Linear Temporal Logic

Lecture #13 of Model Checking

Joost-Pieter Katoen

Lehrstuhl 2: Software Modeling & Verification

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

December 2, 2008

Overview Lecture #12

- Syntax
- Semantics
- Equivalence

LT properties

- An LT property is a set of infinite traces over *AP*
- Specifying such sets explicitly is often inconvenient
- Mutual exclusion is specified over *AP* $= \{ c_1, c_2 \}$ by

$$P_{mutex} \; = \; \text{set of infinite words } A_0\, A_1\, A_2 \ldots \text{ with } \{ c_1, c_2 \} \not\subseteq A_i \text{ for all } 0 \leqslant i$$

- Starvation freedom is specified over *AP* $= \{ c_1, w_1, c_2, w_2 \}$ by

$$P_{nostarve} = \; \text{set of infinite words } A_0\, A_1\, A_2 \ldots \text{ such that:}$$

$$\left(\overset{\infty}{\exists}\, j.\; w_1 \in A_j \right) \Rightarrow \left(\overset{\infty}{\exists}\, j.\; c_1 \in A_j \right) \wedge \left(\overset{\infty}{\exists}\, j.\; w_2 \in A_j \right) \Rightarrow \left(\overset{\infty}{\exists}\, j.\; c_2 \in A_j \right)$$

such properties can be specified succinctly using logic

Syntax

modal logic over infinite sequences [Pnueli 1977]

- Propositional logic
  - $a \in AP$ atomic proposition
  - $\neg\phi$ and $\phi \wedge \psi$ negation and conjunction
- Temporal operators
  - $\bigcirc \phi$ neXt state fulfills ϕ
  - $\phi \, \mathsf{U} \, \psi$ ϕ holds Until a ψ-state is reached

linear temporal logic is a logic for describing LT properties

Derived operators

$$\phi \lor \psi \equiv \neg(\neg\phi \land \neg\psi)$$

$$\phi \Rightarrow \psi \equiv \neg\phi \lor \psi$$

$$\phi \Leftrightarrow \psi \equiv (\phi \Rightarrow \psi) \land (\psi \Rightarrow \phi)$$

$$\phi \oplus \psi \equiv (\phi \land \neg\psi) \lor (\neg\phi \land \psi)$$

$$\text{true} \equiv \phi \lor \neg\phi$$

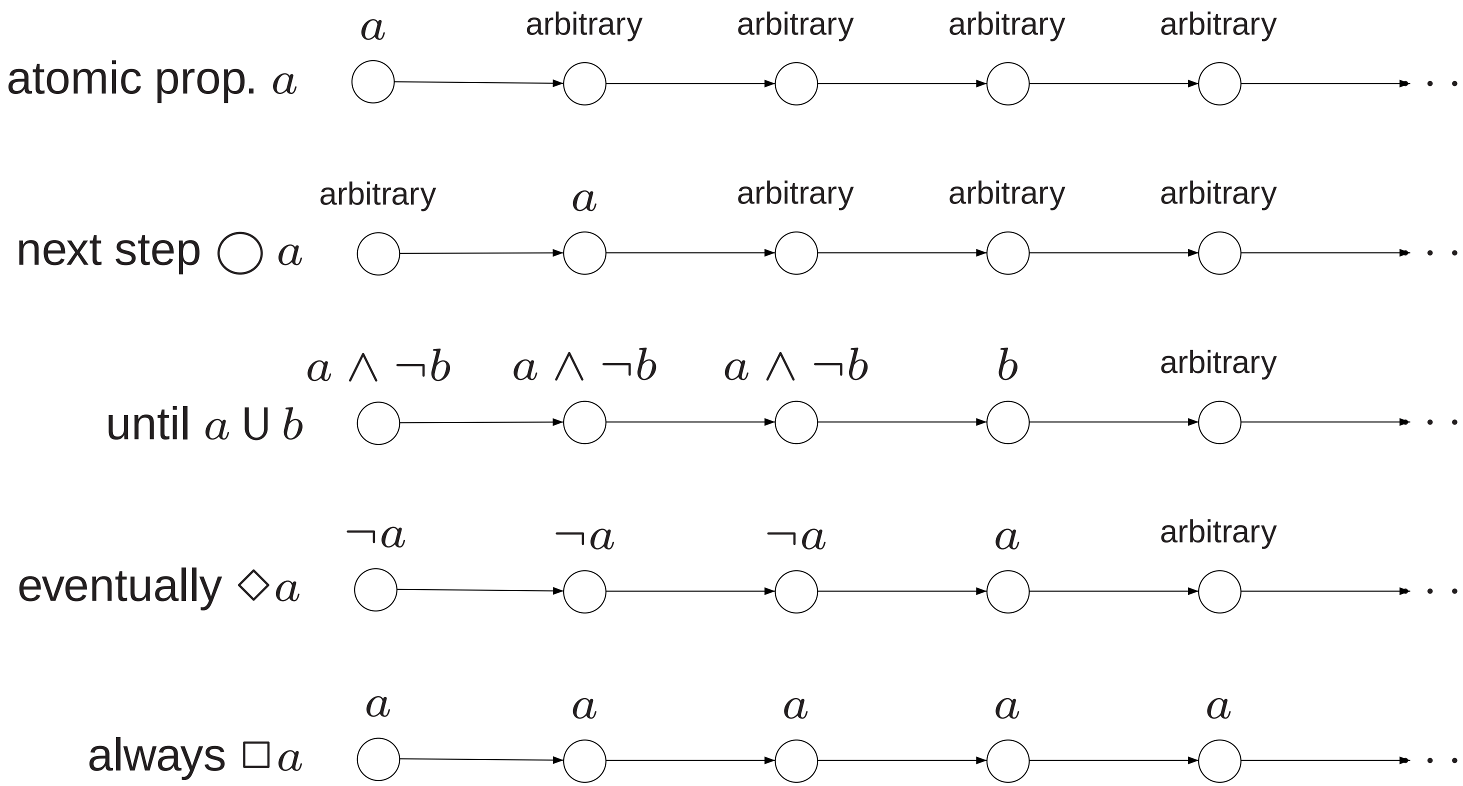
$$\text{false} \equiv \neg\,\text{true}$$

$$\Diamond\phi \equiv \text{true}\,\mathsf{U}\,\phi \qquad \text{“sometimes in the future”}$$

$$\Box\phi \equiv \neg\Diamond\neg\phi \qquad \text{“from now on for ever”}$$

precedence order: the unary operators bind stronger than the binary ones.
$\neg$ and $\bigcirc$ bind equally strong. U takes precedence over $\wedge$, $\vee$, and $\longrightarrow$

Intuitive semantics



Traffic light properties

- Once red, the light cannot become green immediately:

$$\Box\,(\mathit{red}\ \Rightarrow\ \neg \bigcirc\ \mathit{green})$$

- The green light becomes green eventually: $\Diamond\, \mathit{green}$
- Once red, the light becomes green eventually: $\Box\,(\mathit{red}\ \Rightarrow\ \Diamond\, \mathit{green})$
- Once red, the light always becomes green eventually after being yellow for some time inbetween:

$$\Box(\mathit{red} \rightarrow \bigcirc\,(\mathit{red}\ \mathsf{U}\ (\mathit{yellow}\ \wedge\ \bigcirc\ (\mathit{yellow}\ \mathsf{U}\ \mathit{green}))))$$

Practical properties in LTL

- Reachability
  - negated reachability $\Diamond \neg \psi$
  - conditional reachability $\phi \, \mathsf{U} \, \psi$
  - reachability from any state not expressible
- Safety
  - simple safety $\Box \neg \phi$
  - conditional safety $(\phi \, \mathsf{U} \, \psi) \lor \Diamond \phi$
- Liveness $\Box (\phi \Rightarrow \Diamond \psi)$ and others
- Fairness $\Box \Diamond \phi$ and others

Semantics over words

The LT-property induced by LTL formula φ over *AP* is:

$Words(\varphi) \; = \; \left\{ \sigma \in \left(2^{AP}\right)^{\omega} \mid \sigma \models \varphi \right\}$, where $\models$ is the smallest relation satisfying:

$$
\begin{array}{lllll}
\sigma & \models & \text{true} & & \\
\sigma & \models & a & \text{iff} & a \in A_0 \quad (\text{i.e., } A_0 \models a) \\
\sigma & \models & \varphi_1 \wedge \varphi_2 & \text{iff} & \sigma \models \varphi_1 \text{ and } \sigma \models \varphi_2 \\
\sigma & \models & \neg\, \varphi & \text{iff} & \sigma \not\models \varphi \\
\sigma & \models & \bigcirc\, \varphi & \text{iff} & \sigma[1..] = A_1 A_2 A_3 \ldots \models \varphi \\
\sigma & \models & \varphi_1 \,\mathsf{U}\, \varphi_2 & \text{iff} & \exists j \geqslant 0.\ \sigma[j..] \models \varphi_2 \text{ and } \sigma[i..] \models \varphi_1,\ 0 \leqslant i < j
\end{array}
$$

for $\sigma = A_0 A_1 A_2 \ldots$ we have $\sigma[i..] = A_i A_{i+1} A_{i+2} \ldots$ is the suffix of σ from index i on

Semantics of $\Box$, $\Diamond$, $\Box\Diamond$ and $\Diamond\Box$

$$
\begin{array}{llll}
\sigma & \models & \Diamond\varphi & \text{iff} \quad \exists j \geqslant 0.\ \sigma[j..] \models \varphi \\
\sigma & \models & \Box\varphi & \text{iff} \quad \forall j \geqslant 0.\ \sigma[j..] \models \varphi \\
\sigma & \models & \Box\Diamond\varphi & \text{iff} \quad \forall j \geqslant 0.\, \exists i \geqslant j.\ \sigma[i \ldots] \models \varphi \\
\sigma & \models & \Diamond\Box\varphi & \text{iff} \quad \exists j \geqslant 0. \forall j \geqslant i.\ \sigma[j \ldots] \models \varphi
\end{array}
$$

Semantics over paths and states

Let $TS = (S, Act, \rightarrow, I, AP, L)$ be a transition system and φ be an LTL-formula over AP.

- For infinite path fragment π of TS:

$$\pi \models \varphi \qquad \text{iff} \qquad trace(\pi) \models \varphi$$

- For state $s \in S$:

$$s \models \varphi \qquad \text{iff} \qquad \forall \pi \in Paths(s).\ \pi \models \varphi$$

- TS satisfies φ, denoted $TS \models \varphi$, iff $Traces(TS) \subseteq Words(\varphi)$

Semantics for transition systems

$\mathit{TS} \models \varphi$

iff (* transition system semantics *)

$\mathit{Traces}(\mathit{TS}) \subseteq \mathit{Words}(\varphi)$

iff (* definition of $\models$ for LT-properties *)

$\mathit{TS} \models \mathit{Words}(\varphi)$

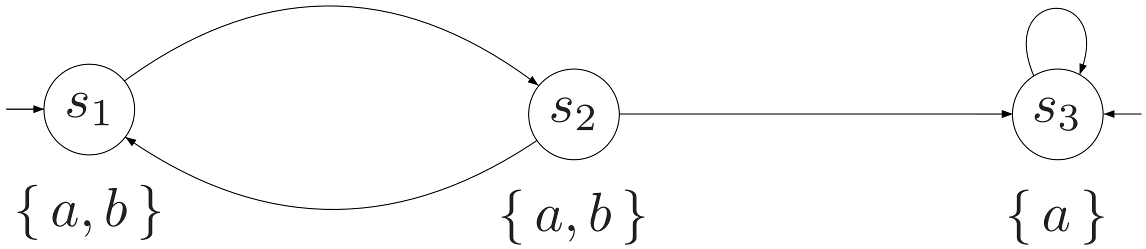
iff (* Definition of *Words*(φ) *)

$\pi \models \varphi$ for all $\pi \in \mathit{Paths}(\mathit{TS})$

iff (* semantics of $\models$ for states *)

$s_0 \models \varphi$ for all $s_0 \in I$.

Example



$$TS \models \Box a \quad TS \not\models \bigcirc (a \wedge b)$$

$$TS \models \Box(\neg b \;\Rightarrow\; \Box(a \wedge \neg b)) \quad TS \not\models b \,\mathsf{U}\, (a \wedge \neg b)$$

Semantics of negation

For paths, it holds $\pi \models \varphi$ if and only if $\pi \not\models \neg\varphi$ since:

$$\mathit{Words}(\neg\varphi) = \left(2^{AP}\right)^{\omega} \setminus \mathit{Words}(\varphi) \quad .$$

But: $TS \not\models \varphi$ and $TS \models \neg\varphi$ are *not* equivalent in general

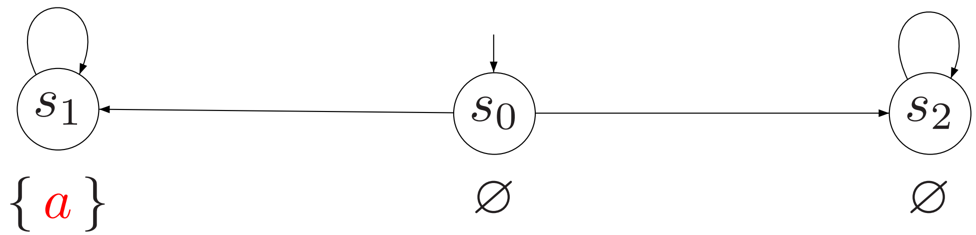
It holds: $TS \models \neg\varphi$ implies $TS \not\models \varphi$. Not always the reverse!

Note that:

$$\begin{aligned} TS \not\models \varphi \quad & \text{iff } \mathit{Traces}(TS) \not\subseteq \mathit{Words}(\varphi) \\ & \text{iff } \mathit{Traces}(TS) \setminus \mathit{Words}(\varphi) \neq \varnothing \\ & \text{iff } \mathit{Traces}(TS) \cap \mathit{Words}(\neg\varphi) \neq \varnothing \quad . \end{aligned}$$

TS neither satisfies φ nor $\neg\varphi$ if there are
paths π_1 and π_2 in *TS* such that $\pi_1 \models \varphi$ and $\pi_2 \models \neg\varphi$

Example



A transition system for which *TS* $\not\models \Diamond a$ and *TS* $\not\models \neg\Diamond a$

Specifying properties in LTL

Equivalence

LTL formulas ϕ, ψ are *equivalent*, denoted $\phi \equiv \psi$, if:

$$\textit{Words}(\phi) = \textit{Words}(\psi)$$

Duality and idempotence laws

Duality:

$$\neg \Box \phi \equiv \Diamond \neg \phi$$

$$\neg \Diamond \phi \equiv \Box \neg \phi$$

$$\neg \bigcirc \phi \equiv \bigcirc \neg \phi$$

Idempotency:

$$\Box \Box \phi \equiv \Box \phi$$

$$\Diamond \Diamond \phi \equiv \Diamond \phi$$

$$\phi \,\mathsf{U}\, (\phi \,\mathsf{U}\, \psi) \equiv \phi \,\mathsf{U}\, \psi$$

$$(\phi \,\mathsf{U}\, \psi) \,\mathsf{U}\, \psi \equiv \phi \,\mathsf{U}\, \psi$$

Absorption and distributive laws

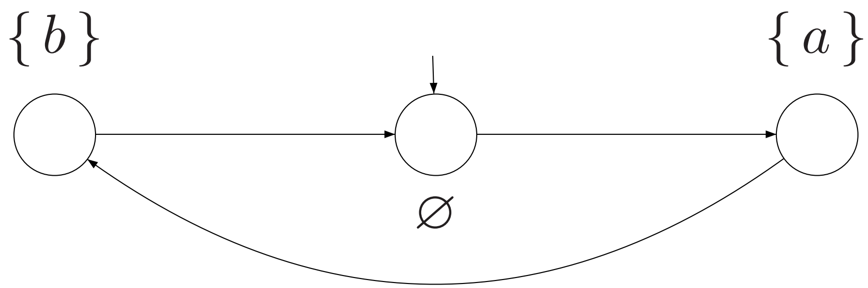
Absorption:
$$\Diamond \Box \Diamond \phi \equiv \Box \Diamond \phi$$
$$\Box \Diamond \Box \phi \equiv \Diamond \Box \phi$$

Distribution:
$$\bigcirc (\phi \, \mathsf{U} \, \psi) \equiv (\bigcirc \phi) \, \mathsf{U} \, (\bigcirc \psi)$$
$$\Diamond(\phi \vee \psi) \equiv \Diamond\phi \vee \Diamond\psi$$
$$\Box(\phi \wedge \psi) \equiv \Box\phi \wedge \Box\psi$$

but . . . . . .:
$$\Diamond(\phi \, \mathsf{U} \, \psi) \not\equiv (\Diamond\phi) \, \mathsf{U} \, (\Diamond\psi)$$
$$\Diamond(\phi \wedge \psi) \not\equiv \Diamond\phi \wedge \Diamond\psi$$
$$\Box(\phi \vee \psi) \not\equiv \Box\phi \vee \Box\psi$$

Distributive laws

$$\Diamond(a \wedge b) \not\equiv \Diamond a \wedge \Diamond b \quad \text{and} \quad \Box(a \vee b) \not\equiv \Box a \vee \Box b$$



TS $\not\models \Diamond(a \wedge b)$ and *TS* $\models \Diamond a \wedge \Diamond b$

CTL, LTL and CTL*

Lecture #19 of Model Checking

Joost-Pieter Katoen

Lehrstuhl 2: Software Modeling & Verification

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

January 7, 2009

Overview Lecture #19

⇒ 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:	$\exists\Diamond\Phi$	$=$	$\exists(\mathsf{true}\,\mathsf{U}\,\Phi)$
inevitably Φ:	$\forall\Diamond\Phi$	$=$	$\forall(\mathsf{true}\,\mathsf{U}\,\Phi)$
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

$\boxed{s \models \Phi \text{ if and only if formula } \Phi \text{ holds in state } s}$

$s \models a$	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

$\boxed{\pi \models \varphi \text{ if and only if path } \pi \text{ satisfies } \varphi}$

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

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

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 #19

- 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:

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

Distributive laws (1)

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

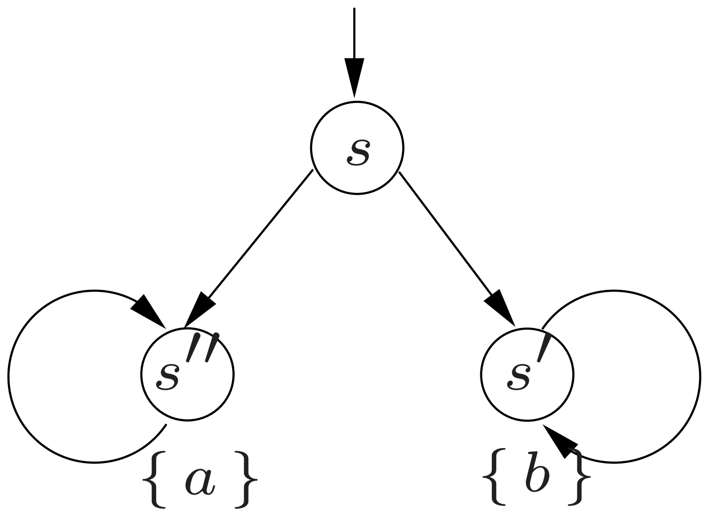
In CTL:

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

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

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 $\pi \in$ *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 #19

- Repetition: CTL syntax and semantics
- CTL equivalence

$\Rightarrow$ 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]

$$\boxed{\Phi \equiv \varphi \text{ or there does not exist any LTL-formula that is equivalent to } \Phi}$$

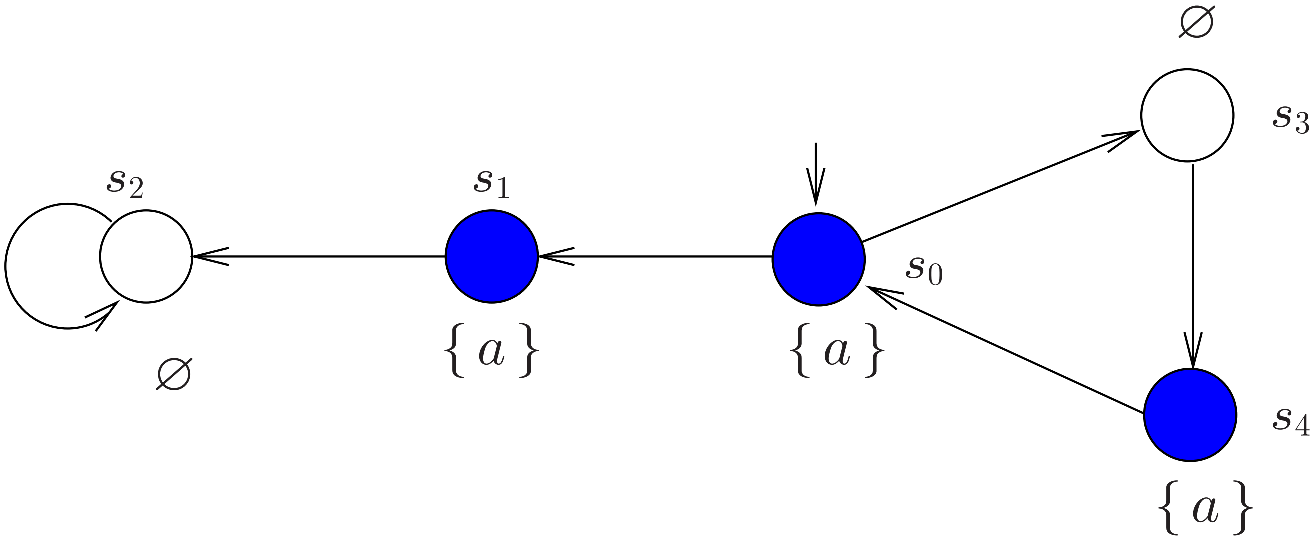
LTL and CTL are incomparable

- 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

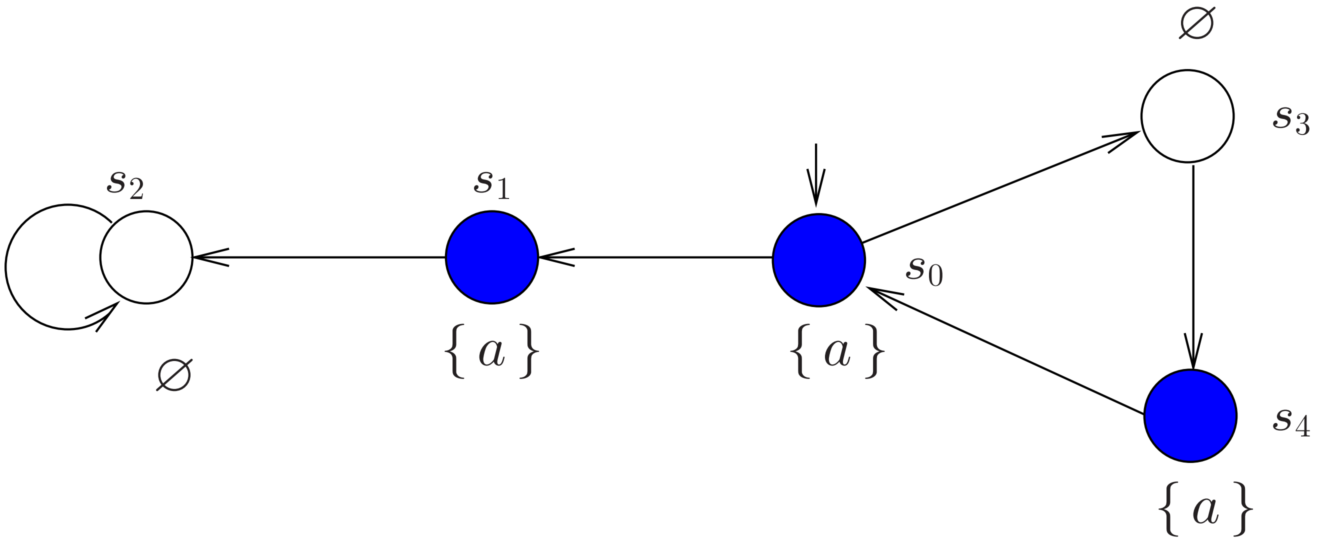
Comparing LTL and CTL (1)

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



Comparing LTL and CTL (1)

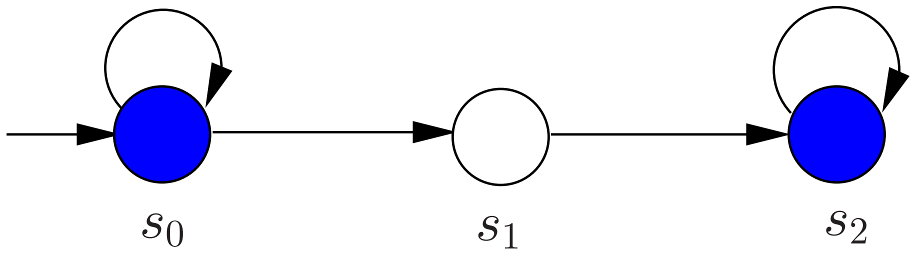
$\Diamond (a \wedge \bigcirc a)$ is not equivalent to $\forall \Diamond (a \wedge \forall \bigcirc a)$



$s_0 \models \Diamond (a \wedge \bigcirc a)$ but $\underbrace{s_0 \not\models \forall \Diamond (a \wedge \forall \bigcirc a)}_{\text{path } s_0\, s_1\, (s_2)^\omega \text{ violates it}}$

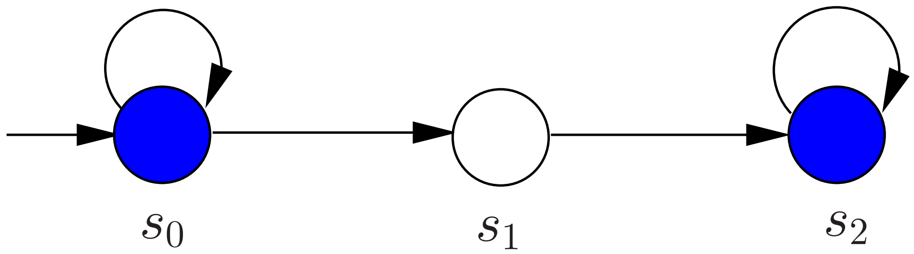
Comparing LTL and CTL (2)

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



Comparing LTL and CTL (2)

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

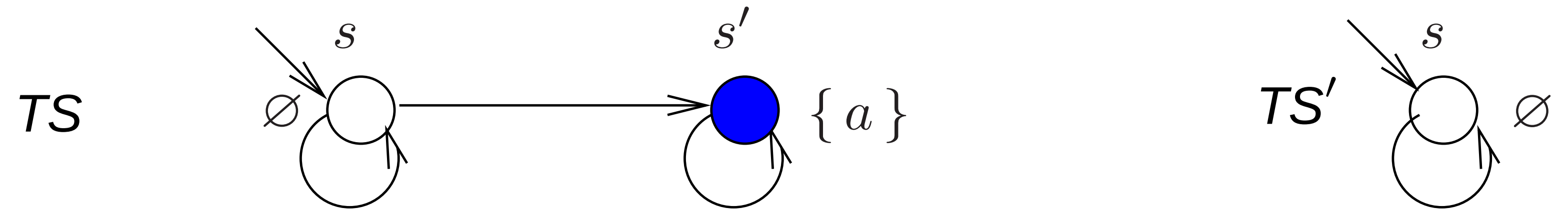


$s_0 \models \Diamond\,\Box\, a$ but $\underbrace{s_0 \not\models \forall\Diamond\,\forall\Box\, a}_{\text{path } s_0^\omega \text{ violates it}}$

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:



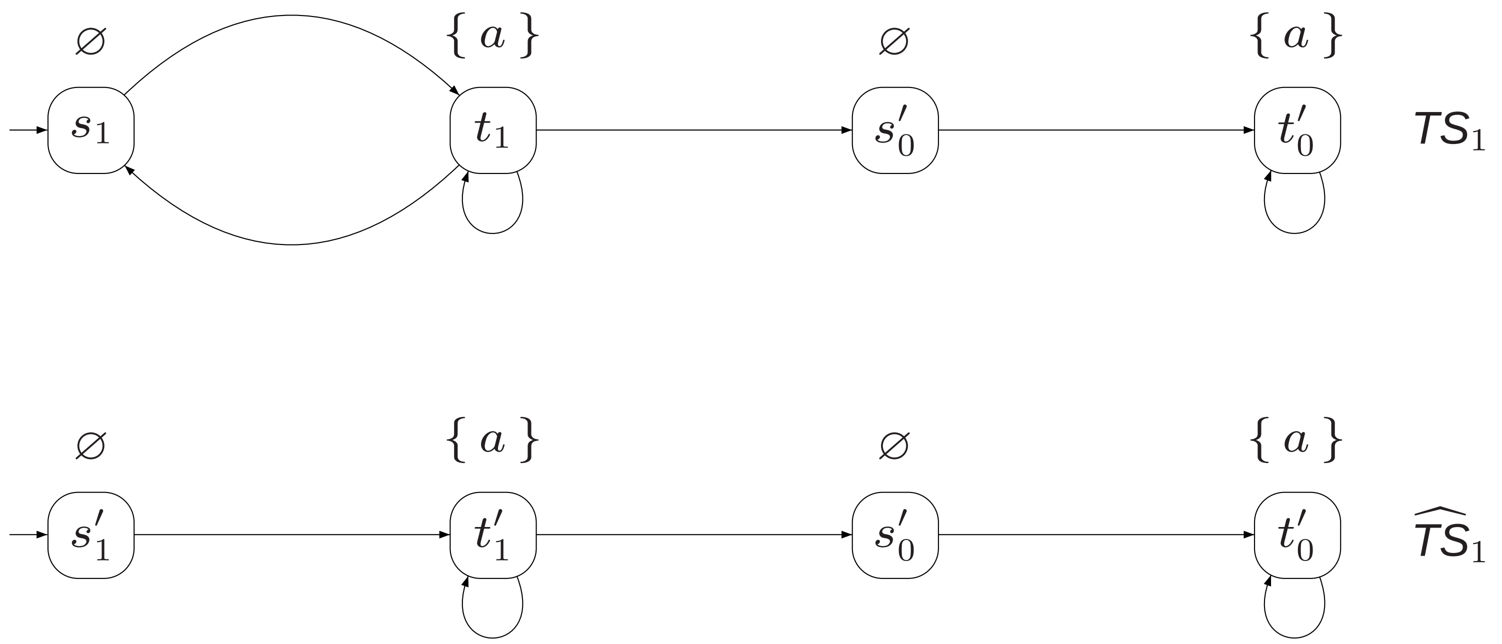
- $TS \models \forall\Box\,\exists\Diamond\, a$, and thus—by assumption—$TS \models \varphi$
- $Paths(TS') \subseteq Paths(TS)$, thus $TS' \models \varphi$
- But $TS' \not\models \forall\Box\,\exists\Diamond\, a$ as path $s^\omega \not\models\; \Box\,\exists\Diamond\, a$

Comparing LTL and CTL (4)

$\boxed{\text{The LTL-formula } \Diamond\Box\, a \text{ cannot be expressed in CTL}}$

- 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 #19

- 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 $\pi \in \mathit{Paths}(s)$

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

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 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:

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

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:

$$\begin{aligned}
\exists\Big(\neg(\Phi_1 \,\mathsf{U}\, \Phi_2)\Big) &\equiv \exists\Big(\,(\Phi_1 \wedge \neg\Phi_2)\,\mathsf{U}\,(\neg\Phi_1 \wedge \neg\Phi_2)\,\Big) \;\vee\; \exists\Box\neg\Phi_2 \\
\exists\Big(\bigcirc\Phi_1 \wedge \bigcirc\Phi_2\Big) &\equiv \exists\bigcirc(\Phi_1 \wedge \Phi_2) \\
\exists\Big(\bigcirc\Phi \wedge (\Phi_1 \,\mathsf{U}\, \Phi_2)\Big) &\equiv \Big(\Phi_2 \wedge \exists\bigcirc\Phi\Big) \;\vee\; \Big(\Phi_1 \wedge \exists\bigcirc(\Phi \wedge \exists(\Phi_1 \,\mathsf{U}\, \Phi_2))\Big) \\
\exists\Big((\Phi_1 \,\mathsf{U}\, \Phi_2) \;\wedge\; (\Psi_1 \,\mathsf{U}\, \Psi_2)\Big) &\equiv \exists\Big((\Phi_1 \wedge \Psi_1)\,\mathsf{U}\,(\Phi_2 \wedge \exists(\Psi_1 \,\mathsf{U}\, \Psi_2))\Big) \;\vee \\
&\; \exists\Big((\Phi_1 \wedge \Psi_1)\,\mathsf{U}\,(\Psi_2 \wedge \exists(\Phi_1 \,\mathsf{U}\, \Phi_2))\Big) \\
&\vdots
\end{aligned}$$

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*