diamond a$

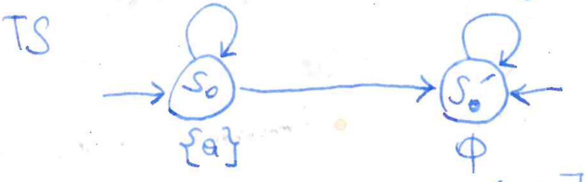
hence again

$TS \models \varphi \;\overset{\not\Leftarrow}{\Rightarrow}\; TS \not\models \neg\varphi$

$TS \not\models \varphi \;\underset{\not\Rightarrow}{\Leftarrow}\; TS \models \neg\varphi$

In CTL: States decide CTL(-state)-formulas, paths decide CTL-path-formul

but: transition systems do not if they have ≥ 2 initial states (they do decide formulas if there is just 1 initial state)



note: 2 initial states

$s_0 \models \exists\Box a$ $\quad$ $s_0' \not\models \exists\Box a$ $\quad$ Hence: $TS \not\models \exists\Box a$

$s_0 \not\models \neg\exists\Box a$ $\quad$ $s_0' \models \neg\exists\Box a$ $\quad$ $TS \not\models \neg\exists\Box a$

$TS \models \exists\varphi \iff \forall s \in I.\ \exists \pi \in Path(s): \pi \models \varphi$ $\quad \overset{\Rightarrow}{\not\Leftarrow}$

in general: $TS \not\models \neg\exists\varphi \iff \exists \pi \in Paths(TS): \pi \models \varphi$

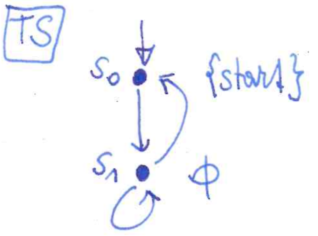
$TS \not\models \neg\exists\varphi \iff$ not $TS \models \neg\exists\varphi$

$\iff$ not $\forall s \in I: s \models \neg\exists\varphi$

$\iff$ not $\forall s \in I: s \not\models \exists\varphi$

$\iff$ not $\forall s \in I:$ not $\exists \pi \in Path(s): \pi \models \varphi$

$\iff \exists s \in I\ \exists \pi \in Paths(s): \pi \models \varphi$ $\quad (\iff \exists \pi \in Path(TS): \pi \models \varphi)$



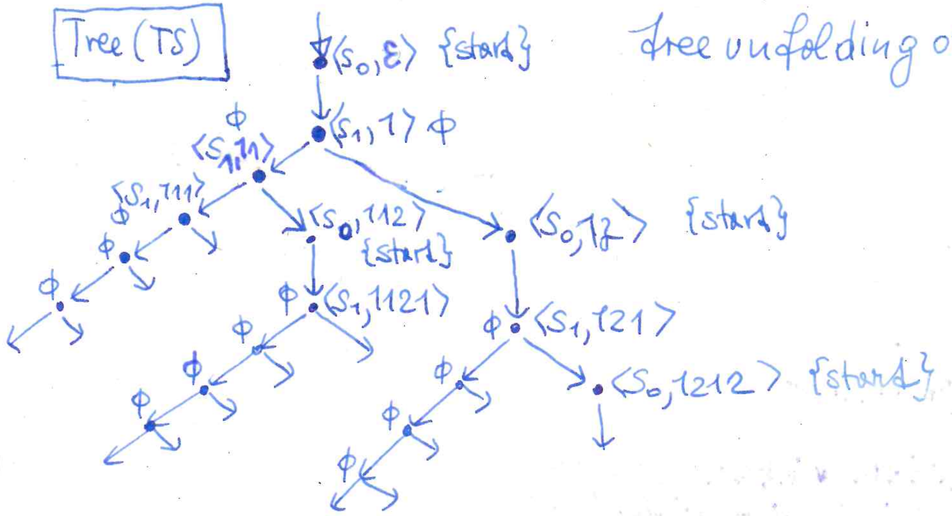
$s_0 \models \forall\Box\exists\Diamond start$

$TS \models \forall\Box\exists\Diamond start$

TL $\{ s_0 \not\models \Box\Diamond start$

or $TS \not\models \Box\Diamond start$

tree unfolding of TS



$\langle s_0, \varepsilon\rangle \models \forall\Box\exists\Diamond start$

$Tree(TS) \models \forall\Box\exists\Diamond start$

$\langle s_0, \varepsilon\rangle \not\models \Box\Diamond start$

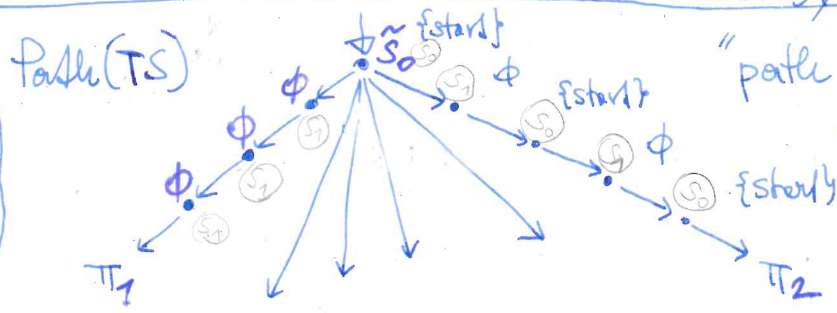
since

$\langle s_0,\varepsilon\rangle\langle s_1,1\rangle\langle s_1,11\rangle \ldots \not\models \Box\Diamond sta$

$Tree(TS) \not\models \Box\Diamond start$

"forcing CTL to look at TS like LTL does"

"path unfolding" of TS



$\tilde{s}_0 \not\models \forall\Box\exists\Diamond start$

since $\pi_1 \not\models \Box\exists\Diamond start$

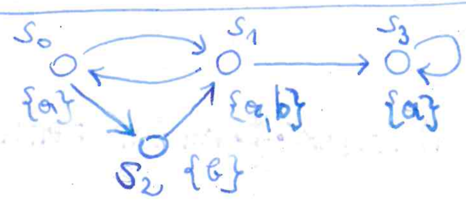
$Path(TS) \not\models \forall\Box\exists\Diamond start$

$s_0 \not\models \Box\Diamond start$

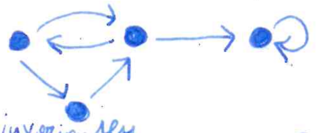
since $\pi_1 \not\models \Box\Diamond start$

$Path(TS) \not\models \Box\Diamond start$

Note: $\pi_2 \models \Box\exists\Diamond start$ and: $\pi_2 \models \Box\Diamond start$

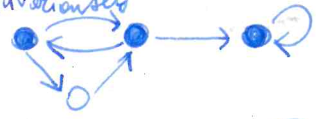


$\exists\Diamond a$



potentially invariantly

$\exists\Box a$

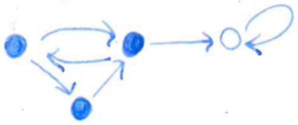


potentially

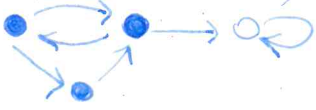
$\exists\Diamond b$

$\forall\Diamond b$

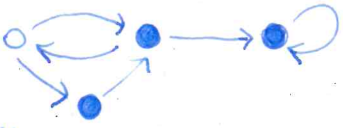
inevitably



$\exists(a\, U (\neg a \wedge \forall(\neg a\, U\, b)))$

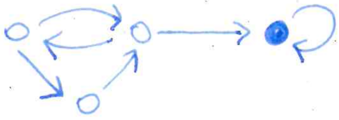


$\forall\Diamond a$

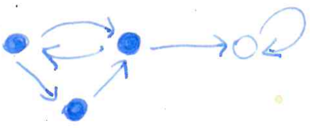


invariantly

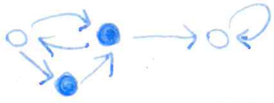
$\forall\Box a$



$\forall(a\, U\, b)$



$\forall(\neg a\, U\, b)$



$\neg a \wedge \forall(a\, U\, b)$