Probabilistic Model Checking

Alessandro Abate

Lecture 4, p3: Computation Tree Logic

On the expressiveness of LTL

- at the end of Lecture 4, we established that the property

$$\forall \pi \in Paths(TS) : \forall m \geq 0 : \exists \pi' \in Paths(\pi[m]) : \exists n \geq 0 : \pi'[n] \models a$$

- cannot be captured by LTL

- let us unravel this formula by looking into its structure

The need for a different temporal logic

$$\forall \pi \in \mathit{Paths}(TS) : \forall m \geq 0 : \exists \pi' \in \mathit{Paths}(\pi[m]) : \underbrace{\exists n \geq 0 : \pi'[n] \models a}_{\Diamond a}$$

The need for a different temporal logic

$$\forall \pi \in Paths(TS) : \forall m \geq 0 : \overbrace{\exists \pi' \in Paths(\pi[m]) : \underbrace{\exists n \geq 0 : \pi'[n] \models a}_{\Diamond a}}^{\exists \Diamond a}$$

The need for a different temporal logic

$$\forall \pi \in \mathit{Paths}(\mathit{TS}) : \underbrace{\forall m \geq 0 : \overbrace{\exists \pi' \in \mathit{Paths}(\pi[m]) : \underbrace{\exists n \geq 0 : \pi'[n] \models a}_{\Diamond a}}^{\exists \Diamond a}}_{\Box \exists \Diamond a}$$

The need for a different temporal logic

$$\overbrace{\forall \pi \in \mathit{Paths}(TS) : \underbrace{\forall m \geq 0 : \overbrace{\exists \pi' \in \mathit{Paths}(\pi[m]) : \underbrace{\exists n \geq 0 : \pi'[n] \models a}_{\Diamond a}}^{\exists \Diamond a}}_{\Box \exists \Diamond a}}^{\forall \Box \exists \Diamond a}$$

The need for a different temporal logic

$$\overbrace{\exists \pi' \in Paths(\pi[m]) : \underbrace{\exists n \geq 0 : \pi'[n] \models a}_{\Diamond a}}^{\exists \Diamond a}$$

- premise: $\pi \models \Diamond a$ means that path π satisfies formula $\Diamond a$
- question: what does $\models \exists \Diamond a$ mean?

The need for a different temporal logic

$$\overbrace{\exists \pi' \in \mathit{Paths}(\pi[m]) : \underbrace{\exists n \geq 0 : \pi'[n] \models a}_{\Diamond a}}^{\exists \Diamond a}$$

- premise: $\pi \models \Diamond a$ means that path π satisfies formula $\Diamond a$
- question: what does $\models \exists \Diamond a$ mean?
- answer: there exists a path π' *starting in a state* (say $s = \pi'[0] = \pi[m]$), such that $\pi' \models \Diamond a$; hence, $s \models \exists \Diamond a$
- consequence: we should distinguish between *path formulae* and *state formulae* (unlike LTL, which only deals with path formulae); consider formulae over *states*

The need for a different temporal logic

$$\overbrace{\forall \pi \in \mathit{Paths}(TS) : \underbrace{\forall m \geq 0 : \overbrace{\exists \pi' \in \mathit{Paths}(\pi[m]) : \underbrace{\exists n \geq 0 : \pi'[n] \models a}_{\Diamond a}}^{\exists \Diamond a}}_{\Box \exists \Diamond a}}^{\forall \Box \exists \Diamond a}$$

Computational tree logic (CTL)

- Computational tree logic has been introduced in

1. Edmund M. Clarke and Allen E. Emerson. Design and synthesis of synchronization skeletons using branching time temporal logic. In, D. Kozen, editor, *Proceedings of Workshop on Logic of Programs*, volume 131 of *Lecture Notes in Computer Science*, pages 52–71. Yorktown Heights, NY, USA, May 1981. Springer-Verlag.
2. Jean-Pierre Queille and Joseph Sifakis. Specification and verification of concurrent systems in CESAR. In, M. Dezani-Ciancaglini and U. Montanari, editors, *Proceedings of the 5th International Symposium on Programming*, volume 137 of *Lecture Notes in Computer Science*, pages 337–351. Torino, Italy, April 1982. Springer-Verlag.

Syntax of CTL

- *state formulae* are defined by

$$\Phi ::= \mathrm{true} \mid a \mid \Phi \wedge \Phi \mid \neg\Phi \mid \exists\varphi \mid \forall\varphi$$

- *path formulae* are defined by

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

Syntax of CTL

- *state formulae* are defined by

$$\Phi ::= \text{true} \mid a \mid \Phi \wedge \Phi \mid \neg\Phi \mid \exists\varphi \mid \forall\varphi$$

- *path formulae* are defined by

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

- two temporal modalities are introduced as

$$\begin{aligned} \exists\Diamond\Phi &= \exists(\text{true}\,\mathsf{U}\,\Phi) \\ \forall\Diamond\Phi &= \forall(\text{true}\,\mathsf{U}\,\Phi) \\ \exists\Box\Phi &= \neg\forall(\text{true}\,\mathsf{U}\,\neg\Phi) \\ \forall\Box\Phi &= \neg\exists(\text{true}\,\mathsf{U}\,\neg\Phi) \end{aligned}$$

CTL properties for the traffic light model

- question: how to express
  "the light is infinitely often green"
  in CTL?

CTL properties for the traffic light model

- question: how to express

  "the light is infinitely often green"

  in CTL?

- answer: $\forall\Box\forall\Diamond$green

CTL properties for the traffic light model

- question: how to express
  "each red light is preceded by an amber light"
  in CTL?

CTL properties for the traffic light model

- question: how to express
  "each red light is preceded by an amber light"
  in CTL?

- answer: $\neg \text{red} \wedge \forall\Box(\exists\bigcirc\neg\text{red} \vee \text{amber})$

  this can be derived via implication rule, from LTL case:
  $\neg\text{red} \wedge \forall\Box((\forall\bigcirc\text{red}) \Rightarrow \text{amber})$

Semantics of CTL

$$TS \models \Phi \text{ iff } \forall s \in I : s \models \Phi$$

where (cf. CTL syntax)

$$\begin{array}{rcl} s \models \text{true} & & \\ s \models a & \text{iff} & a \in L(s) \\ s \models \Phi \wedge \Psi & \text{iff} & s \models \Phi \wedge s \models \Psi \\ s \models \neg\Phi & \text{iff} & s \not\models \Phi \\ s \models \exists\varphi & \text{iff} & \exists\pi \in Paths(s) : \pi \models \varphi \\ s \models \forall\varphi & \text{iff} & \forall\pi \in Paths(s) : \pi \models \varphi \end{array}$$

and where

$$\begin{array}{rcl} \pi \models \bigcirc\Phi & \text{iff} & \pi[1] \models \Phi \\ \pi \models \Phi \,\mathsf{U}\, \Psi & \text{iff} & \exists i \geq 0 : \pi[i] \models \Psi \wedge \forall 0 \leq j < i : \pi[j] \models \Phi \end{array}$$

Semantics of CTL

$$TS \models \Phi \text{ iff } \forall s \in I : s \models \Phi$$

where (cf. CTL syntax)

$$\begin{array}{rcl}
s \models \text{true} & & \\
s \models a & \text{iff} & a \in L(s) \\
s \models \Phi \wedge \Psi & \text{iff} & s \models \Phi \wedge s \models \Psi \\
s \models \neg\Phi & \text{iff} & s \not\models \Phi \\
s \models \exists\varphi & \text{iff} & \exists\pi \in Paths(s) : \pi \models \varphi \\
s \models \forall\varphi & \text{iff} & \forall\pi \in Paths(s) : \pi \models \varphi
\end{array}$$

and where

$$\begin{array}{rcl}
\pi \models \bigcirc\Phi & \text{iff} & \pi[1] \models \Phi \\
\pi \models \Phi \,\mathsf{U}\, \Psi & \text{iff} & \exists i \geq 0 : \pi[i] \models \Psi \wedge \forall 0 \leq j < i : \pi[j] \models \Phi
\end{array}$$

The *satisfaction set* $Sat(\Phi)$ is defined by

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

On the semantics of CTL temporal modalities

- recall that

$$\exists\Diamond\Phi = \exists(\text{true}\ \mathsf{U}\ \Phi)$$

- question: how is

$$s \models \exists\Diamond\Phi$$

  defined?

On the semantics of CTL temporal modalities

- recall that

$$\exists\Diamond\Phi = \exists(\mathsf{true}\ \mathsf{U}\ \Phi)$$

- question: how is

$$s \models \exists\Diamond\Phi$$

defined?

- answer:

$$\exists\pi \in Paths(s) : \exists i \geq 0 : \pi[i] \models \Phi$$

On the semantics of CTL temporal modalities

- recall that
$$\forall\Diamond\Phi = \forall(\mathsf{true}\ \mathsf{U}\ \Phi)$$
- question: how is
$$s \models \forall\Diamond\Phi$$
defined?

On the semantics of CTL temporal modalities

- recall that

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

- question: how is

$$s \models \forall\Diamond\Phi$$

  defined?

- answer:

$$\forall\pi \in \mathit{Paths}(s) : \exists i \geq 0 : \pi[i] \models \Phi$$

  (here index i can depend on specific path π)

On the semantics of CTL temporal modalities

- recall that
$$\exists\Box\Phi = \neg\forall(\text{true}\ \mathsf{U}\ \neg\Phi)$$

- question: how is
$$s \models \exists\Box\Phi$$
defined?

On the semantics of CTL temporal modalities

- recall that

$$\exists \Box \Phi = \neg \forall (\text{true} \, \mathsf{U} \, \neg \Phi)$$

- question: how is

$$s \models \exists \Box \Phi$$

  defined?

- answer:

$$\exists \pi \in \mathit{Paths}(s) : \forall i \geq 0 : \pi[i] \models \Phi$$

On the semantics of CTL temporal modalities

- recall that
$$\forall\Box\Phi = \neg\exists(\mathsf{true}\;\mathsf{U}\;\neg\Phi)$$
- question: how is
$$s \models \forall\Box\Phi$$
defined?

On the semantics of CTL temporal modalities

- recall that

$$\forall\Box\Phi = \neg\exists(\mathsf{true}\,\mathsf{U}\,\neg\Phi)$$

- question: how is

$$s \models \forall\Box\Phi$$

  defined?

- answer:

$$\forall\pi \in \mathit{Paths}(s) : \forall i \geq 0 : \pi[i] \models \Phi$$

Side note: CTL syntax and semantics of negation

- recall that in LTL $\Box\Phi = \neg\Diamond\neg\Phi$
  however in CTL $\exists\Box\Phi \neq \exists\neg\Diamond\neg\Phi$ (why?),
  instead $\exists\Box\Phi = \neg\forall\Diamond\neg\Phi$

Side note: CTL syntax and semantics of negation

- recall that in LTL $\Box\Phi = \neg\Diamond\neg\Phi$
  however in CTL $\exists\Box\Phi \neq \exists\neg\Diamond\neg\Phi$ (why?),
  instead $\exists\Box\Phi = \neg\forall\Diamond\neg\Phi$

- we have that $s \not\models \Psi \Leftrightarrow s \models \neg\Psi$
- however, for a given TS, we can have both $TS \not\models \Psi$ and $TS \not\models \neg\Psi$

Side note: CTL syntax and semantics of negation

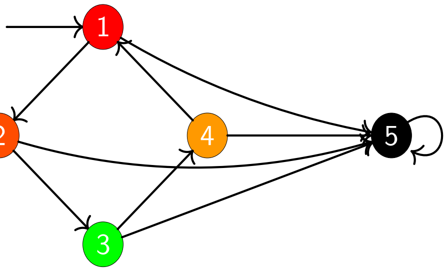
- recall that in LTL $\Box\Phi = \neg\Diamond\neg\Phi$
  however in CTL $\exists\Box\Phi \neq \exists\neg\Diamond\neg\Phi$ (why?),
  instead $\exists\Box\Phi = \neg\forall\Diamond\neg\Phi$

- we have that $s \not\models \Psi \Leftrightarrow s \models \neg\Psi$
- however, for a given TS, we can have both $TS \not\models \Psi$ and $TS \not\models \neg\Psi$, e.g.: $\Psi = \exists\Box r$

here $1 \models \Psi$, whereas $3 \not\models \Psi$, so $TS \not\models \Psi$;
at the same time, $TS \not\models \neg\Psi$, since $1 \not\models \neg\Psi \equiv \forall\Diamond\neg r$

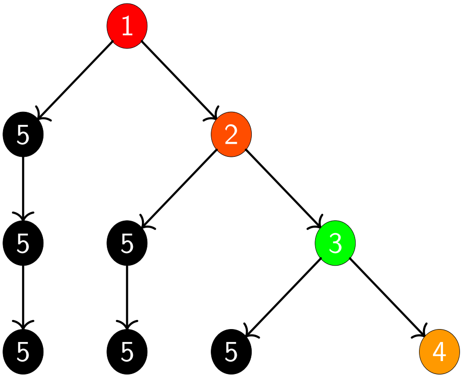
CTL – unfolding of a transition system

- the following transition system

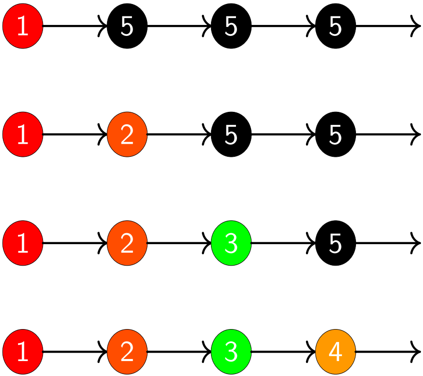


- can be unfolded via its paths as follows . . .

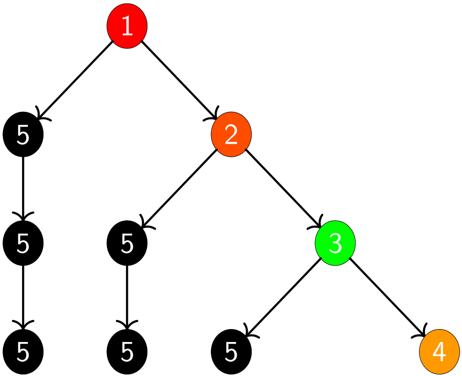
CTL – unfolding of a transition system



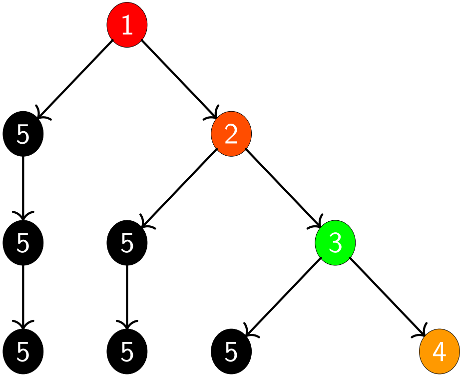
LTL is a linear temporal logic



CTL is a branching temporal logic

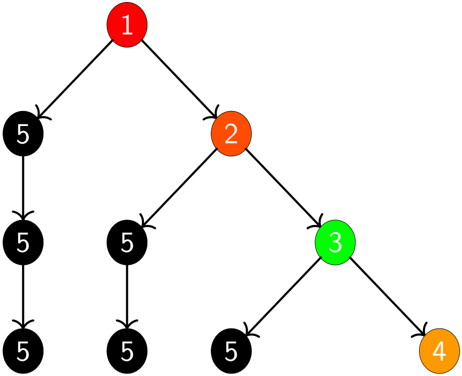


Example of CTL semantics via TS unfolding



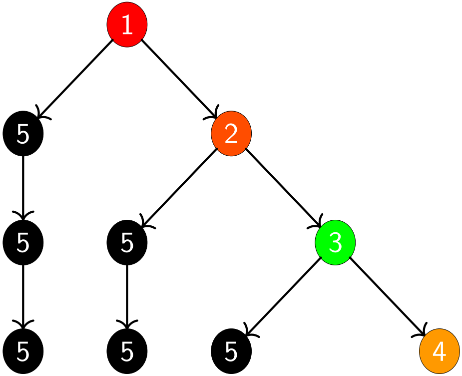
- question: $\exists\Diamond$green?
- answer:

Example of CTL semantics via TS unfolding



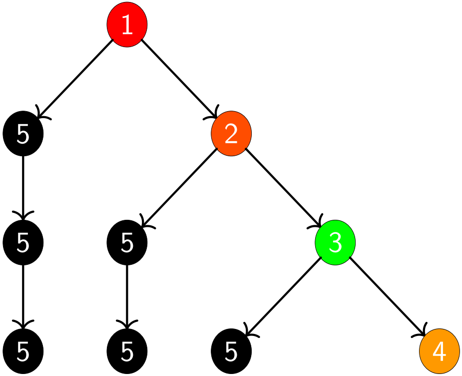
- question: $\exists\Diamond$green?
- answer: yes

Example of CTL semantics via TS unfolding



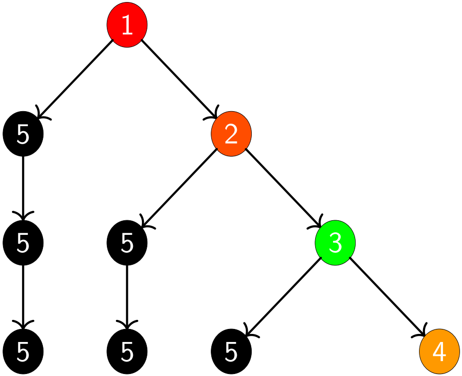
- question: $\forall\Diamond$black?
- answer:

Example of CTL semantics via TS unfolding



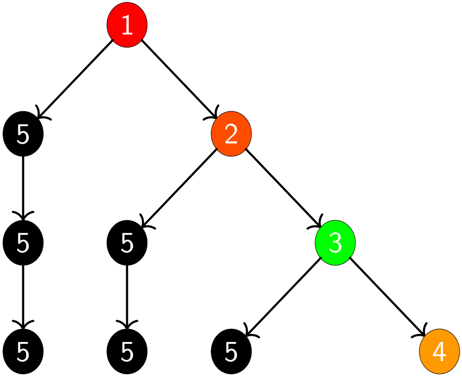
- question: $\forall\Diamond$black?
- answer: no

Example of CTL semantics via TS unfolding



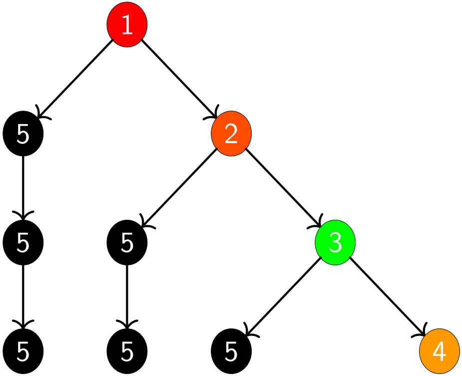
- question: $\exists\Box\neg$black?
- answer:

Example of CTL semantics via TS unfolding



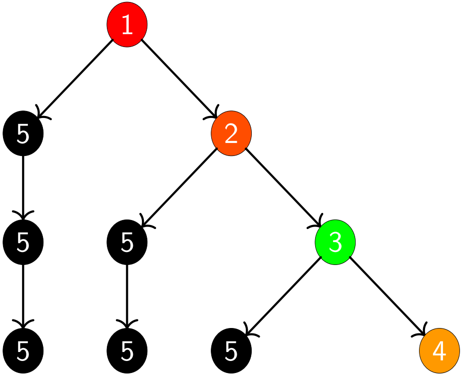
- question: $\exists\Box\neg$black?
- answer: yes

Example of CTL semantics via TS unfolding



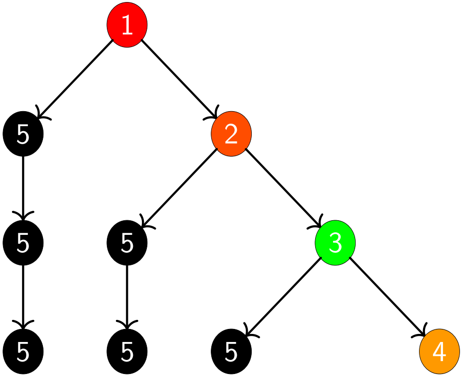
- question: $\forall\Box$black?
- answer:

Example of CTL semantics via TS unfolding



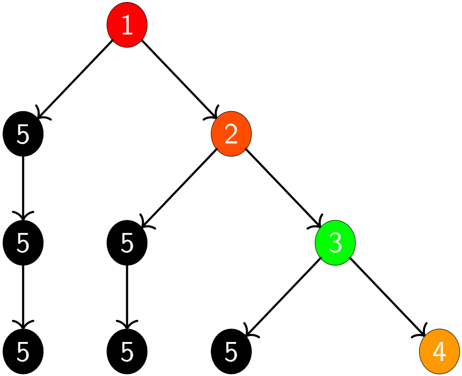
- question: $\forall\Box$black?
- answer: no

Example of CTL semantics via TS unfolding



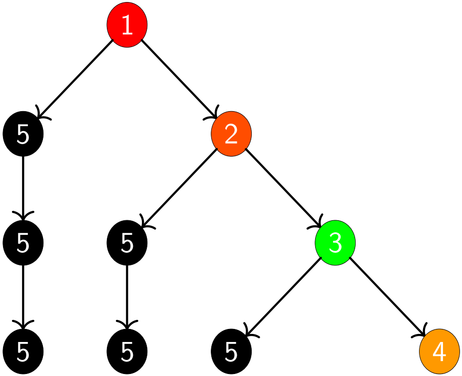
- question: $\exists((\neg \text{black}) \text{ U black})$?
- answer:

Example of CTL semantics via TS unfolding



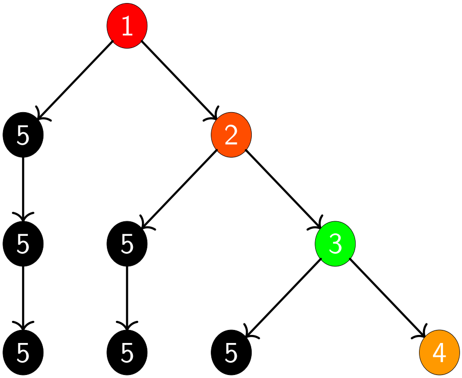
- question: $\exists((\neg\text{black})\ \mathsf{U}\ \text{black})$?
- answer: yes

Example of CTL semantics via TS unfolding



- question: $\neg\forall((\neg\text{black})\ \mathsf{U}\ \text{black})$?

Example of CTL semantics via TS unfolding



- question: $\neg\forall((\neg \text{black}) \cup \text{black})$?
- answer: yes

More on CTL

Definition

CTL formulae Φ and Ψ are *equivalent*, denoted $\Phi \equiv \Psi$, if $Sat_{TS}(\Phi) = Sat_{TS}(\Psi)$ for all transition systems TS.

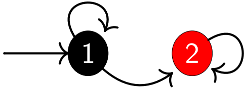
- as for LTL, CTL admits syntactic expression via normal forms
  - PNF
  - Existential normal form (ENF)
- analogous distributive and expansion laws

$\forall\Box\exists\Diamond a$ cannot be expressed in LTL

- consider formula $\forall\Box\exists\Diamond a$; for visual simplicity we pick $a = \textit{red}$

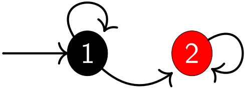
$\forall\Box\exists\Diamond a$ cannot be expressed in LTL

- consider formula $\forall\Box\exists\Diamond a$; for visual simplicity we pick $a = red$
- assume there exists LTL formula ϕ equivalent to $\forall\Box\exists\Diamond a$
- build TS: $TS \models \forall\Box\exists\Diamond a$; since $\phi \equiv \forall\Box\exists\Diamond a$, $TS \models \phi$, namely $Traces(TS) \subseteq Words(\phi)$

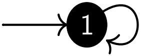


$\forall\Box\exists\Diamond a$ cannot be expressed in LTL

- consider formula $\forall\Box\exists\Diamond a$; for visual simplicity we pick $a = \textit{red}$
- assume there exists LTL formula ϕ equivalent to $\forall\Box\exists\Diamond a$
- build TS: $TS \models \forall\Box\exists\Diamond a$; since $\phi \equiv \forall\Box\exists\Diamond a$, $TS \models \phi$, namely $\mathit{Traces}(TS) \subseteq \mathit{Words}(\phi)$

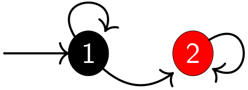


- build TS' : $\mathit{Traces}(TS') \subseteq \mathit{Traces}(TS)$; since $1^\omega \in \mathit{Paths}(TS)$, TS' :

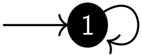


$\forall\Box\exists\Diamond a$ cannot be expressed in LTL

- consider formula $\forall\Box\exists\Diamond a$; for visual simplicity we pick $a = red$
- assume there exists LTL formula ϕ equivalent to $\forall\Box\exists\Diamond a$
- build TS: $TS \models \forall\Box\exists\Diamond a$; since $\phi \equiv \forall\Box\exists\Diamond a$, $TS \models \phi$, namely $Traces(TS) \subseteq Words(\phi)$



- build TS' : $Traces(TS') \subseteq Traces(TS)$;
  since $1^\omega \in Paths(TS)$, TS' :



- $Traces(TS') \subseteq Traces(TS) \wedge TS \models \phi \Rightarrow TS' \models \phi$, which contradicts the fact that $TS' \not\models \forall\Box\exists\Diamond red$ (since $1 \not\models \forall\Box\exists\Diamond red$)

Expressiveness of LTL and CTL

- we have just established the following result

Theorem

The property

$$\forall \pi \in Paths(TS) : \forall m \geq 0 : \exists \pi' \in Paths(\pi[m]) : \exists n \geq 0 : \pi'[n] \models a$$

cannot be captured by LTL, but is captured by the CTL formula $\forall \Box \exists \Diamond a$.

- so CTL might seem to be more expressive than LTL?

Expressiveness of LTL and CTL

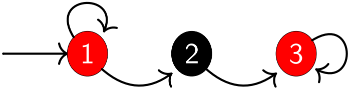
- at the same time, it can be shown that the CTL formula $\forall\Box\forall\Diamond a$ is equivalently expressed by the LTL formula $\Box\Diamond a$

Expressiveness of LTL and CTL

- at the same time, it can be shown that the CTL formula $\forall\Box\forall\Diamond a$ is equivalently expressed by the LTL formula $\Box\Diamond a$
- so perhaps: can universally quantified CTL formulae directly lead to LTL ones by dropping the (path) quantifications?
- answer: yes, *whenever* such an LTL formula exists (i.e., either the two formulae are equivalent, or there exists no equivalent LTL formula)

$\Diamond\Box red$ is not equivalent to $\forall\Diamond\forall\Box red$

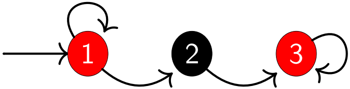
- on the flip side, consider now the following
- build TS, such that $TS \models \Diamond\Box red$:



- notice that $Paths(TS) = \{1^\omega, 1^+23^\omega\}$

$\Diamond\Box red$ is not equivalent to $\forall\Diamond\forall\Box red$

- on the flip side, consider now the following
- build TS, such that $TS \models \Diamond\Box red$:



- notice that $Paths(TS) = \{1^\omega, 1^+23^\omega\}$

- consider now CTL formula $\forall\Diamond\forall\Box red$, where $\Psi = \forall\Box red$
- notice that $1 \not\models \Psi$, because of the potential path 1^+23^ω
- this means that $1^\omega \not\models \Diamond\Psi$, hence $1 \not\models \forall\Diamond\Psi$
- so the two formulae are not equivalent

Expressiveness of LTL and CTL

- we can further establish the following result

Theorem

The property

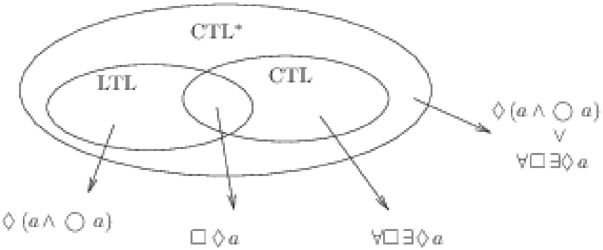
$$\forall \pi \in Paths(TS) : \exists i \geq 0 : \forall j \geq i : \pi[j] \models a$$

which is captured by the LTL formula $\Diamond\Box a$, cannot be captured by any CTL formula.

- in conclusion, it looks like LTL and CTL are incomparable

→ topology of the relationship btw LTL and CTL

Expressiveness of LTL, CTL and CTL*



Extended computation tree logic

- this has led to a generalised logic, known as extended computation tree logic (CTL*)
- introduced by:
  - E.A. Emerson, J,Y. Halpern,
    "Sometimes" and "not never" revisited:
    on branching versus linear time temporal logic.
    J. ACM 33, 1, pp. 151–178, Jan. 1986.

Syntax of CTL*

- *state formulae* are defined by

$$\Phi ::= \mathsf{true} \mid a \mid \Phi \wedge \Phi \mid \neg\Phi \mid \exists\varphi$$

- *path formulae* are defined by

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

Syntax of CTL*

- *state formulae* are defined by

$$\Phi ::= \text{true} \mid a \mid \Phi \wedge \Phi \mid \neg\Phi \mid \exists\varphi$$

- *path formulae* are defined by

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

- $\forall\varphi = \neg\exists\neg\varphi$
- $\Diamond\varphi = \mathit{true} \cup \varphi$
- $\Box\varphi = \neg\Diamond\neg\varphi$

Semantics of CTL*

$$
\begin{array}{rcl}
s \models \text{true} & & \\
s \models a & \text{iff} & a \in L(s) \\
s \models \Phi \wedge \Psi & \text{iff} & s \models \Phi \wedge s \models \Psi \\
s \models \neg\Phi & \text{iff} & s \not\models \Phi \\
s \models \exists\varphi & \text{iff} & \exists\pi \in \mathit{Paths}(s) : \pi \models \varphi
\end{array}
$$

and

$$
\begin{array}{rcl}
\pi \models \Phi & \text{iff} & \pi[0] \models \Phi \\
\pi \models \varphi \wedge \psi & \text{iff} & \pi \models \varphi \wedge \pi \models \psi \\
\pi \models \neg\varphi & \text{iff} & \pi \not\models \varphi \\
\pi \models \bigcirc\varphi & \text{iff} & \pi[1..] \models \varphi \\
\pi \models \varphi \,\mathsf{U}\, \psi & \text{iff} & \exists i \geq 0 : \pi[i..] \models \psi \wedge \forall 0 \leq j < i : \pi[j..] \models \varphi
\end{array}
$$

The satisfaction set is defined as in CTL

Formulae equivalence in CTL*

Definition

CTL* formulae Φ and Ψ are *equivalent*, denoted $\Phi \equiv \Psi$, if $Sat_{TS}(\Phi) = Sat_{TS}(\Psi)$ for all transition systems TS.

Formulae equivalence in CTL*

Definition

CTL* formulae Φ and Ψ are *equivalent*, denoted $\Phi \equiv \Psi$, if $Sat_{TS}(\Phi) = Sat_{TS}(\Psi)$ for all transition systems TS.

- question: $\forall(\bigcirc p \vee \bigcirc\bigcirc p) \equiv (\forall\bigcirc p) \vee (\forall\bigcirc\bigcirc p)$?

Formulae equivalence in CTL*

Definition

CTL* formulae Φ and Ψ are *equivalent*, denoted $\Phi \equiv \Psi$, if $Sat_{TS}(\Phi) = Sat_{TS}(\Psi)$ for all transition systems TS.

- question: $\forall(\bigcirc p \vee \bigcirc\bigcirc p) \equiv (\forall\bigcirc p) \vee (\forall\bigcirc\bigcirc p)$?
- answer: no
- question: how can you prove your answer?

Formulae equivalence in CTL*

Definition

CTL* formulae Φ and Ψ are *equivalent*, denoted $\Phi \equiv \Psi$, if $Sat_{TS}(\Phi) = Sat_{TS}(\Psi)$ for all transition systems TS.

- question: $\forall(\bigcirc p \vee \bigcirc\bigcirc p) \equiv \forall\bigcirc(p \vee \forall\bigcirc p)$?

Formulae equivalence in CTL*

Definition

CTL* formulae Φ and Ψ are *equivalent*, denoted $\Phi \equiv \Psi$, if $Sat_{TS}(\Phi) = Sat_{TS}(\Psi)$ for all transition systems TS.

- question: $\forall(\bigcirc p \vee \bigcirc\bigcirc p) \equiv \forall\bigcirc(p \vee \forall\bigcirc p)$?
- answer: yes
- question: how do you prove your answer?

Expressiveness of LTL and CTL*

Theorem

For each LTL formula φ and state s,

$$\underbrace{s \models \varphi}_{\text{LTL semantics}} \quad \textit{iff} \quad \underbrace{s \models \forall \varphi}_{\text{CTL}^* \text{ semantics}}$$

- LTL can be thought of as a fragment of CTL*

Expressiveness of LTL, CTL and CTL*

Theorem
The CTL property*

$$\underbrace{(\Diamond\Box a)}_{\textit{cannot be expressed in CTL}} \quad \vee \quad \underbrace{(\forall\Box\exists\Diamond b)}_{\textit{cannot be expressed in LTL}}$$

cannot be captured in either LTL or CTL.

Today's reading material

- Section 6.1–6.2.3, Theorem 6.21, Section 6.8.1 of
  - Christel Baier and Joost-Pieter Katoen. *Principles of Model Checking*. The MIT Press. Cambridge, MA, USA. 2008.
- M.Y. Vardi, *Branching vs. Linear Time: Final Showdown*. Tools and Algorithms for the Construction and Analysis of Systems, LNCS vol. 2031, pp 1-22, 2001.