Computation Tree Logic CTL

Motivation LTL-formulas quantify universally over paths

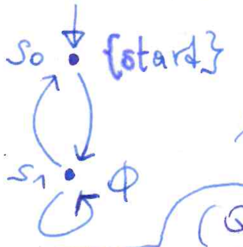
LTL: $s \models \varphi \iff \forall \pi \in Paths(s) : \pi \models \varphi$

thus LTL permits to 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)

(CTL-formulas) STATE formulas $\Phi ::= true \mid \overset{AP}{a} \mid \neg\Phi \mid \Phi\wedge\Phi \mid \overset{\text{for some path}}{\exists\varphi} \mid \overset{\text{for all paths}}{\forall\varphi}$

PATH formular $\varphi ::= \bigcirc\Phi \mid \Phi\,\mathsf{U}\,\Phi$

Defined operators (path formular)

eventually: $\overset{\text{potentially}}{\exists\Diamond\Phi} := \exists(true\,\mathsf{U}\,\Phi)$

$\overset{\text{inevitably}}{\forall\Diamond\Phi} := \forall(true\,\mathsf{U}\,\Phi)$

always: $\overset{\text{potentially invariantly}}{\exists\Box\Phi} := \neg\forall\Diamond\neg\Phi \equiv \neg\forall(true\,\mathsf{U}\,\neg\Phi)$

$\overset{\text{invariantly}}{\forall\Box\Phi} := \neg\exists\Diamond\neg\Phi \equiv \neg\exists(true\,\mathsf{U}\,\neg\Phi)$

LTL

$\Diamond\Phi := true\,\mathsf{U}\,\Phi$

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

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

$\exists\bigcirc(x=1)$, $\forall\bigcirc(x=1)$, $x<2 \vee x=1$, $\exists((x<2)\,\mathsf{U}\,(x\geq 3))$, $\forall(true\,\mathsf{U}\,(x<2))$

Non-examples: $\underbrace{\exists\underbrace{(x=1\wedge\underbrace{\forall\bigcirc(x\geq3)}_{\text{state formula}})}_{\text{state, but not path formula}}}_{\text{incorrect as CTL-formula}}$, $\underbrace{\exists\bigcirc\underbrace{(true\,\mathsf{U}\,(x=1))}_{\text{path, but not a state formula}}}_{\text{incorrect as CTL-formula}}$

Examples:

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

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

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

Semantics

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

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

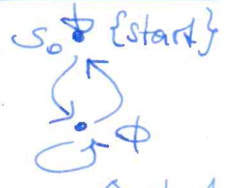
For $TS = \langle S, A, \rightarrow, I, AP, L\rangle$, all $s \in S$, state formulas Φ, Ψ and paths π and path formulas φ:

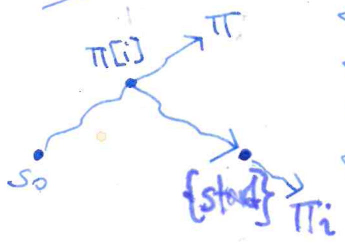
state formulas	path formulas
$s \models true$ $s \models a \;:\Leftrightarrow\; a \in L(s)$ $s \models \neg\Phi \;:\Leftrightarrow\;$ not $s \models \Phi$ $s \models \Phi \wedge \Psi \;:\Leftrightarrow\; s \models \Phi$ and $s \models \Psi$ $s \models \exists\varphi \;:\Leftrightarrow\; \exists \pi \in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

For the defined path formulas $\Diamond\Phi$ and $\Box\Phi$ it follows, for all paths π:

$$\pi \models \Diamond\Phi \Leftrightarrow \exists j \geq 0: \pi[j] \models \Phi,$$
$$\pi \models \Box\Phi \Leftrightarrow \forall j \geq 0: \pi[j] \models \Phi.$$

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





$\Leftrightarrow \forall \pi \in Paths(s_0): \pi \models \Box\exists\Diamond start$

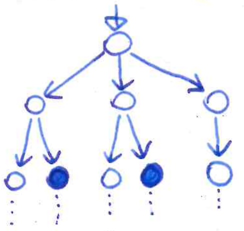
$\Leftrightarrow \forall \pi \in Paths(s_0)\ \forall i \geq 0:\ \pi[i] \models \exists\Diamond start$

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

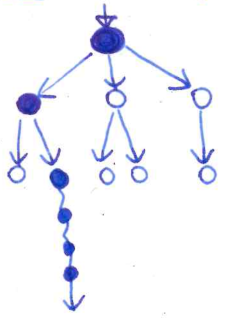
$\Leftrightarrow \forall \pi \in Paths(s_0)\ \forall i \geq 0\ \exists \pi_i \in Paths(\pi[i])\ \exists j \geq 0: \pi_i[j] \models start$

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

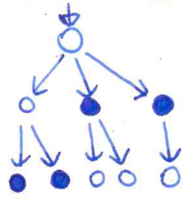
potentially $\exists\Diamond$ black



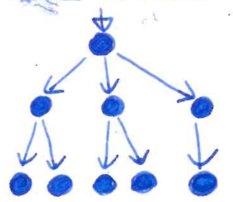
potentially invariantly $\exists\Box$ black



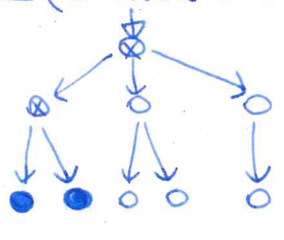
inevitably $\forall\Diamond$ black



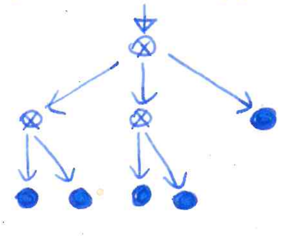
invariantly $\forall\Box$ black



$\exists$(crossed U black)



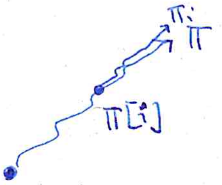
$\forall$(crossed U black)



Example. Infinitely Often

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

„$\Rightarrow$":



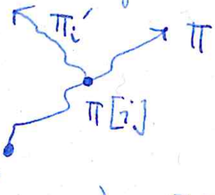
we consider an arbitrary path π from s

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 $\underline{\pi[j] \models a}$ for some $j \geq i$.

$a \in L(\pi[j])$

„$\Leftarrow$": 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, there is some $j \geq 0$ such that $a \in L(\pi_i'$



But then $\pi' := \pi[0]\ldots\pi[i]\cdot\pi_i'$ 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 $\underline{\text{decide}}$ formulas, but $\overbrace{\text{transition systems}}^{\text{states, and}}$ $\underline{\text{do not}}$

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

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

$TS \models \varphi \underset{\not\Longleftarrow}{\Longrightarrow} TS \not\models \neg\varphi$

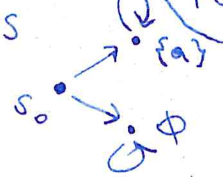
$TS \not\models \varphi \underset{\not\Longrightarrow}{\Longleftarrow} TS \models \neg\varphi$

hence again

$TS \models \varphi \underset{}{\overset{\not\Longleftarrow}{\Longrightarrow}} TS \not\models \neg\varphi$

$TS \not\models \varphi \underset{\not\Longrightarrow}{\Longleftarrow} TS \models \neg\varphi$

TS



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

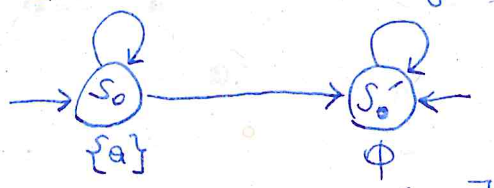
$s_0 \not\models \neg\Diamond a$ $TS \not\models \neg\Diamond a$

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

but: transition systems do not if they have ≥ 2 initial states

(they do decide formulas if there is just 1 initial stat

TS



note: 2 initial states

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

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

$TS \models \exists\varphi \iff \forall s\in I.\ \exists\pi\in Path(s): \pi \models \varphi$

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 $TS \models \neg\exists\varphi$

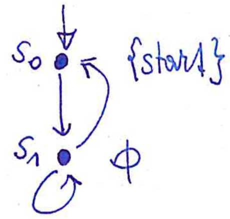
$\iff$ not $\forall s\in I: s \models \neg\exists\varphi$

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

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

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

TS



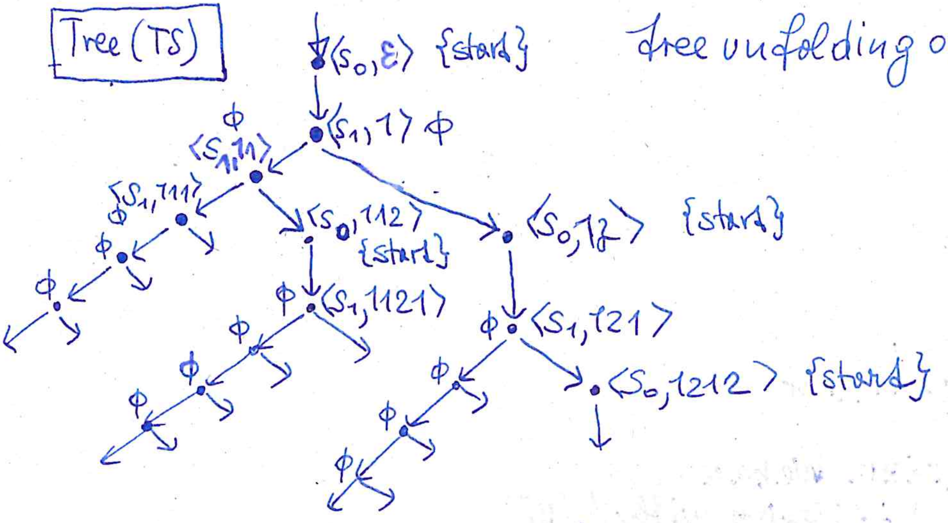
$s_0 \models \forall\Box\exists\Diamond start$

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

LTL $\left\{ \begin{array}{l} s_0 \not\models \Box\Diamond start \\ \text{or } TS \not\models \Box\Diamond start \end{array}\right.$

Tree (TS)

tree unfolding of TS


$\langle s_0,\varepsilon\rangle$ {start}
$\langle s_1,1\rangle$ Φ
$\langle s_1,11\rangle$ Φ
$\langle s_1,111\rangle$ Φ
$\langle s_0,112\rangle$ {start}
$\langle s_0,12\rangle$ {start}
$\langle s_1,1121\rangle$ Φ
$\langle s_1,121\rangle$ Φ
$\langle s_0,1212\rangle$ {start}
Φ

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

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

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

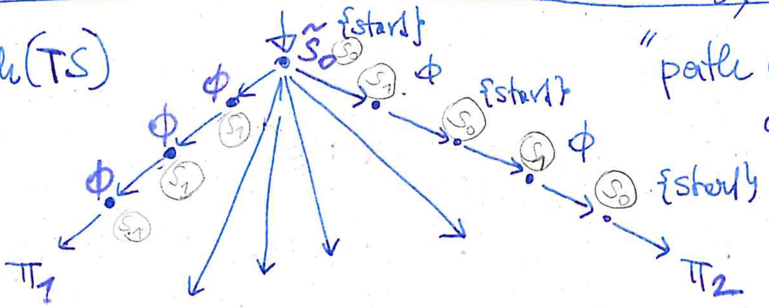
since

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

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

"forcing CTL to look at TS like LTL does"

Path(TS)



"path unfolding" of TS

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

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

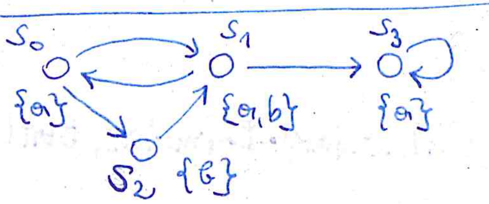
$Path(TS) \not\models \forall\Box\exists\Diamond start$

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

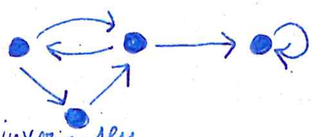
since $\pi_1 \not\models \Box\Diamond start$

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

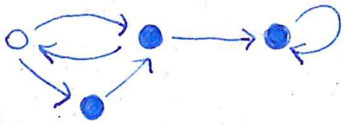
Note: $\pi_2 \models \Box\exists\Diamond start$ and: $\pi_2 \models \Box\Diamond start$



$\exists\bigcirc a$

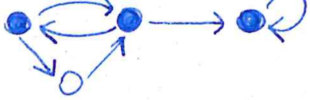


$\forall\bigcirc a$



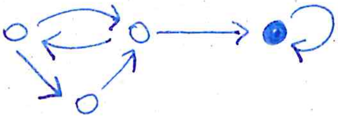
potentially invariantly

$\exists\Box a$



invariantly

$\forall\Box a$

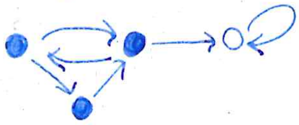


potentially

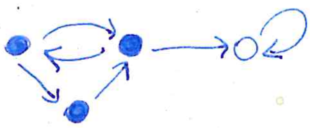
$\exists\Diamond b$

$\forall\Diamond b$

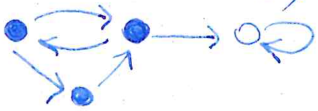
inevitably



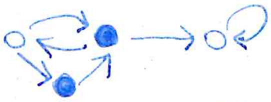
$\forall(a\,U\,b)$



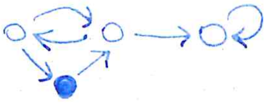
$\exists(a\,U(\neg a \wedge \forall(\neg a\,U\,b)))$



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



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



Aspect	Linear time	Branching time
"behaviour" in a state s	path-based: trace(s)	state-based computation tree of s
temporal logic	LTL: path formula φ $s \models \varphi \Leftrightarrow$ $\Leftrightarrow \forall \pi \in Paths(s): \pi \models \varphi$	CTL: state formulae existential path quantification universal path quantification
complexity of model checking problems	PSPACE-complete $O(\lvert TS\rvert \cdot \exp(\lvert\varphi\rvert))$	PTIME $O(\lvert TS\rvert \cdot \lvert\Phi\rvert)$
adequate subsumption and equivalence relations	trace inclusion and trace equivalence (can be checked in PSPACE-complete)	bisimulation subsumption bisimulation equivalence (can be checked in polynomial time)
fairness	no special techniques needed	special techniques needed

CTL-formulas Φ and Ψ over AP are equivalent (denoted $\Phi \equiv \Psi$) if $Sat(\Phi) = Sat(\Psi)$ for all transition systems TS over

Normal Forms

Existential Normal Form (ENF)

$$\Phi ::= true \mid \overset{\overset{AP}{\cup}}{a} \mid \Phi \wedge \Phi \mid \neg \Phi \mid \exists \bigcirc \Phi \mid \exists(\Phi \, U \, \Phi) \mid \exists \Box \Phi$$

Thm. For every CTL-formula there is an equivalent CTL-formula in ENF.

Positive Normal Form

$$\Phi ::= true \mid false \mid a \mid \neg a \mid \Phi_1 \wedge \Phi_2 \mid \Phi_1 \vee \Phi_2 \mid \exists \varphi \mid \forall \varphi$$

$$\varphi ::= \bigcirc \Phi \mid \Phi_1 \, U \, \Phi_2 \mid \Phi_1 \underset{\text{weak until}}{W} \Phi_2$$

Thm. For each CTL-formula there is an equivalent CTL-formula in PN

Weak Until:

$$\pi \models \Phi W \Psi \quad \Leftrightarrow \quad \pi \models \Phi U \Psi \text{ or } \pi \models \Box(\Phi \wedge \neg\Psi)$$
$$\Leftrightarrow \quad \pi \models \Phi U \Psi \text{ or } \pi \models \Box\Phi$$

Can be obtained by defining:

$$\exists(\Phi W \Psi) := \neg \forall((\Phi \wedge \neg\Psi) U (\neg\Phi \wedge \neg\Psi))$$
$$\forall(\Phi W \Psi) := \neg \exists((\Phi \wedge \neg\Psi) U (\neg\Phi \wedge \neg\Psi))$$

6

$\boxed{\text{CTL}^+}$ Extending CTL path-formulas with Boolean Connectives

Syntax $\Phi ::= true \mid \overset{\in AP}{a} \mid \Phi_1 \wedge \Phi_2 \mid \neg\Phi \mid \exists\varphi \mid \forall\varphi$ same as for CTL

$\varphi ::= \varphi_1 \wedge \varphi_2 \mid \neg\varphi \mid \bigcirc\Phi \mid \Phi_1 \mathsf{U} \Phi_2$ (CTL$^+$ path formulas

Examples $\underbrace{\exists(a \mathsf{W} b)}_{\substack{\text{CTL-formula}\\ \text{(after using the}\\ \text{definition of W}}} \equiv \underbrace{\exists((a\mathsf{U}b) \vee \Box a)}_{\substack{\text{CTL}^+\text{-formula}\\ \text{but not a CTL-formula}}}$

$\underbrace{\exists(\Diamond a \wedge \Diamond b)}_{\text{CTL}^+\text{-formula}} \equiv \underbrace{\exists\Diamond(a \wedge \exists\Diamond b) \vee \exists\Diamond(b \wedge \exists\Diamond a))}_{\text{CTL-formula}}$

Thm. Every CTL$^+$-formula is equivalent to a CTL-formula.

Incomparable expressiveness of LTL and CTL

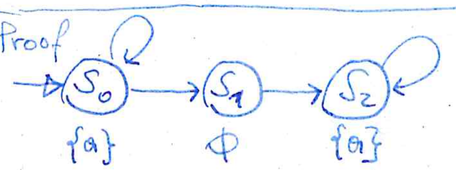
Lemma. Φ a CTL-formula, φ a LTL-formula that results by eliminating all path quantifiers from Φ.
Then: $\underbrace{\Phi \equiv \varphi}_{\text{Forall TS: } TS \models \Phi \Leftrightarrow TS \models \varphi}$ or there is no LTL-formula that is equivalen
to Φ.

Examples:

$\forall\bigcirc a \equiv \bigcirc a$, $\forall(a\mathsf{U}b) \equiv a\mathsf{U}b$, $\forall\Diamond a \equiv \Diamond a$, $\forall\Box a \equiv \Box a$
$\forall\Box\forall\Diamond a \equiv \Box\Diamond a$

Proposition. $\forall\Diamond\forall\Box a \not\equiv \Diamond\Box a$.

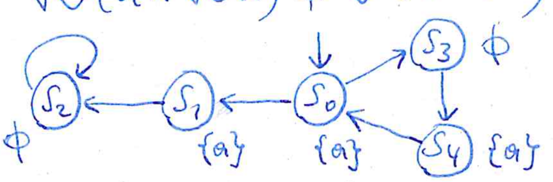
Proof



$s_0 \models_{CTL} \Diamond\Box a$ because $\pi = s_0 \ldots$ $\left\langle \begin{array}{l} \pi = s_0^\omega \models \Diamond\Box a \\ \pi = s_0^* s_1 s_2^\omega \models \Diamond\Box a \end{array}\right.$

$s_0 \not\models \forall\Diamond\forall\Box a$ because
$s_0^\omega \not\models \Diamond\forall\Box a$

$\forall\Diamond(a \wedge \forall\bigcirc a) \not\equiv \Diamond(a \wedge \bigcirc a)$



$s_0 s_1 s_2^\omega \models_{CTL} \Diamond(a \wedge \bigcirc a)$ since $s_0 s_1 s_2^\omega \models a \wedge \bigcirc a$
$s_0 s_1 s_2^\omega \not\models_{CTL} \Diamond(a \wedge \forall\bigcirc a)$ since $s_0 \not\models_{CTL} \forall\bigcirc a$, $s_1 \not\models_{CTL} \forall\bigcirc a$, $s_2 \not\models \forall\bigcirc a$
Hence $s_0 \not\models_{CTL} \forall\Diamond(a \wedge \forall\bigcirc a)$, yet $s_0 \models_{CTL} \Diamond(a \wedge \bigcirc a)$
since $(s_0 s_3 s_4)^* s_1 s_2^\omega \models \Diamond(a \wedge \bigcirc a)$
$(s_0 s_3 s_4)^\omega \models \Diamond(a \wedge \bigcirc a)$.

Thm. Incomparable Expressiveness LTL/CTL
(a) There are LTL-formulas for which no equivalent CTL-formulas exist
e.g. $\Diamond\Box a$ and $\Diamond(a \wedge \bigcirc a)$.

(b) There are CTL-formulas for which no equivalent LTL-formulas ex
E.g. $\forall\Diamond\forall\Box a$ and $\forall\Diamond(a \wedge \forall\bigcirc a)$ and $\forall\Box\exists\Diamond a$

CTL* (Emerson, Halpern, 1985/86)

Syntax $\Phi ::= \mathit{true} \mid \overset{\overset{AP}{\cup}}{a} \mid \Phi \wedge \Phi \mid \neg \Phi \mid \exists \varphi$ (CTL*-formulas, state formulas

$\varphi ::\ \Phi \mid \varphi \wedge \varphi \mid \neg \varphi \mid \bigcirc \varphi \mid \varphi \,\mathsf{U}\, \varphi$ path formulas

defined: $\Diamond \varphi := \mathit{true}\, \mathsf{U}\, \varphi$ $\quad \forall \varphi :\equiv \neg \exists \neg \varphi$

$\Box \varphi := \neg \Diamond \neg \varphi$

Example $\forall \Box (\bigcirc \Diamond a \wedge \neg (b \,\mathsf{U}\, \Box c))$, not CTL-formul

$\forall \bigcirc \Box \neg a \wedge \exists \Diamond \Box (a \vee \forall (b \,\mathsf{U}\, a))$

Semantics

For $a \in AP$ and $TS = \langle S, Act, \rightarrow, I, AP, L \rangle$ a transition system, and all $s \in S$:

$s \models a \quad :\Leftrightarrow \quad a \in L(s)$

$s \models \neg \Phi \quad :\Leftrightarrow \quad$ not $s \models \Phi$ (i.e. $s \not\models \Phi$)

$s \models \Phi \wedge \Psi \quad :\Leftrightarrow \quad (s \models \Phi)$ and $(s \models \Psi)$

$s \models \exists \varphi \quad :\Leftrightarrow \quad \pi \models \varphi$ for some $\pi \in Paths(s)$.

same as for CTL

For all paths π in S:

$\pi \models \Phi \quad :\Leftrightarrow \quad \pi[0] \models \Phi$ } NEW for CTL*

$\pi \models \varphi_1 \wedge \varphi_2 \quad :\Leftrightarrow \quad \pi \models \varphi_1$ and $\pi \models \varphi_2$

$\pi \models \neg \varphi \quad :\Leftrightarrow \quad \pi \not\models \varphi$

same as for CTL+

$\pi \models \bigcirc \varphi \quad :\Leftrightarrow \quad \pi_{\geq 1} \models \varphi$

$\pi \models \varphi_1 \,\mathsf{U}\, \varphi_2 \quad :\Leftrightarrow \quad \exists j \geq 0. (\pi_{\geq j} \models \varphi_2 \wedge \forall 0 \leq k < j : \pi_{\geq k} \models \varphi_1)$

same as for CTL

$Sat(\Phi) := \{ s \in S \,/\, s \models \Phi \}$

$TS \models \Phi \quad :\Leftrightarrow \quad \forall s_0 \in I : s_0 \models \Phi.$

Embedding of LTL in CTL*

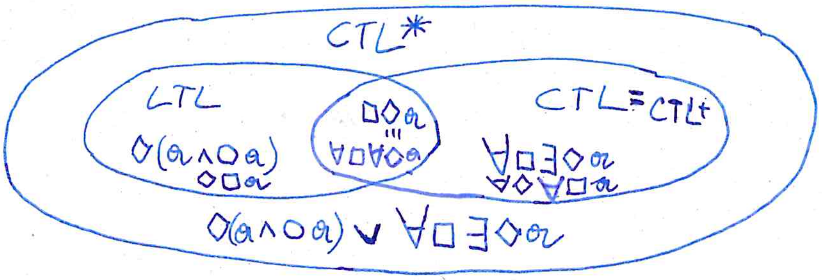
Thm. $TS = \langle S, Act, \rightarrow, I, AP, L \rangle$ a transition system without terminal state

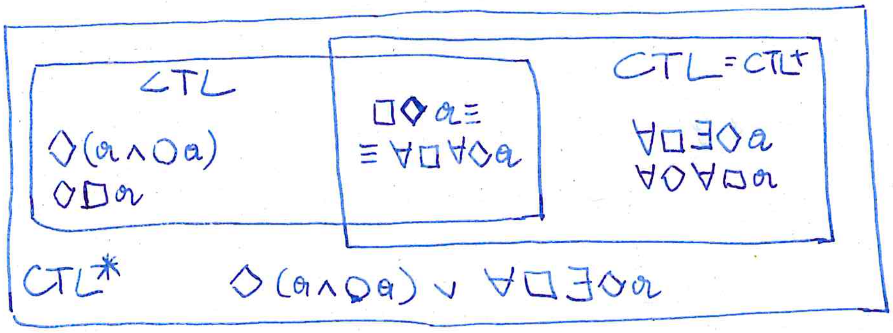
For every LTL-formula φ and for each $s \in S$:

$$\underbrace{s \models \varphi}_{\text{LTL-semantics}} \quad \Leftrightarrow \quad \underbrace{s \models \forall \varphi}_{\text{CTL*-semantics}}$$

$$TS \models_{LTL} \varphi \quad \Leftrightarrow \quad TS \models_{CTL^*} \forall \varphi$$

Relationship between LTL, CTL, and CTL*


CTL*
LTL
$\Diamond(a\wedge\bigcirc a)$
$\Diamond\Box a$
$\Box\Diamond a$
$\equiv$
$\forall\Box\forall\Diamond a$
CTL=CTL+
$\forall\Box\exists\Diamond a$
$\forall\Diamond\forall\Box a$
$\Diamond(a\wedge\bigcirc a)\vee\forall\Box\exists\Diamond a$


LTL
$\Diamond(a\wedge\bigcirc a)$
$\Diamond\Box a$
$\Box\Diamond a\equiv$
$\equiv\forall\Box\forall\Diamond a$
CTL=CTL+
$\forall\Box\exists\Diamond a$
$\forall\Diamond\forall\Box a$
CTL*
$\Diamond(a\wedge\bigcirc a)\vee\forall\Box\exists\Diamond a$

Thm. For the CTL*-formula $\Diamond(a\wedge\bigcirc a)\vee\forall\Box\exists\Diamond a$ there does not exist any equivalent LTL or CTL-formula.

	CTL	LTL	CTL*
model checking	PTIME	PSPACE-complete	PSPACE-Complete
without fairness	$size(TS)\cdot\lvert\Phi\rvert$	$size(TS)\cdot exp(\lvert\Phi\rvert)$	$size(TS)\cdot exp(\lvert\Phi\rvert)$
with fairness	$size(TS)\cdot\lvert\Phi\rvert\cdot\lvert fair\rvert$	$size(TS)\cdot exp(\lvert\Phi\rvert)\cdot\lvert fair\rvert$	$size(TS)\cdot exp(\lvert\Phi\rvert)\cdot\lvert fair\rvert$
for fixed specifications	$O(size(TS))$	$O(size(TS))$	$O(size(TS))$
Satisfiability check	EXPTIME	PSPACE-complete	2EXPTIME
best known technique upper bound	$O(exp(\lvert\Phi\rvert))$	$exp(\lvert\Phi\rvert)$	$exp(exp(\Phi))$

Exercises from Last time

Ex 5.24 (d) Is $\Phi := \Box a \,\mathsf{U}\, \Diamond b \rightarrow \Box(a \,\mathsf{U}\, \Diamond b)$ valid/satisfiable?

Φ is satisfiable: $\underbrace{\emptyset\emptyset\emptyset\ldots}_{\emptyset^\omega \in (2^{AP})^\omega} \models \Phi$ because $\emptyset^\omega \not\models \Diamond b$

$\emptyset^\omega \not\models \Box a \,\mathsf{U}\, \Diamond b$

hence $\emptyset^\omega \models \Box a \,\mathsf{U}\, \Diamond b \rightarrow \ldots$

Φ is not valid: $\sigma := \underbrace{\{b\}\emptyset\emptyset\ldots}_{\{b\}\emptyset^\omega} \not\models \Phi$ because $\sigma = \{b\}\emptyset^\omega \models \Diamond b$

$\sigma = \{b\}\emptyset^\omega \models \Box a \,\mathsf{U}\, \Diamond b$

$\sigma \not\models \Box(a \,\mathsf{U}\, \Diamond b)$

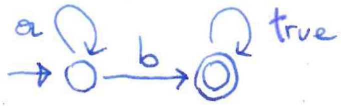
since $\sigma_{\geq 1} \not\models \Diamond b$

Marokh's argumentation: $\Psi \,\mathsf{U}\, \Diamond \Phi \equiv \Diamond \Phi$ for all Φ, Ψ!

Hence $(\Box a \,\mathsf{U}\, \Diamond b \rightarrow \Box(a \,\mathsf{U}\, \Diamond b)) \equiv \Diamond b \rightarrow \Box \Diamond b$ which is obviously not valid!

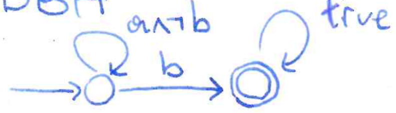
nondeterministic Büchi automaton

NBA for $a \,\mathsf{U}\, b$ with 2 states

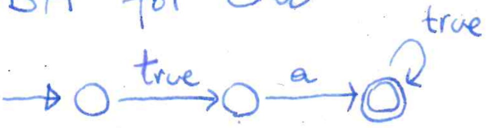


deterministic Büchi-automaton with

DBA 2 states

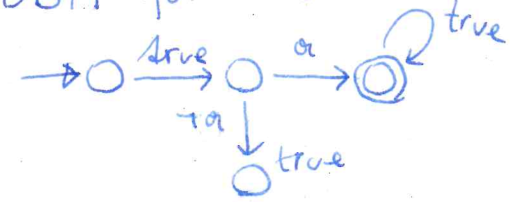


NBA for $\bigcirc a$

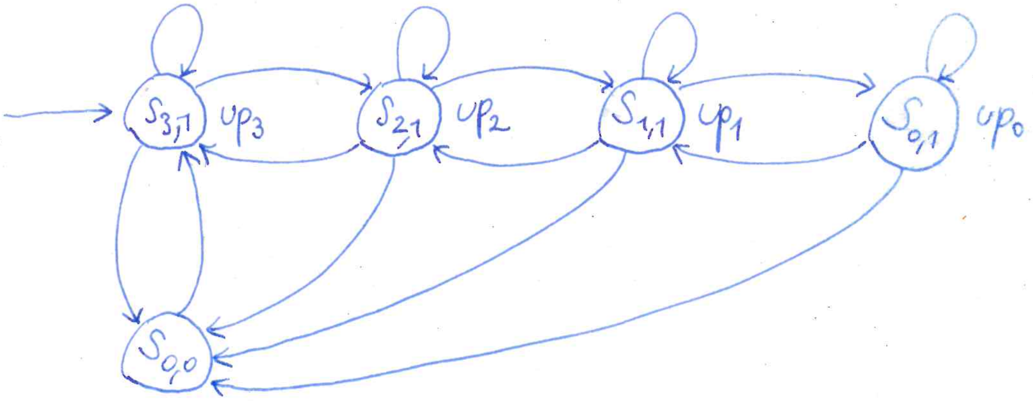


3 states

DBA for $\bigcirc a$



4 states



Possibly the system never goes down $\exists \Box \neg down$

Invariantly the system never goes down $\forall \Box \neg down$

It is always possible to start as new $\forall \Box \exists \Diamond up_3$

The system always eventually goes down and is operational until going down $\forall ((up_3 \vee up_2) \,\mathsf{U}\, down)$