CTL, LTL and CTL*

Lecture #18 of Model Checking

Joost-Pieter Katoen

Lehrstuhl 2: Software Modeling & Verification

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

June 12, 2007

Overview Lecture #18

$\Rightarrow$ Repetition: CTL syntax and semantics

- CTL equivalence
- Expressiveness of LTL versus CTL
- CTL*: extended CTL

Computation tree logic

modal logic over infinite trees [Clarke & Emerson 1981]

- Statements over states
  - $a \in AP$ — atomic proposition
  - $\neg\, \Phi$ and $\Phi \wedge \Psi$ — negation and conjunction
  - $\exists \varphi$ — there *exists* a path fulfilling φ
  - $\forall \varphi$ — *all* paths fulfill φ
- Statements over paths
  - $\bigcirc\, \Phi$ — the next state fulfills Φ
  - $\Phi \,\mathsf{U}\, \Psi$ — Φ holds until a Ψ-state is reached

$\Rightarrow$ note that $\bigcirc$ and U *alternate* with $\forall$ and $\exists$

Derived operators

potentially Φ:	$\exists\Diamond\Phi$	$=$	$\exists(\text{true}\ \mathsf{U}\ \Phi)$
inevitably Φ:	$\forall\Diamond\Phi$	$=$	$\forall(\text{true}\ \mathsf{U}\ \Phi)$
potentially always Φ:	$\exists\Box\Phi$	$:=$	$\neg\forall\Diamond\neg\Phi$
invariantly Φ:	$\forall\Box\Phi$	$=$	$\neg\exists\Diamond\neg\Phi$
weak until:	$\exists(\Phi\ \mathsf{W}\ \Psi)$	$=$	$\neg\forall\big((\Phi \wedge \neg\Psi)\ \mathsf{U}\ (\neg\Phi \wedge \neg\Psi)\big)$
	$\forall(\Phi\ \mathsf{W}\ \Psi)$	$=$	$\neg\exists\big((\Phi \wedge \neg\Psi)\ \mathsf{U}\ (\neg\Phi \wedge \neg\Psi)\big)$

the boolean connectives are derived as usual

Semantics of CTL state-formulas

Defined by a relation $\models$ such that

$s \models \Phi$ if and only if formula Φ holds in state s

$s \models a$	iff	$a \in L(s)$
$s \models \neg\Phi$	iff	$\neg(s \models \Phi)$
$s \models \Phi \wedge \Psi$	iff	$(s \models \Phi) \wedge (s \models \Psi)$
$s \models \exists\varphi$	iff	$\pi \models \varphi$ for *some* path π that starts in s
$s \models \forall\varphi$	iff	$\pi \models \varphi$ for *all* paths π that start in s

Semantics of CTL path-formulas

Define a relation $\models$ such that

$\pi \models \varphi$ if and only if path π satisfies φ

$$\pi \models \bigcirc \Phi \quad \text{iff } \pi[1] \models \Phi$$

$$\pi \models \Phi \, \mathsf{U} \, \Psi \quad \text{iff } (\exists\, j \geqslant 0.\, \pi[j] \models \Psi \ \wedge \ (\forall\, 0 \leqslant k < j.\, \pi[k] \models \Phi))$$

where $\pi[i]$ denotes the state s_i in the path π

Transition system semantics

- For CTL-state-formula Φ, the *satisfaction set* *Sat*(Φ) is defined by:

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

- *TS* satisfies CTL-formula Φ iff Φ holds in all its initial states:

$$\mathit{TS} \models \Phi \quad \text{if and only if} \quad \forall s_0 \in I.\, s_0 \models \Phi$$

  - this is equivalent to $I \subseteq \mathit{Sat}(\Phi)$

- Point of attention: *TS* $\not\models \Phi$ and *TS* $\not\models \neg\Phi$ is possible!
  - because of several initial states, e.g. $s_0 \models \exists\Box\,\Phi$ and $s_0' \not\models \exists\Box\,\Phi$

Overview Lecture #18

- Repetition: CTL syntax and semantics

$\Rightarrow$ CTL equivalence

- Expressiveness of LTL versus CTL
- CTL*: extended CTL

CTL equivalence

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

$$\Phi \equiv \Psi \quad \text{iff} \quad (TS \models \Phi \quad \text{if and only if} \quad TS \models \Psi)$$

Duality laws

$$\forall \bigcirc \Phi \equiv \neg \exists \bigcirc \neg \Phi$$

$$\exists \bigcirc \Phi \equiv \neg \forall \bigcirc \neg \Phi$$

$$\forall \Diamond \Phi \equiv \neg \exists \Box \neg \Phi$$

$$\exists \Diamond \Phi \equiv \neg \forall \Box \neg \Phi$$

$$\forall (\Phi \, \mathsf{U} \, \Psi) \equiv \neg \exists ((\Phi \wedge \neg \Psi) \, \mathsf{W} \, (\neg \Phi \wedge \neg \Psi))$$

Expansion laws

Recall in LTL: $\varphi \mathsf{U} \psi \equiv \psi \vee (\varphi \wedge \bigcirc (\varphi \mathsf{U} \psi))$

In CTL:

$$\begin{aligned}
\forall(\Phi \mathsf{U} \Psi) &\equiv \Psi \vee (\Phi \wedge \forall \bigcirc \forall(\Phi \mathsf{U} \Psi)) \\
\forall\Diamond\Phi &\equiv \Phi \vee \forall \bigcirc \forall\Diamond\Phi \\
\forall\Box\Phi &\equiv \Phi \wedge \forall \bigcirc \forall\Box\Phi \\
\\
\exists(\Phi \mathsf{U} \Psi) &\equiv \Psi \vee (\Phi \wedge \exists \bigcirc \exists(\Phi \mathsf{U} \Psi)) \\
\exists\Diamond\Phi &\equiv \Phi \vee \exists \bigcirc \exists\Diamond\Phi \\
\exists\Box\Phi &\equiv \Phi \wedge \exists \bigcirc \exists\Box\Phi
\end{aligned}$$

Distributive laws (1)

Recall in LTL: $\Box(\varphi \wedge \psi) \equiv \Box\varphi \wedge \Box\psi$ and $\Diamond(\varphi \vee \psi) \equiv \Diamond\varphi \vee \Diamond\psi$

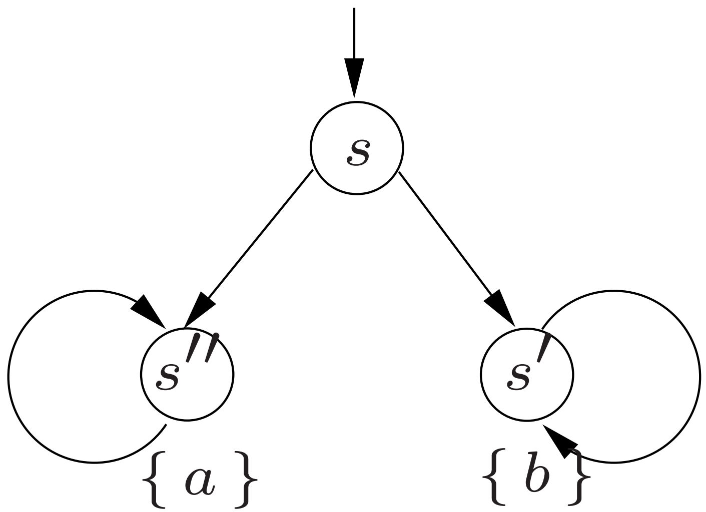
In CTL:

$$\forall\Box(\Phi \wedge \Psi) \equiv \forall\Box\Phi \wedge \forall\Box\Psi$$

$$\exists\Diamond(\Phi \vee \Psi) \equiv \exists\Diamond\Phi \vee \exists\Diamond\Psi$$

note that $\exists\Box(\Phi \wedge \Psi) \not\equiv \exists\Box\Phi \wedge \exists\Box\Psi$ and $\forall\Diamond(\Phi \vee \Psi) \not\equiv \forall\Diamond\Phi \vee \forall\Diamond\Psi$

Distributive laws (2)



$s \models \forall \Diamond (a \vee b)$ since for all $\pi \in \textit{Paths}(s).\ \pi \models \Diamond (a \vee b)$

But: $s\,(s'')^\omega \models \Diamond a$ but $s\,(s'')^\omega \not\models \Diamond b$ Thus: $s \not\models \forall \Diamond b$

A similar reasoning applied to path $s\,(s')^\omega$ yields $s \not\models \forall \Diamond a$

Thus, $s \not\models \forall \Diamond a \vee \forall \Diamond b$

Overview Lecture #18

- Repetition: CTL syntax and semantics
- CTL equivalence

⇒ Expressiveness of LTL versus CTL

- CTL*: extended CTL

Equivalence of LTL and CTL formulas

- CTL-formula Φ and LTL-formula φ (both over *AP*) are *equivalent*, denoted $\Phi \equiv \varphi$, if for any transition system *TS* (over *AP*):

$$TS \models \Phi \quad \text{if and only if} \quad TS \models \varphi$$

- Let Φ be a CTL-formula, and φ the LTL-formula obtained by eliminating all path quantifiers in Φ. Then: [Clarke & Draghicescu]

> $\Phi \equiv \varphi$ or there does not exist any LTL-formula that is equivalent to Φ

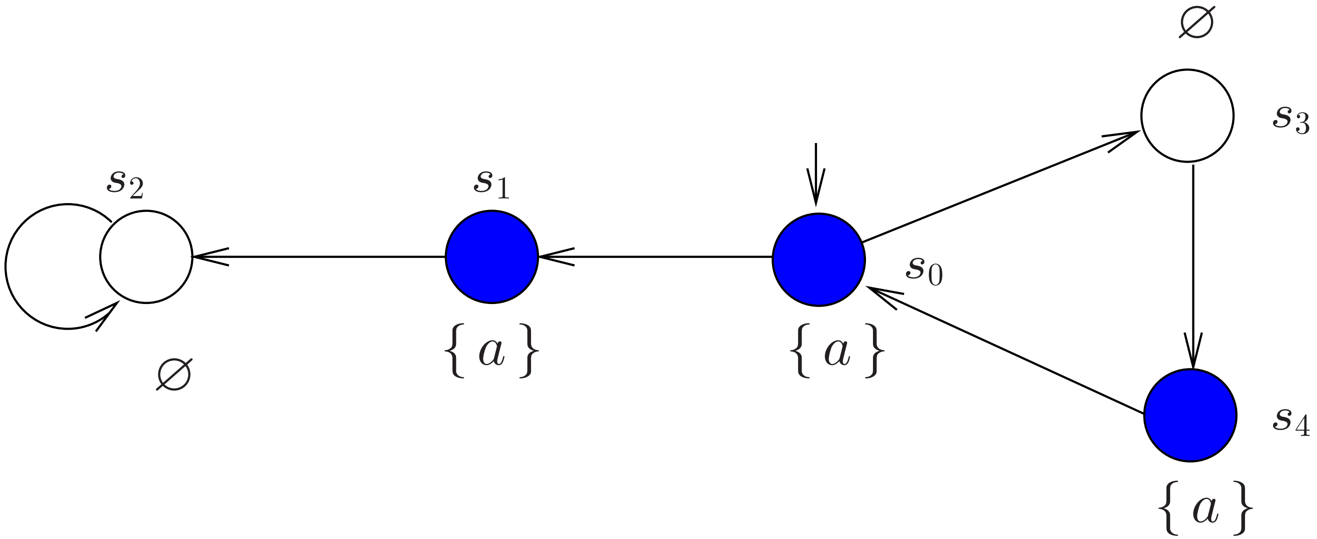
LTL and CTL are incomparable

- Some LTL-formulas cannot be expressed in CTL, e.g.,
  - $\Diamond \Box a$
  - $\Diamond (a \wedge \bigcirc a)$
- Some CTL-formulas cannot be expressed in LTL, e.g.,
  - $\forall\Diamond \forall\Box a$
  - $\forall\Diamond (a \wedge \forall\bigcirc a)$
  - $\forall\Box \exists\Diamond a$

$\Rightarrow$ Cannot be expressed = there does not exist an equivalent formula

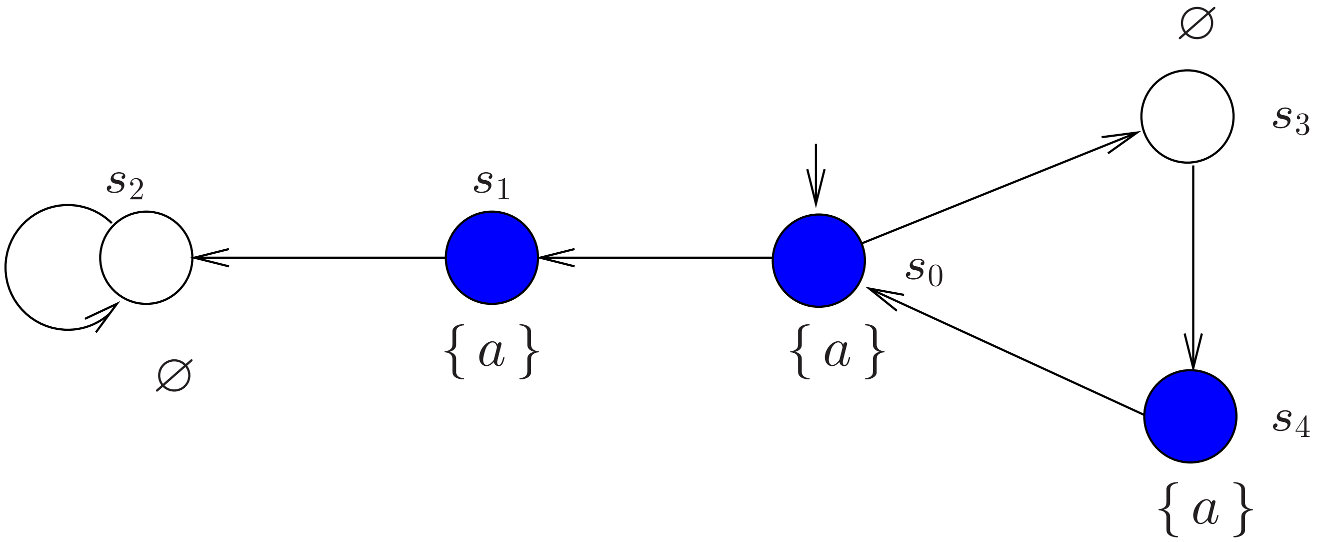
Comparing LTL and CTL (1)

$\Diamond (a \wedge \bigcirc a)$ is not equivalent to $\forall\Diamond (a \wedge \forall\bigcirc a)$



Comparing LTL and CTL (1)

$\Diamond (a \wedge \bigcirc a)$ is not equivalent to $\forall\Diamond (a \wedge \forall\bigcirc a)$

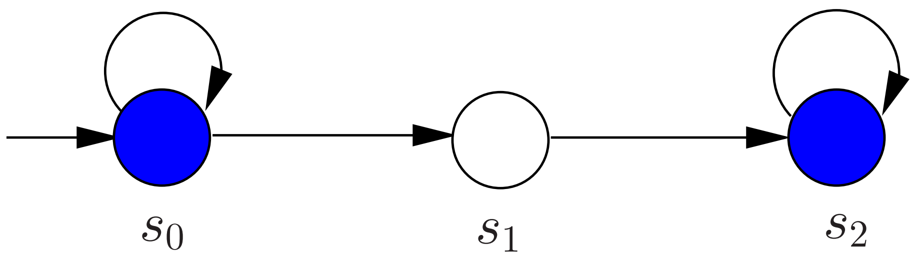


$s_0 \models \Diamond (a \wedge \bigcirc a)$ but $\underbrace{s_0 \not\models \forall\Diamond (a \wedge \forall\bigcirc a)}$

path $s_0\, s_1\, (s_2)^\omega$ violates it

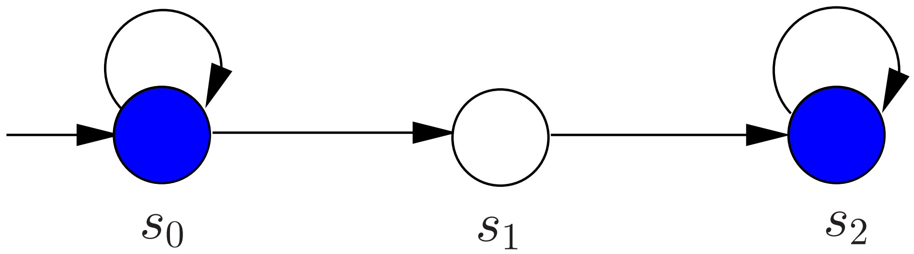
Comparing LTL and CTL (2)

$\forall\Diamond\,\forall\Box\, a$ is not equivalent to $\Diamond\,\Box\, a$



Comparing LTL and CTL (2)

$\forall\Diamond\,\forall\Box\, a$ is not equivalent to $\Diamond\,\Box\, a$

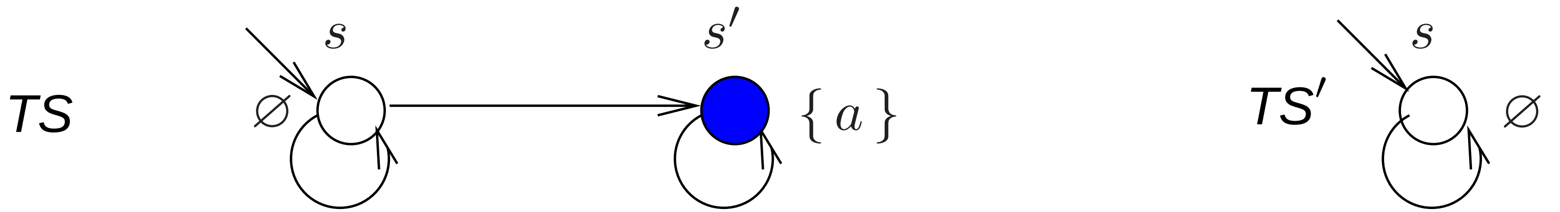


$s_0 \models \Diamond\,\Box\, a$ but $\underbrace{s_0 \not\models \forall\Diamond\,\forall\Box\, a}_{\text{path } s_0^\omega \text{ violates it}}$

Comparing LTL and CTL (3)

The CTL-formula $\forall\Box\,\exists\Diamond\, a$ cannot be expressed in LTL

- This is shown by contradiction: assume $\varphi \;\equiv\; \forall\Box\,\exists\Diamond\, a$; let:



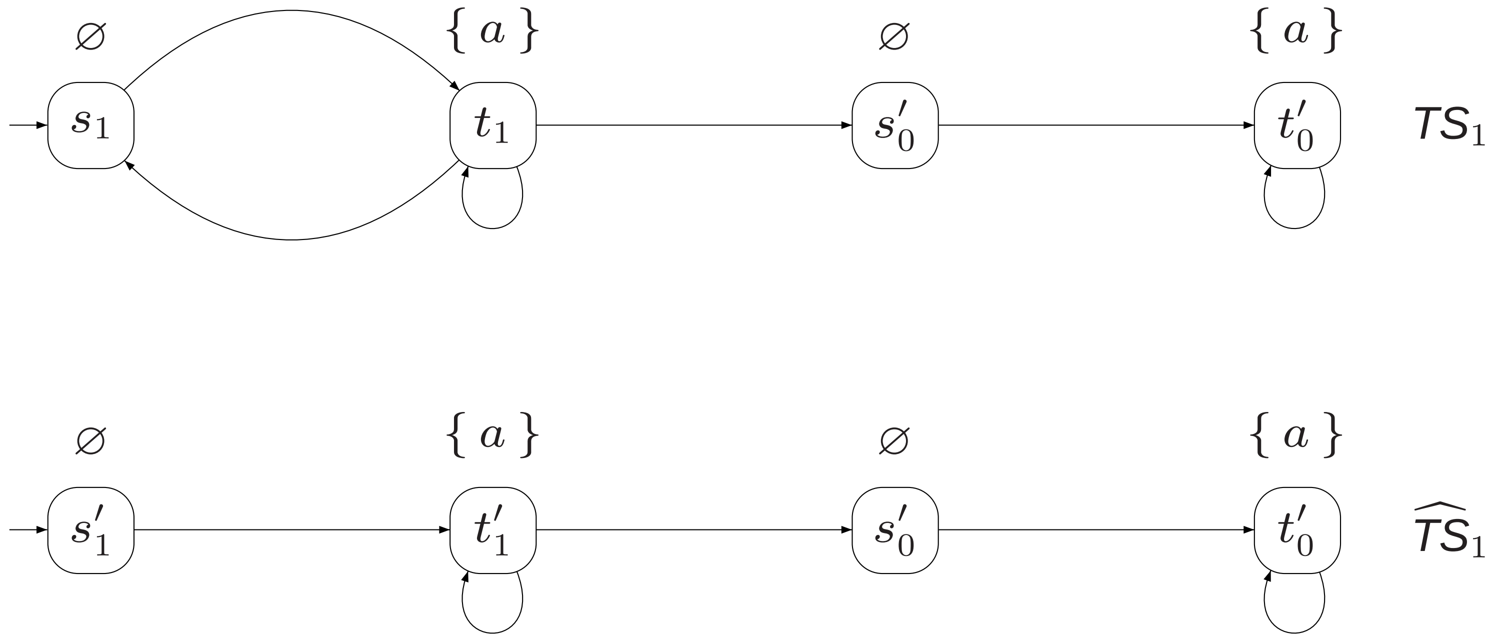
- $TS \models \forall\Box\,\exists\Diamond\, a$, and thus—by assumption—$TS \models \varphi$
- $Paths(TS') \subseteq Paths(TS)$, thus $TS' \models \varphi$
- But $TS' \not\models \forall\Box\,\exists\Diamond\, a$ as path $s^\omega \not\models \;\Box\,\exists\Diamond\, a$

Comparing LTL and CTL (4)

The LTL-formula $\Diamond\Box\, a$ cannot be expressed in CTL

- Provide two series of transition systems TS_n and $\widehat{TS}_n$
- Such that $TS_n \not\models \Diamond\Box\, a$ and $\widehat{TS}_n \models \Diamond\Box\, a$ (*), and
- for any $\forall$CTL-formula Φ with $|\Phi| \leqslant n : TS_n \models \Phi$ iff $\widehat{TS}_n \models \Phi$ (**)
  - proof is by induction on n (omitted here)
- Assume there is a CTL-formula $\Phi \equiv \Diamond\Box\, a$ with $|\Phi| = n$
  - by (*), it follows $TS_n \not\models \Phi$ and $\widehat{TS}_n \models \Phi$
  - but this contradicts (**): $TS_n \models \Phi$ if and only if $\widehat{TS}_n \models \Phi$

The transition systems TS_n and $\widehat{TS}_n$ $(n = 1)$



only difference: TS_n includes $t_n \longrightarrow s_n$, while $\widehat{TS}_n$ does not

Overview Lecture #18

- Repetition: CTL syntax and semantics
- CTL equivalence
- Expressiveness of LTL versus CTL

$\Rightarrow$ CTL*: extended CTL

Syntax of CTL*

CTL* *state-formulas* are formed according to:

$$\Phi ::= \text{true} \;\Big|\; a \;\Big|\; \Phi_1 \wedge \Phi_2 \;\Big|\; \neg\Phi \;\Big|\; \exists\varphi$$

where $a \in AP$ and φ is a path-formula

CTL* *path-formulas* are formed according to the grammar:

$$\varphi ::= \Phi \;\Big|\; \varphi_1 \wedge \varphi_2 \;\Big|\; \neg\varphi \;\Big|\; \bigcirc\varphi \;\Big|\; \varphi_1 \,\mathsf{U}\, \varphi_2$$

where Φ is a state-formula, and φ, φ_1 and φ_2 are path-formulas

in CTL*: $\forall\varphi = \neg\exists\neg\varphi$. This does not hold in CTL!

Example CTL* formulas

CTL* semantics

$s \models a$	iff	$a \in L(s)$
$s \models \neg\Phi$	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 $\pi \in$ *Paths*(s)

$\pi \models \Phi$	iff	$\pi[0] \models \Phi$
$\pi \models \varphi_1 \wedge \varphi_2$	iff	$\pi \models \varphi_1$ and $\pi \models \varphi_2$
$\pi \models \neg\varphi$	iff	$\pi \not\models \varphi$
$\pi \models \bigcirc\Phi$	iff	$\pi[1..] \models \Phi$
$\pi \models \Phi \,\mathsf{U}\, \Psi$	iff	$\exists\, j \geqslant 0.\ (\pi[j..] \models \Psi \ \wedge\ (\forall\, 0 \leqslant k < j.\ \pi[k..] \models \Phi))$

Transition system semantics

- For CTL*-state-formula Φ, the *satisfaction set* $\mathit{Sat}(\Phi)$ is defined by:

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

- *TS* satisfies CTL*-formula Φ iff Φ holds in all its initial states:

$$\mathit{TS} \models \Phi \quad \text{if and only if} \quad \forall s_0 \in I.\, s_0 \models \Phi$$

this is exactly as for CTL

Embedding of LTL in CTL*

For LTL formula φ and *TS* without terminal states (both over *AP*) and for each $s \in S$:

$$\underbrace{s \models \varphi}_{\text{LTL semantics}} \quad \text{if and only if} \quad \underbrace{s \models \forall \varphi}_{\text{CTL}^* \text{ semantics}}$$

In particular:

$$TS \models_{LTL} \varphi \quad \text{if and only if} \quad TS \models_{CTL*} \forall \varphi$$

CTL* is more expressive than LTL and CTL

For the CTL*-formula over $AP = \{ a, b \}$:

$$\Phi = (\forall \Diamond \Box a) \vee (\forall \Box \exists \Diamond b)$$

there does *not* exist any equivalent LTL- or CTL formula

This logic is as expressive as CTL

CTL$^+$ *state-formulas* are formed according to:

$$\Phi ::= \text{true} \;\Big|\; a \;\Big|\; \Phi_1 \wedge \Phi_2 \;\Big|\; \neg\Phi \;\Big|\; \exists\varphi \;\Big|\; \forall\varphi$$

where $a \in AP$ and φ is a path-formula

CTL$^+$ *path-formulas* are formed according to the grammar:

$$\varphi ::= \varphi_1 \wedge \varphi_2 \;\Big|\; \neg\varphi \;\Big|\; \bigcirc\, \Phi \;\Big|\; \Phi_1 \,\mathsf{U}\, \Phi_2$$

where Φ, Φ_1, Φ_2 are state-formulas, and φ, φ_1 and φ_2 are path-formulas

CTL$^+$ is as expressive as CTL

For example:

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

Some rules for transforming CTL$^+$ formulae into equivalent CTL ones:

$$\exists\Big(\neg(\Phi_1 \,\mathsf{U}\, \Phi_2)\Big) \equiv \exists\Big(\,(\Phi_1 \wedge \neg\Phi_2)\,\mathsf{U}\,(\neg\Phi_1 \wedge \neg\Phi_2)\,\Big) \;\vee\; \exists\Box\neg\Phi_2$$

$$\exists\Big(\bigcirc\Phi_1 \wedge \bigcirc\Phi_2\Big) \equiv \exists\bigcirc(\Phi_1 \wedge \Phi_2)$$

$$\exists\Big(\bigcirc\Phi \wedge (\Phi_1 \,\mathsf{U}\, \Phi_2)\Big) \equiv \Big(\Phi_2 \wedge \exists\bigcirc\Phi\Big) \;\vee\; \Big(\Phi_1 \wedge \exists\bigcirc(\Phi \wedge \exists(\Phi_1 \,\mathsf{U}\, \Phi_2))\Big)$$

$$\exists\Big((\Phi_1 \,\mathsf{U}\, \Phi_2) \;\wedge\; (\Psi_1 \,\mathsf{U}\, \Psi_2)\Big) \equiv \exists\Big((\Phi_1 \wedge \Psi_1)\,\mathsf{U}\,(\Phi_2 \wedge \exists(\Psi_1 \,\mathsf{U}\, \Psi_2))\Big) \;\vee\; \exists\Big((\Phi_1 \wedge \Psi_1)\,\mathsf{U}\,(\Psi_2 \wedge \exists(\Phi_1 \,\mathsf{U}\, \Phi_2))\Big)$$

$$\vdots$$

adding boolean combinations of path formulae to CTL does not change its expressiveness

but CTL$^+$ formulae can be much shorter than shortest equivalent CTL formulae

Relationship between LTL, CTL and CTL*