4507/6507 Software and Hardware Verification
Computation Tree Logic (CTL)

Andrei Popescu

University of Sheffield

These slides contain material from Denisa Diaconescu and Traian Florin Șerbănuță

Introduction

CTL

CTL = Computation Tree Logic

Introduced by Edmund M. Clarke and E. Allen Emerson in 1981

A temporal logic for reasoning about transition systems

An alternative to LTL

Syntax

CTL Syntax – Definition

CTL Syntax – Definition

Assume some set *Atoms* of atomic propositions (atoms), ranged over by a, b, c.

CTL Syntax – Definition

Assume some set *Atoms* of atomic propositions (atoms), ranged over by a, b, c.

CTL formulas, usually denoted φ, ψ, etc., are defined by:

$$\varphi ::= a$$

CTL Syntax – Definition

Assume some set *Atoms* of atomic propositions (atoms), ranged over by a, b, c.

CTL formulas, usually denoted φ, ψ, etc., are defined by:

$$\varphi ::= a \mid \neg\varphi \mid \varphi \vee \psi \mid \varphi \wedge \psi \mid \varphi \rightarrow \psi \mid$$

CTL Syntax – Definition

Assume some set *Atoms* of atomic propositions (atoms), ranged over by a, b, c.

CTL formulas, usually denoted φ, ψ, etc., are defined by:

$$\varphi \ ::= \ a \mid \neg\varphi \mid \varphi \vee \psi \mid \varphi \wedge \psi \mid \varphi \to \psi \mid$$
$$\forall\bigcirc\varphi \mid \exists\bigcirc\varphi \mid \forall\Diamond\varphi \mid \exists\Diamond\varphi \mid \forall\Box\varphi \mid \exists\Box\varphi \mid \varphi\,\forall\mathsf{U}\,\psi \mid \varphi\,\exists\mathsf{U}\,\psi$$

CTL Syntax – Definition

Assume some set *Atoms* of atomic propositions (atoms), ranged over by a, b, c.

CTL formulas, usually denoted φ, ψ, etc., are defined by:

$$\varphi \ ::= \ a \mid \neg\varphi \mid \varphi \vee \psi \mid \varphi \wedge \psi \mid \varphi \rightarrow \psi \mid$$
$$\forall\bigcirc\varphi \mid \exists\bigcirc\varphi \mid \forall\Diamond\varphi \mid \exists\Diamond\varphi \mid \forall\Box\varphi \mid \exists\Box\varphi \mid \varphi\,\forall\mathsf{U}\,\psi \mid \varphi\,\exists\mathsf{U}\,\psi$$

Each temporal connective is a pair of a path quantifier:

- $\forall$ – for all paths
- $\exists$ – there exists a path

and an LTL-like temporal operator $\bigcirc$, $\Diamond$, $\Box$, U.

CTL Syntax – Definition

Assume some set *Atoms* of atomic propositions (atoms), ranged over by a, b, c.

CTL formulas, usually denoted φ, ψ, etc., are defined by:

$$\varphi \ ::= \ a \mid \neg\varphi \mid \varphi \vee \psi \mid \varphi \wedge \psi \mid \varphi \rightarrow \psi \mid$$
$$\forall\bigcirc\varphi \mid \exists\bigcirc\varphi \mid \forall\Diamond\varphi \mid \exists\Diamond\varphi \mid \forall\Box\varphi \mid \exists\Box\varphi \mid \varphi\forall\mathsf{U}\,\psi \mid \varphi\exists\mathsf{U}\,\psi$$

Each temporal connective is a pair of a path quantifier:

- $\forall$ – for all paths
- $\exists$ – there exists a path

and an LTL-like temporal operator $\bigcirc$, $\Diamond$, $\Box$, U.

Precedence: As usual, unary connectives bind more strongly than binary ones.

CTL Syntax – Examples

The following are CTL formulas:

$$\forall\Box\,(b \rightarrow \exists\Box\, c)$$

$$(\exists\Diamond\, a)\, \exists\mathsf{U}\, b$$

$$a\, \forall\mathsf{U}\, (\exists\Diamond\, b)$$

$$(\exists\Diamond\, \exists\Box\, a) \rightarrow (\forall\Diamond\, b)$$

CTL Syntax – Examples

The following are CTL formulas:

$$\forall\Box\,(b \to \exists\Box\, c)$$

$$(\exists\Diamond\, a)\,\exists\mathsf{U}\, b$$

$$a\,\forall\mathsf{U}\,(\exists\Diamond\, b)$$

$$(\exists\Diamond\,\exists\Box\, a) \to (\forall\Diamond\, b)$$

The following are not CTL formulas:

$$\exists\Diamond\,\Box b$$

$$\forall\neg\Box\neg a$$

$$\Diamond(a\,\mathsf{U}\, b)$$

$$\exists\Diamond\,(a\,\mathsf{U}\, b)$$

Semantics

Labeled Transition Systems and Paths Recalled

(These are the same as for LTL.)

A labeled transition system (LTS for short) is a triple $\mathcal{M} = (S, \rightarrow, L)$ consisting of:

- S a finite set of states
- $\rightarrow \subseteq S \times S$ a transition relation
- $L : S \rightarrow \mathcal{P}(Atoms)$ a labeling function

such that every state has an outward transition, i.e., for all $s_1 \in S$ there exists $s_2 \in S$ with $s_1 \rightarrow s_2$.

Labeled Transition Systems and Paths Recalled

(These are the same as for LTL.)

A labeled transition system (LTS for short) is a triple $\mathcal{M} = (S, \rightarrow, L)$ consisting of:

- S a finite set of states
- $\rightarrow \subseteq S \times S$ a transition relation
- $L : S \rightarrow \mathcal{P}(Atoms)$ a labeling function

such that every state has an outward transition, i.e., for all $s_1 \in S$ there exists $s_2 \in S$ with $s_1 \rightarrow s_2$.

A path π in an LTS $\mathcal{M} = (S, \rightarrow, L)$ is an infinite sequence of states $s_0 s_1 s_2 \ldots$ such that for all $i \geq 0$, $s_i \rightarrow s_{i+1}$.

Given $s \in S$, we write $Paths_s(\mathcal{M})$ for the set of all paths in $\mathcal{M}$ that start from s.

Towards Defining the CTL Semantics

Recall:

- LTL satisfaction is first defined on linear structures, i.e., on infinite sequences of states $\pi = s_0 s_1 \ldots$ (given $L : S \to \mathcal{P}(Atoms)$): $\pi \models_L \varphi$

Towards Defining the CTL Semantics

Recall:

- LTL satisfaction is first defined on linear structures, i.e., on infinite sequences of states $\pi = s_0 s_1 \ldots$ (given $L : S \to \mathcal{P}(Atoms)$): $\pi \models_L \varphi$
- ... and later extended to branching structures, i.e., to LTSs $\mathcal{M} = (S, \to, L)$, by quantifying universally over all paths:

$$\mathcal{M}, s \models \varphi \text{ defined as } \pi \models_L \varphi \text{ for all } \pi \in Paths_s(\mathcal{M})$$

Towards Defining the CTL Semantics

Recall:

- LTL satisfaction is first defined on linear structures, i.e., on infinite sequences of states $\pi = s_0 s_1 \ldots$ (given $L : S \to \mathcal{P}(Atoms)$): $\pi \models_L \varphi$
- ... and later extended to branching structures, i.e., to LTSs $\mathcal{M} = (S, \to, L)$, by quantifying universally over all paths:

$$\mathcal{M}, s \models \varphi \text{ defined as } \pi \models_L \varphi \text{ for all } \pi \in Paths_s(\mathcal{M})$$

By contrast, for CTL:

- satisfaction will be defined directly on the branching structures, i.e., on LTSs

Towards Defining the CTL Semantics

Recall:

- LTL satisfaction is first defined on linear structures, i.e., on infinite sequences of states $\pi = s_0 s_1 \ldots$ (given $L : S \to \mathcal{P}(\mathit{Atoms})$): $\pi \models_L \varphi$
- … and later extended to branching structures, i.e., to LTSs $\mathcal{M} = (S, \to, L)$, by quantifying universally over all paths:

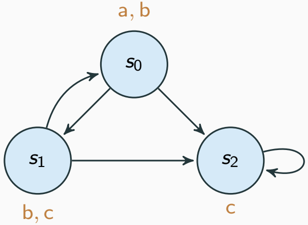
$$\mathcal{M}, s \models \varphi \text{ defined as } \pi \models_L \varphi \text{ for all } \pi \in \mathit{Paths}_s(\mathcal{M})$$

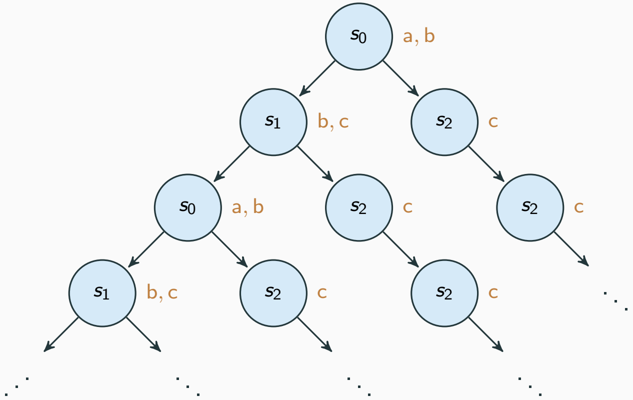
By contrast, for CTL:

- satisfaction will be defined directly on the branching structures, i.e., on LTSs
- … and it is most intuitive to think in terms of the unwinding tree, a.k.a. computation tree, of an LTS

Recall: The Unwinding (Computation) Tree of an LTS

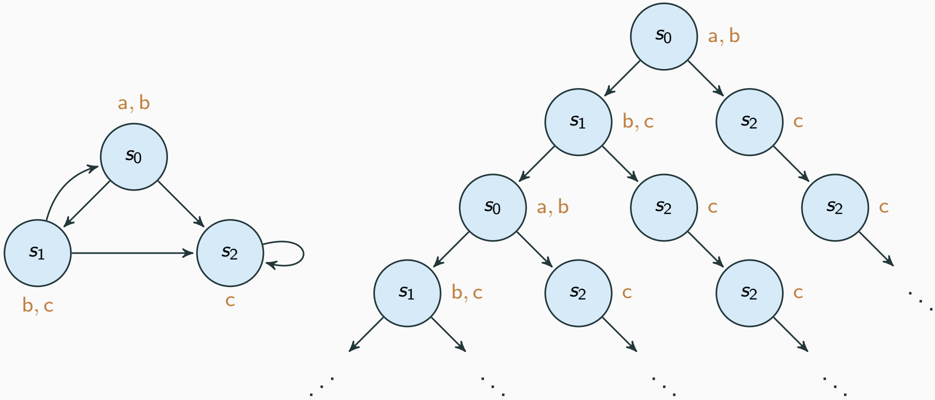
An LTS (on the left) and its unwinding tree starting in s_0 (on the right):





Recall: The Unwinding (Computation) Tree of an LTS

An LTS (on the left) and its unwinding tree starting in s_0 (on the right):



Note: The LTS and its unwinding tree have the same paths.

CTL Semantics in Pictures

In the following illustrations:

- we consider specific CTL formulas
- we draw part of a sample unwinding tree of an LTS
- using color coding when necessary, we highlight states where subformulas of the formula are supposed to hold according to the intended semantics

CTL Semantics in Pictures

In the following illustrations:

- we consider specific CTL formulas
- we draw part of a sample unwinding tree of an LTS
- using color coding when necessary, we highlight states where subformulas of the formula are supposed to hold according to the intended semantics

- by "a current or future state" we mean "the current state or a future state"
- by "all current and future states" we mean "the current state and all the future states"

CTL Semantics in Pictures

"For All Eventually"

$$\forall \Diamond \varphi$$

For all paths, φ eventually holds.

(I.e., every path has a state where φ holds.)

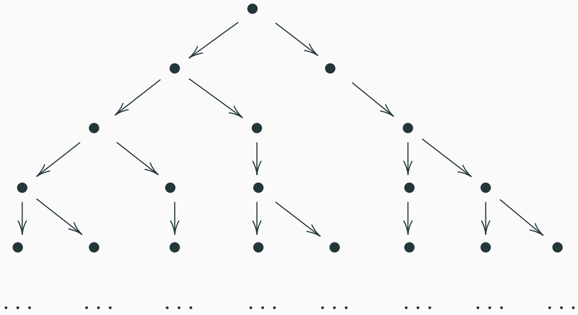
CTL Semantics in Pictures

"For All Eventually"

$\forall\Diamond\varphi$

For all paths, φ eventually holds.

(I.e., every path has a state where φ holds.)



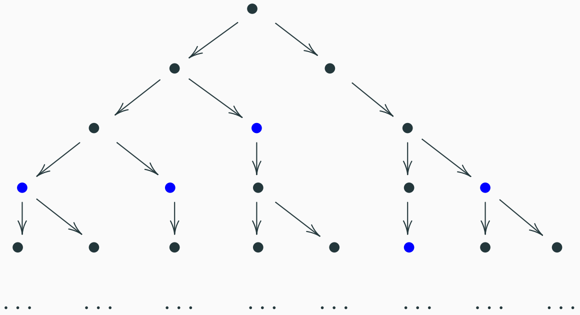
CTL Semantics in Pictures

"For All Eventually"

$\forall \Diamond \varphi$

For all paths, φ eventually holds.

(I.e., every path has a state where φ holds.)



CTL Semantics in Pictures

"There Exists Eventually"

$$\exists\Diamond\,\varphi$$

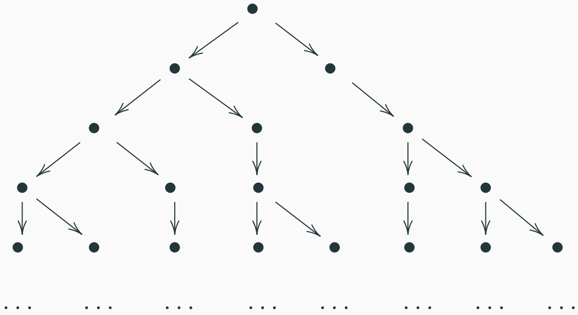
There exists a path on which φ eventually holds.
(I.e., there exists a current or future state where φ holds.)

CTL Semantics in Pictures

"There Exists Eventually"

$$\exists\Diamond\,\varphi$$

There exists a path on which φ eventually holds.
(I.e., there exists a current or future state where φ holds.)



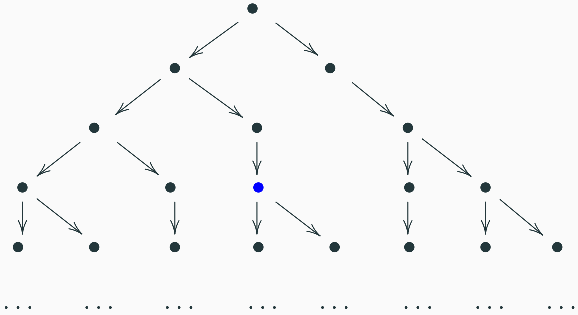
CTL Semantics in Pictures

"There Exists Eventually"

$\exists\Diamond\,\varphi$

There exists a path on which φ eventually holds.

(I.e., there exists a current or future state where φ holds.)



CTL Semantics in Pictures

"For All Next"

$\forall \bigcirc \varphi$

For all paths, φ holds next.

(I.e., on all next states, φ holds.)

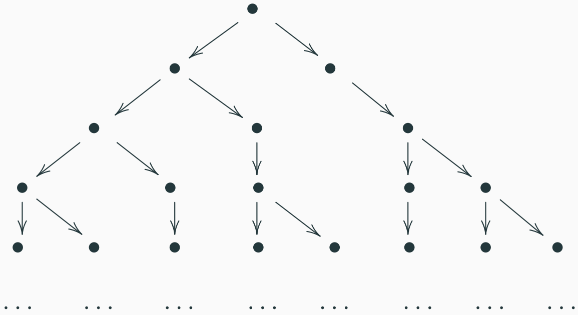
CTL Semantics in Pictures

"For All Next"

$\forall \bigcirc \varphi$

For all paths, φ holds next.

(I.e., on all next states, φ holds.)



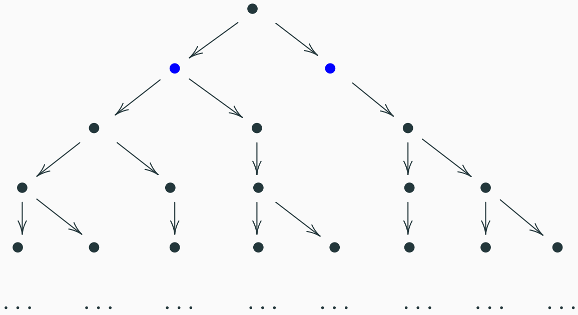
CTL Semantics in Pictures

"For All Next"

$\forall \bigcirc \varphi$

For all paths, φ holds next.

(I.e., on all next states, φ holds.)



CTL Semantics in Pictures

"There Exists Next"

$$\exists\bigcirc\varphi$$

There exists a path on which φ holds next.

(I.e., on some next state, φ holds.)

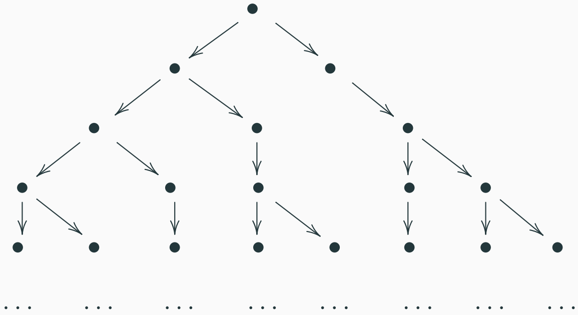
CTL Semantics in Pictures

"There Exists Next"

$\exists \bigcirc \varphi$

There exists a path on which φ holds next.

(I.e., on some next state, φ holds.)

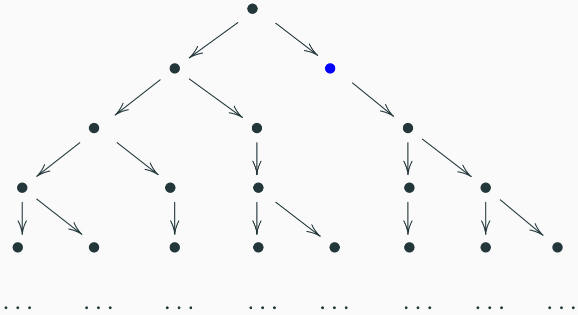


CTL Semantics in Pictures

"There Exists Next"

$\exists \bigcirc \varphi$

There exists a path on which φ holds next.
(I.e., on some next state, φ holds.)



CTL Semantics in Pictures

"For All Always"

$$\forall \Box \varphi$$

For all paths, φ always holds.

(I.e., φ holds on all current and future states.)

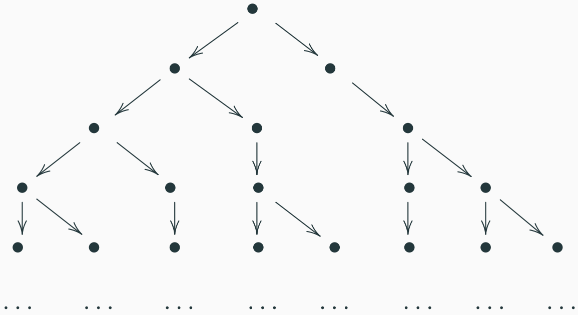
CTL Semantics in Pictures

"For All Always"

$\forall\Box\varphi$

For all paths, φ always holds.

(I.e., φ holds on all current and future states.)



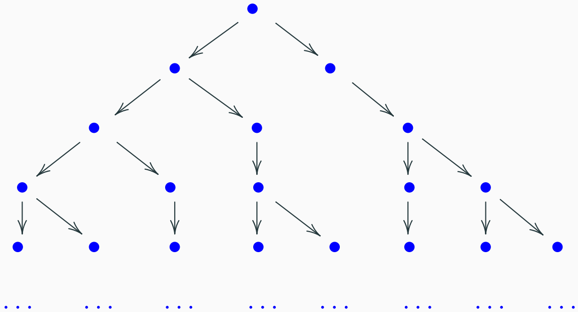
CTL Semantics in Pictures

"For All Always"

$\forall \Box \varphi$

For all paths, φ always holds.

(I.e., φ holds on all current and future states.)



CTL Semantics in Pictures

"There Exists Always"

$$\exists\Box\varphi$$

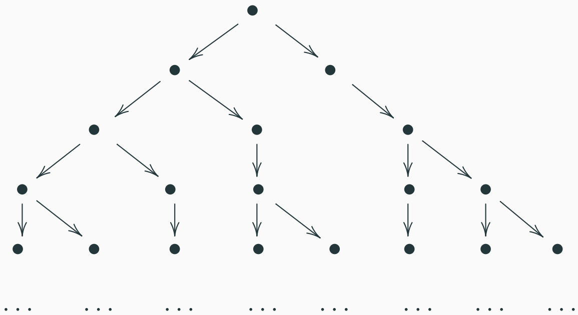
There exists a path on which φ always holds.

CTL Semantics in Pictures

"There Exists Always"

$\exists\Box\varphi$

There exists a path on which φ always holds.

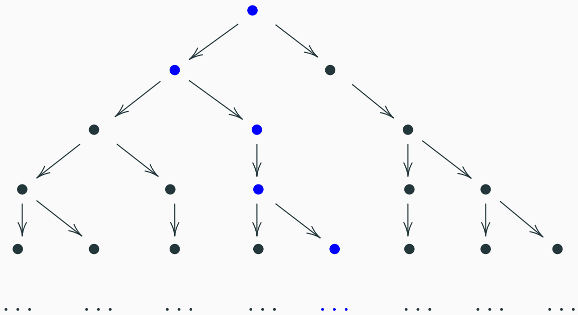


CTL Semantics in Pictures

"There Exists Always"

$\exists\Box\varphi$

There exists a path on which φ always holds.



CTL Semantics in Pictures

"For All Until"

$\varphi \, \forall\mathsf{U} \, \psi$

For all paths, φ until ψ holds.

(I.e., for all paths, ψ eventually holds and φ holds in the meantime.)

(I.e., every path starts with a (possibly empty) sequence of states where φ holds, followed by a state where ψ holds.)

CTL Semantics in Pictures

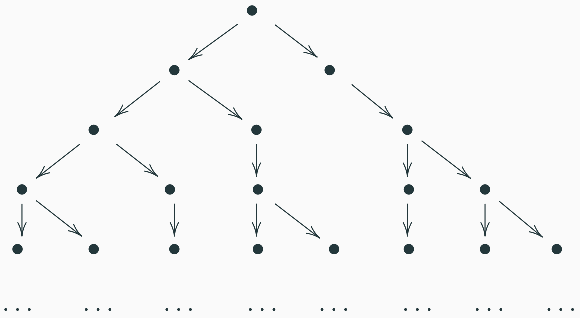
"For All Until"

$\varphi \, \forall\mathsf{U} \, \psi$

For all paths, φ until ψ holds.

(I.e., for all paths, ψ eventually holds and φ holds in the meantime.)

(I.e., every path starts with a (possibly empty) sequence of states where φ holds, followed by a state where ψ holds.)



CTL Semantics in Pictures

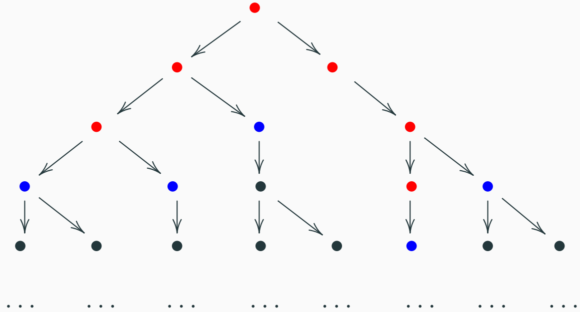
"For All Until"

$\varphi \, \forall \mathsf{U} \, \psi$

For all paths, φ until ψ holds.

(I.e., for all paths, ψ eventually holds and φ holds in the meantime.)

(I.e., every path starts with a (possibly empty) sequence of states where φ holds, followed by a state where ψ holds.)



CTL Semantics in Pictures

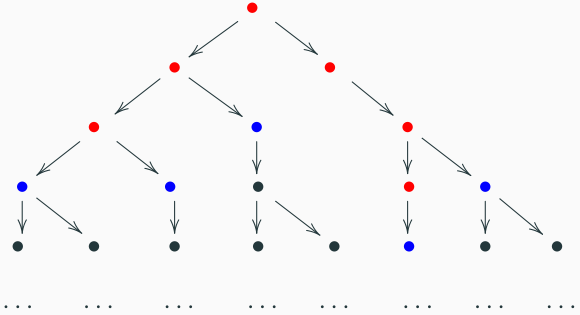
"For All Until"

$\varphi \, \forall\mathsf{U} \, \psi$

For all paths, φ until ψ holds.

(I.e., for all paths, ψ eventually holds and φ holds in the meantime.)

(I.e., every path starts with a (possibly empty) sequence of states where φ holds, followed by a state where ψ holds.)



Note: It is not forbidden that ψ also holds in a state where φ holds.

CTL Semantics in Pictures

"There Exists Until"

$\varphi\ \exists\mathsf{U}\ \psi$

For some path, φ until ψ holds.

(I.e., for some path, ψ eventually holds and φ holds in the meantime.)

(I.e., there exists a path that starts with a (possibly empty) sequence of states where φ holds, followed by a state where ψ holds.)

CTL Semantics in Pictures

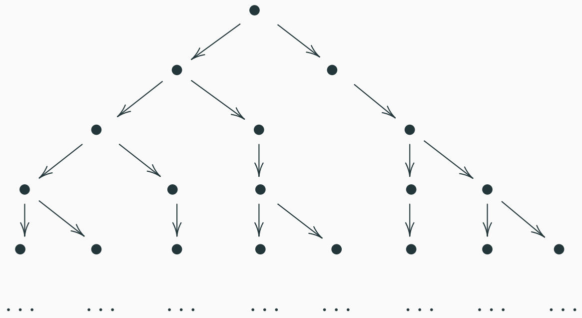
"There Exists Until"

$\varphi \, \exists \mathsf{U} \, \psi$

For some path, φ until ψ holds.

(I.e., for some path, ψ eventually holds and φ holds in the meantime.)

(I.e., there exists a path that starts with a (possibly empty) sequence of states where φ holds, followed by a state where ψ holds.)



CTL Semantics in Pictures

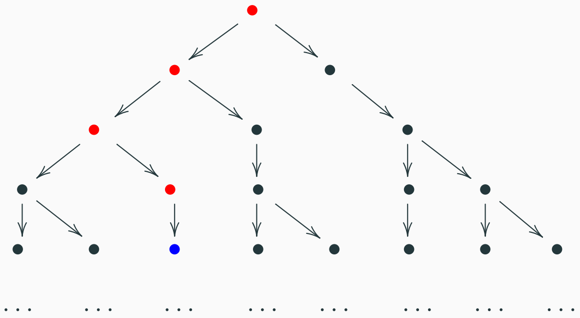
"There Exists Until"

$\varphi \, \exists \mathsf{U} \, \psi$

For some path, φ until ψ holds.

(I.e., for some path, ψ eventually holds and φ holds in the meantime.)

(I.e., there exists a path that starts with a (possibly empty) sequence of states where φ holds, followed by a state where ψ holds.)



CTL Semantics in Pictures

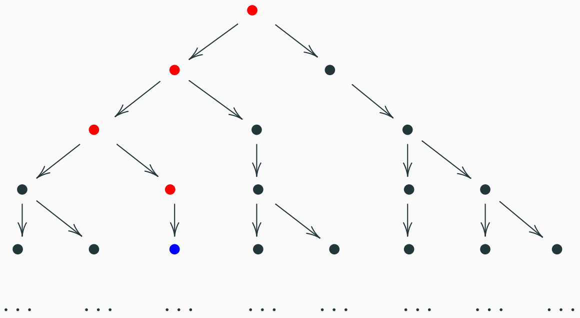
"There Exists Until"

$\varphi \, \exists \mathsf{U} \, \psi$

For some path, φ until ψ holds.

(I.e., for some path, ψ eventually holds and φ holds in the meantime.)

(I.e., there exists a path that starts with a (possibly empty) sequence of states where φ holds, followed by a state where ψ holds.)



Note: Again, it is not forbidden that ψ also holds in a state where φ holds.

CTL Semantics in Pictures – Combining Connectives

In what follows:

- "further up in all possible futures" will mean
  "on the current state and on all future states from there"
- "further up in a possible future" will mean
  "on the current state or on some future state form there"

CTL Semantics in Pictures – Combining Connectives

"For All Always" followed by "There Exists Eventually"

$\forall\Box\exists\Diamond\varphi$

for all current and future states,

there exists a state further up in a possible future from there where φ holds

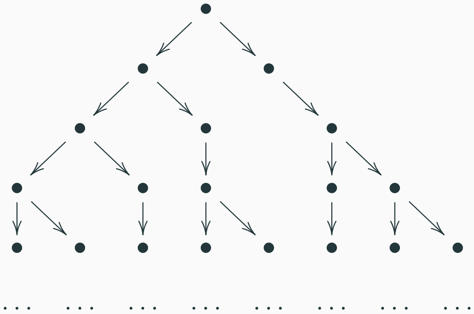
CTL Semantics in Pictures – Combining Connectives

"For All Always" followed by "There Exists Eventually"

$\forall\Box\exists\Diamond\varphi$

for all current and future states,

there exists a state further up in a possible future from there where φ holds



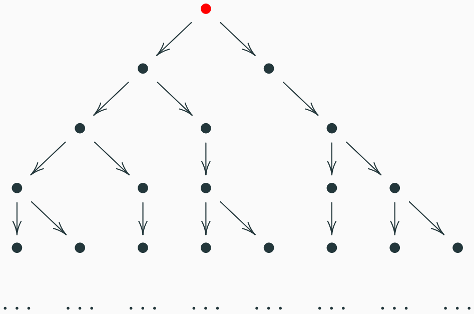
CTL Semantics in Pictures – Combining Connectives

"For All Always" followed by "There Exists Eventually"

$\forall\Box\exists\Diamond\varphi$

for all current and future states,

there exists a state further up in a possible future from there where φ holds



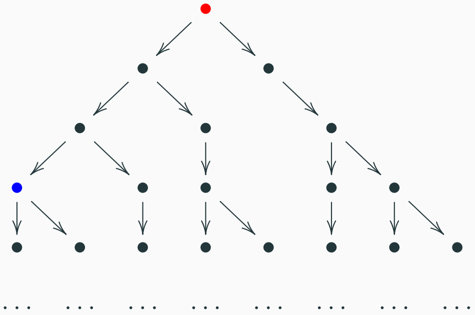
CTL Semantics in Pictures – Combining Connectives

"For All Always" followed by "There Exists Eventually"

$\forall\Box\exists\Diamond\varphi$

for all current and future states,

there exists a state further up in a possible future from there where φ holds



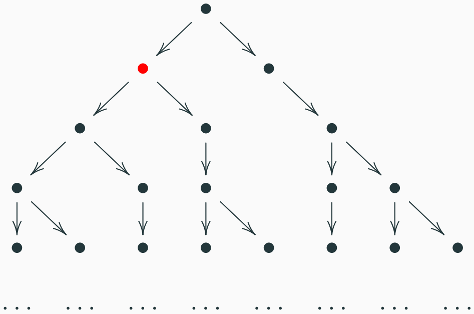
CTL Semantics in Pictures – Combining Connectives

"For All Always" followed by "There Exists Eventually"

$\forall\Box\exists\Diamond\varphi$

for all current and future states,

there exists a state further up in a possible future from there where φ holds



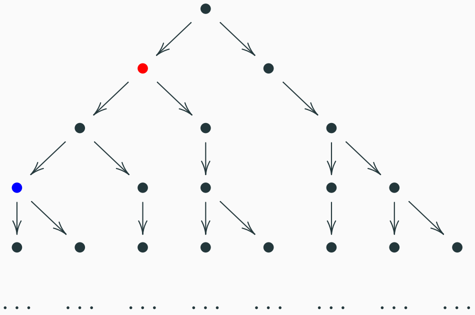
CTL Semantics in Pictures – Combining Connectives

"For All Always" followed by "There Exists Eventually"

$\forall\Box\exists\Diamond\,\varphi$

for all current and future states,

there exists a state further up in a possible future from there where φ holds



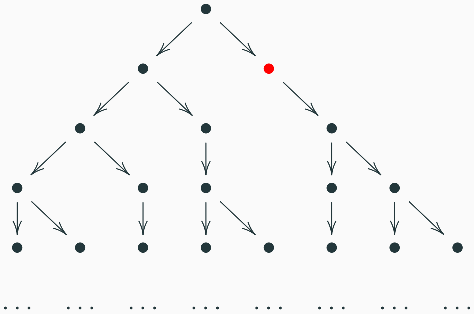
CTL Semantics in Pictures – Combining Connectives

“For All Always” followed by “There Exists Eventually”

$\forall\Box\exists\Diamond\varphi$

for all current and future states,

there exists a state further up in a possible future from there where φ holds



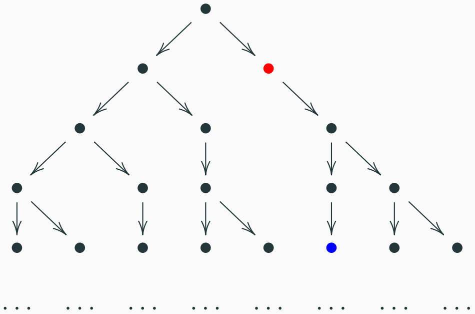
CTL Semantics in Pictures – Combining Connectives

"For All Always" followed by "There Exists Eventually"

$\forall\Box\exists\Diamond\varphi$

for all current and future states,

there exists a state further up in a possible future from there where φ holds



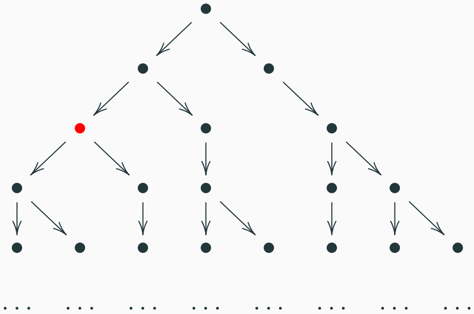
CTL Semantics in Pictures – Combining Connectives

"For All Always" followed by "There Exists Eventually"

$\forall\Box\exists\Diamond\varphi$

for all current and future states,

there exists a state further up in a possible future from there where φ holds



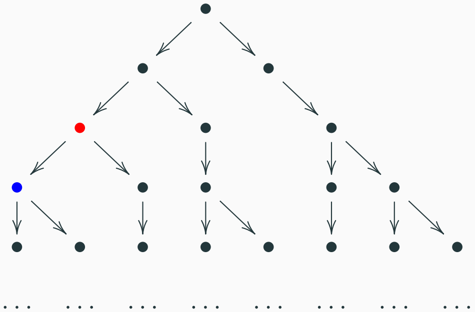
CTL Semantics in Pictures – Combining Connectives

"For All Always" followed by "There Exists Eventually"

$\forall\Box\exists\Diamond\varphi$

for all current and future states,

there exists a state further up in a possible future from there where φ holds



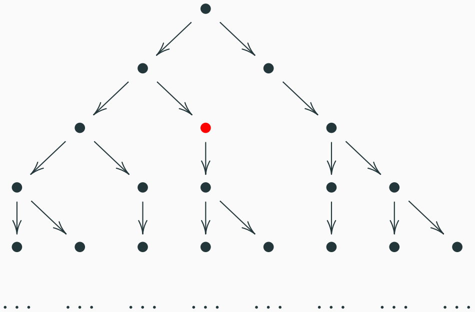
CTL Semantics in Pictures – Combining Connectives

“For All Always” followed by “There Exists Eventually”

$\forall\Box\exists\Diamond\,\varphi$

for all current and future states,

there exists a state further up in a possible future from there where φ holds



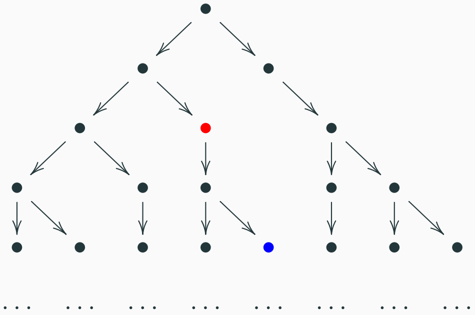
CTL Semantics in Pictures – Combining Connectives

"For All Always" followed by "There Exists Eventually"

$\forall\Box\exists\Diamond\varphi$

for all current and future states,

there exists a state further up in a possible future from there where φ holds



CTL Semantics in Pictures – Combining Connectives

"For All Always" followed by "There Exists Always"

$\forall\Box\exists\Box\varphi$

for all current and future states,

there exists a path starting there on which φ always holds

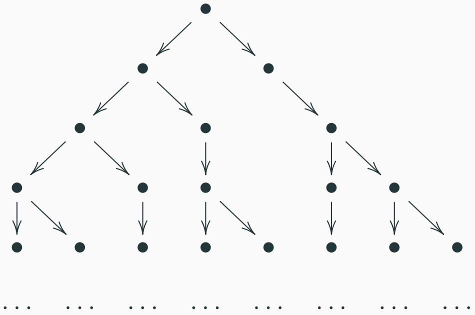
CTL Semantics in Pictures – Combining Connectives

"For All Always" followed by "There Exists Always"

$\forall\Box\exists\Box\varphi$

for all current and future states,

there exists a path starting there on which φ always holds



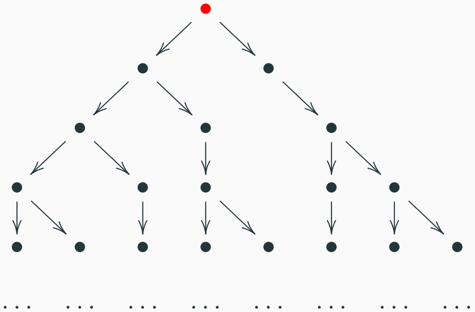
CTL Semantics in Pictures – Combining Connectives

"For All Always" followed by "There Exists Always"

$\forall\Box\exists\Box\varphi$

for all current and future states,

there exists a path starting there on which φ always holds



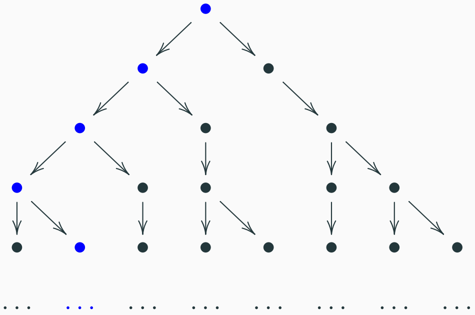
CTL Semantics in Pictures – Combining Connectives

"For All Always" followed by "There Exists Always"

$\forall\Box\exists\Box\varphi$

for all current and future states,

there exists a path starting there on which φ always holds



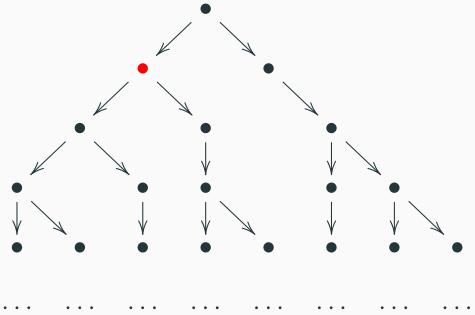
CTL Semantics in Pictures – Combining Connectives

"For All Always" followed by "There Exists Always"

$\forall\Box\exists\Box\varphi$

for all current and future states,

there exists a path starting there on which φ always holds



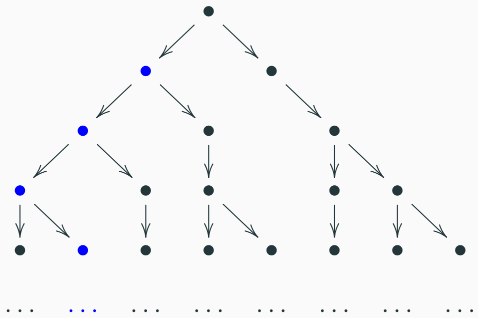
CTL Semantics in Pictures – Combining Connectives

"For All Always" followed by "There Exists Always"

$\forall\Box\exists\Box\varphi$

for all current and future states,

there exists a path starting there on which φ always holds



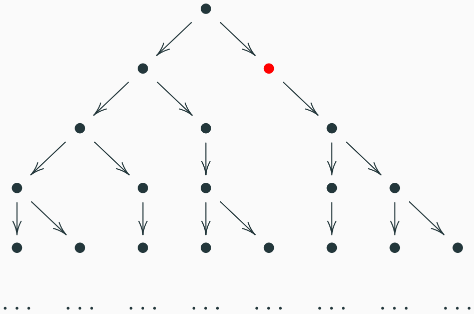
CTL Semantics in Pictures – Combining Connectives

"For All Always" followed by "There Exists Always"

$\forall\Box\exists\Box\varphi$

for all current and future states,

there exists a path starting there on which φ always holds



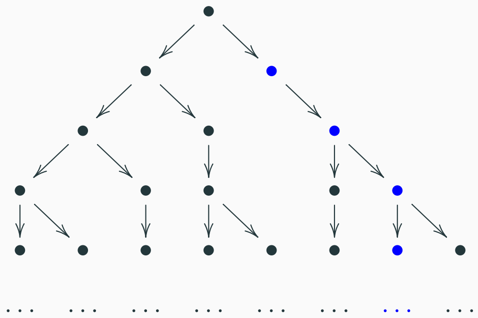
CTL Semantics in Pictures – Combining Connectives

"For All Always" followed by "There Exists Always"

$\forall\Box\exists\Box\varphi$

for all current and future states,

there exists a path starting there on which φ always holds



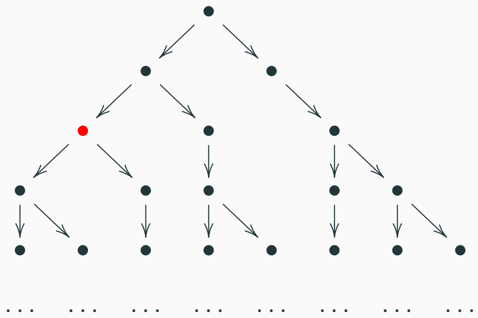
CTL Semantics in Pictures – Combining Connectives

"For All Always" followed by "There Exists Always"

$\forall\Box\exists\Box\varphi$

for all current and future states,

there exists a path starting there on which φ always holds



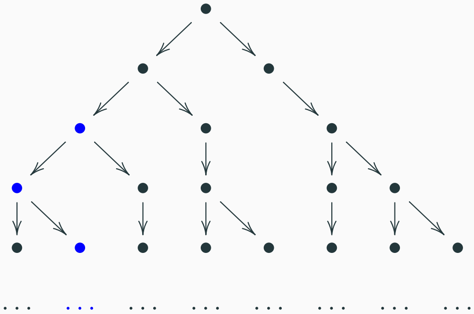
CTL Semantics in Pictures – Combining Connectives

"For All Always" followed by "There Exists Always"

$\forall\Box\exists\Box\varphi$

for all current and future states,

there exists a path starting there on which φ always holds



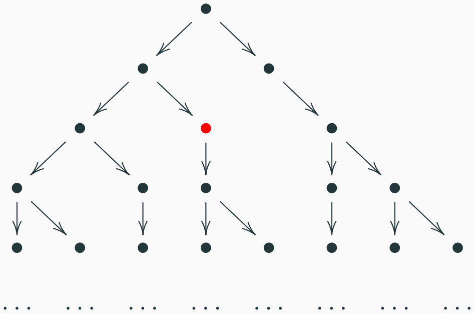
CTL Semantics in Pictures – Combining Connectives

"For All Always" followed by "There Exists Always"

$\forall\Box\exists\Box\varphi$

for all current and future states,

there exists a path starting there on which φ always holds



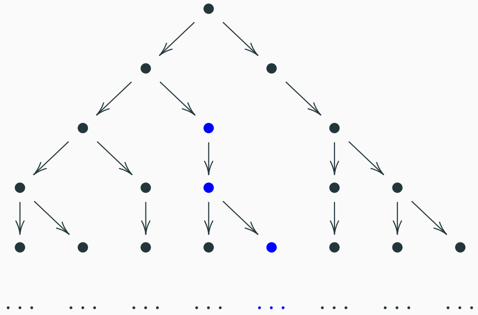
CTL Semantics in Pictures – Combining Connectives

"For All Always" followed by "There Exists Always"

$\forall\Box\exists\Box\varphi$

for all current and future states,

there exists a path starting there on which φ always holds



CTL Semantics in Pictures – Combining Connectives

"There Exists Eventually" followed by "For All Always"

$$\exists\Diamond\forall\Box\varphi$$

there exists a current or future state

such that φ holds on all states further up in all possible futures from there

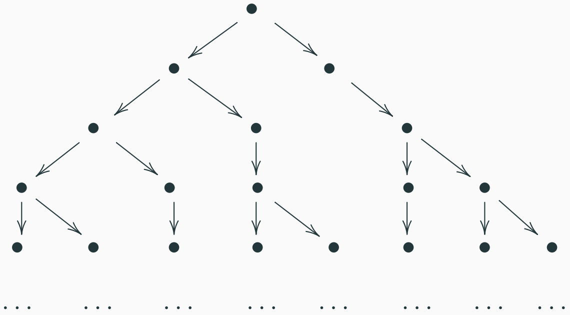
CTL Semantics in Pictures – Combining Connectives

"There Exists Eventually" followed by "For All Always"

$\exists\Diamond\forall\Box\varphi$

there exists a current or future state

such that φ holds on all states further up in all possible futures from there



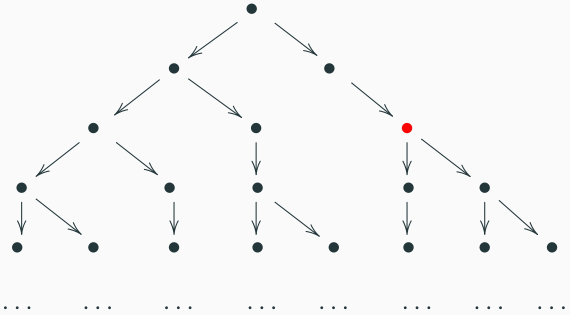
CTL Semantics in Pictures – Combining Connectives

"There Exists Eventually" followed by "For All Always"

$\exists\Diamond\forall\Box\varphi$

there exists a current or future state

such that φ holds on all states further up in all possible futures from there



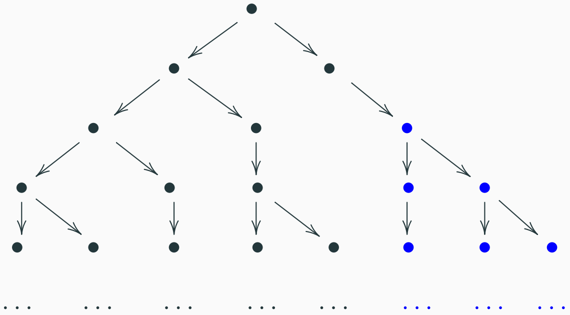
CTL Semantics in Pictures – Combining Connectives

"There Exists Eventually" followed by "For All Always"

$\exists\Diamond\forall\Box\varphi$

there exists a current or future state

such that φ holds on all states further up in all possible futures from there



CTL Semantics in Pictures – Combining Connectives

"There Exists Eventually" followed by "For All Eventually"

$\exists\Diamond\forall\Diamond\varphi$

there exists a current or future state such that,

for all paths starting there, φ eventually holds

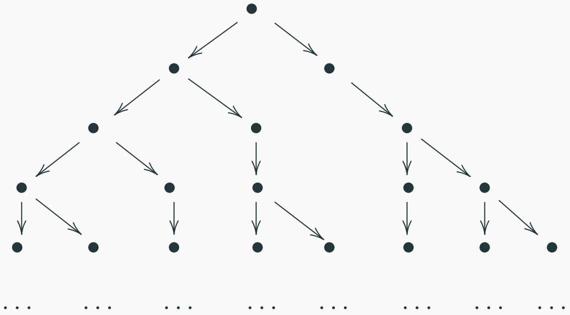
CTL Semantics in Pictures – Combining Connectives

"There Exists Eventually" followed by "For All Eventually"

$\exists\Diamond\forall\Diamond\varphi$

there exists a current or future state such that,

for all paths starting there, φ eventually holds



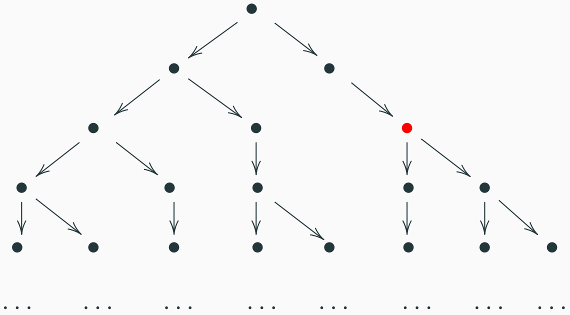
CTL Semantics in Pictures – Combining Connectives

"There Exists Eventually" followed by "For All Eventually"

$\exists\Diamond\forall\Diamond\varphi$

there exists a current or future state such that,

for all paths starting there, φ eventually holds



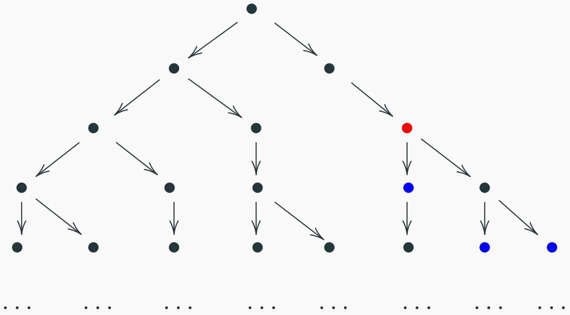
CTL Semantics in Pictures – Combining Connectives

"There Exists Eventually" followed by "For All Eventually"

$\exists\Diamond\forall\Diamond\varphi$

there exists a current or future state such that,

for all paths starting there, φ eventually holds



CTL Semantics in Pictures – Combining Connectives

"For All Always" followed by "Something Implies For All Always"

$$\forall\Box(\varphi \rightarrow \forall\Box\psi)$$

for all current and future states were φ holds,

ψ holds on all states further up in all possible futures from there

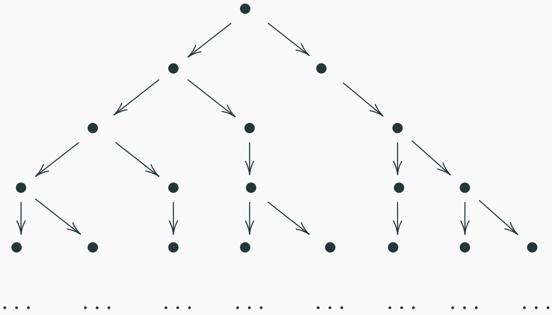
CTL Semantics in Pictures – Combining Connectives

"For All Always" followed by "Something Implies For All Always"

$$\forall\Box(\varphi \to \forall\Box\psi)$$

for all current and future states were φ holds,

ψ holds on all states further up in all possible futures from there



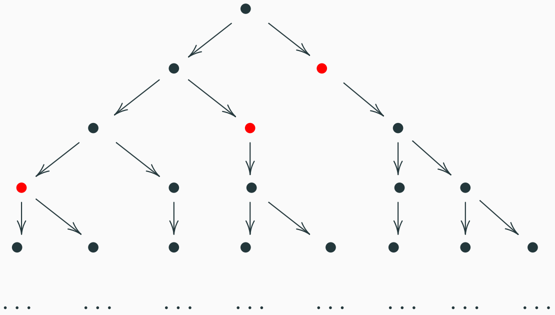
CTL Semantics in Pictures – Combining Connectives

"For All Always" followed by "Something Implies For All Always"

$$\forall\Box(\varphi \rightarrow \forall\Box\psi)$$

for all current and future states were φ holds,

ψ holds on all states further up in all possible futures from there



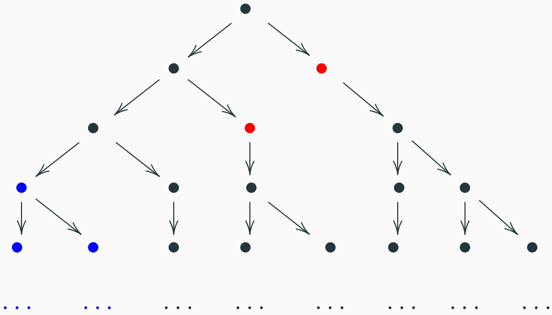
CTL Semantics in Pictures – Combining Connectives

"For All Always" followed by "Something Implies For All Always"

$$\forall\Box(\varphi \to \forall\Box\psi)$$

for all current and future states were φ holds,

ψ holds on all states further up in all possible futures from there



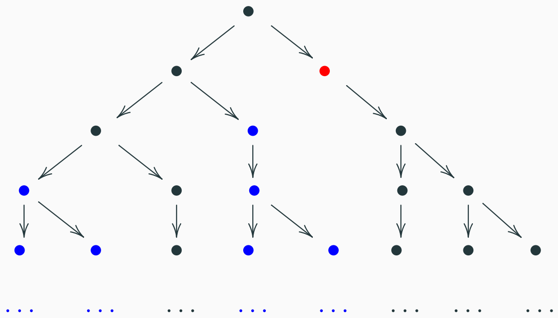
CTL Semantics in Pictures – Combining Connectives

"For All Always" followed by "Something Implies For All Always"

$$\forall\Box(\varphi \to \forall\Box\psi)$$

for all current and future states were φ holds,

ψ holds on all states further up in all possible futures from there



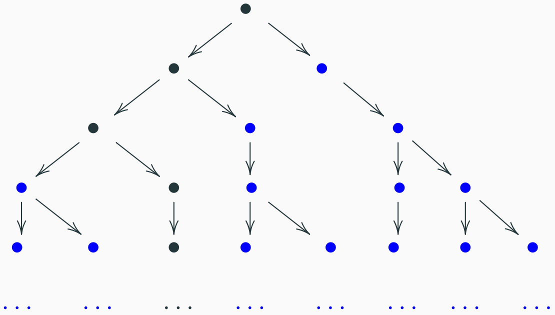
CTL Semantics in Pictures – Combining Connectives

"For All Always" followed by "Something Implies For All Always"

$$\forall\Box(\varphi \rightarrow \forall\Box\psi)$$

for all current and future states were φ holds,

ψ holds on all states further up in all possible futures from there



CTL Semantics in Pictures – Combining Connectives

"For All Always" followed by "Something Implies There Exists Always"

$$\forall\Box(\varphi \to \exists\Box\psi)$$

for all current and future states were φ holds,

there exists a path starting from there on which ψ always holds

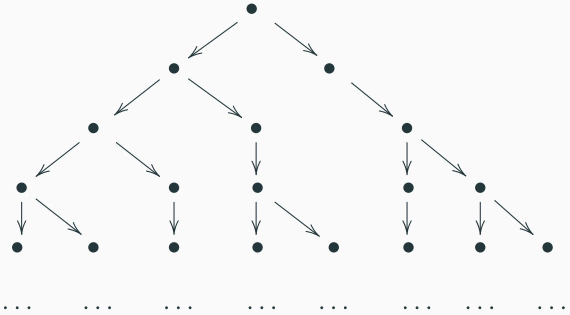
CTL Semantics in Pictures – Combining Connectives

"For All Always" followed by "Something Implies There Exists Always"

$$\forall\Box(\varphi \to \exists\Box\psi)$$

for all current and future states were φ holds,

there exists a path starting from there on which ψ always holds



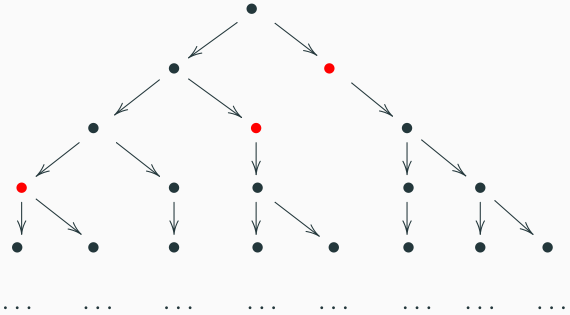
CTL Semantics in Pictures – Combining Connectives

"For All Always" followed by "Something Implies There Exists Always"

$$\forall\Box(\varphi \rightarrow \exists\Box\psi)$$

for all current and future states were φ holds,

there exists a path starting from there on which ψ always holds



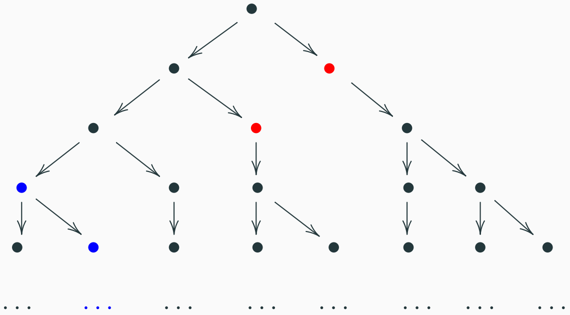
CTL Semantics in Pictures – Combining Connectives

"For All Always" followed by "Something Implies There Exists Always"

$$\forall\Box(\varphi \to \exists\Box\psi)$$

for all current and future states were φ holds,

there exists a path starting from there on which ψ always holds



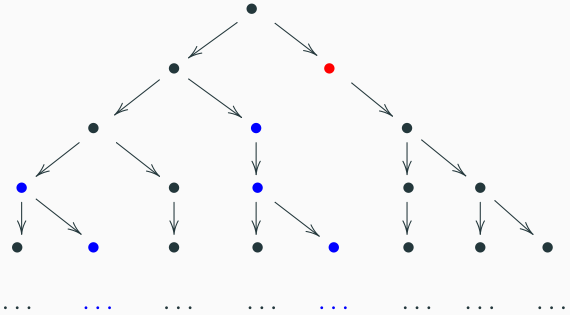
CTL Semantics in Pictures – Combining Connectives

"For All Always" followed by "Something Implies There Exists Always"

$$\forall\Box(\varphi \to \exists\Box\psi)$$

for all current and future states were φ holds,

there exists a path starting from there on which ψ always holds



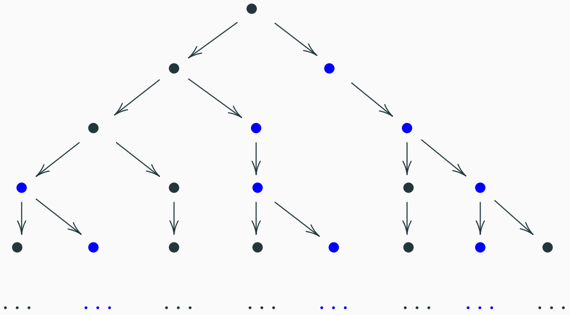
CTL Semantics in Pictures – Combining Connectives

"For All Always" followed by "Something Implies There Exists Always"

$$\forall\Box(\varphi \rightarrow \exists\Box\psi)$$

for all current and future states were φ holds,

there exists a path starting from there on which ψ always holds



CTL Semantics in Pictures – Combining Connectives

"For All Always" followed by "Something Implies For All Eventually"

$$\forall\Box(\varphi \to \forall\Diamond\,\psi)$$

for all current and future states were φ holds

and for all paths starting from there, ψ eventually holds

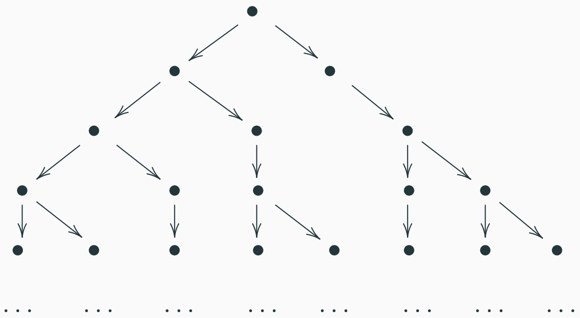
CTL Semantics in Pictures – Combining Connectives

“For All Always” followed by “Something Implies For All Eventually”

$$\forall\Box(\varphi \rightarrow \forall\Diamond\,\psi)$$

for all current and future states were φ holds

and for all paths starting from there, ψ eventually holds



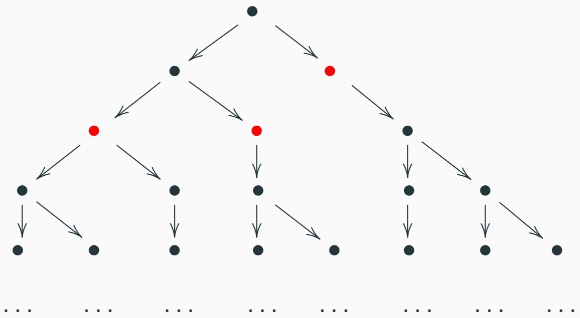
CTL Semantics in Pictures – Combining Connectives

“For All Always” followed by “Something Implies For All Eventually”

$$\forall\Box(\varphi \to \forall\Diamond\,\psi)$$

for all current and future states were φ holds

and for all paths starting from there, ψ eventually holds



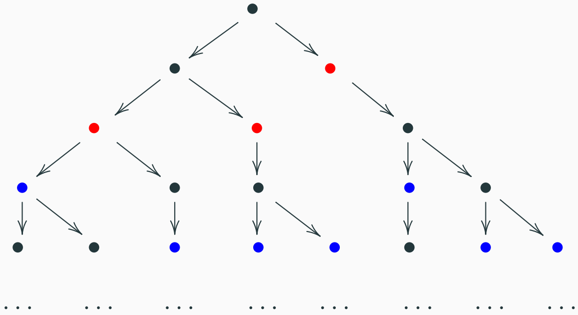
CTL Semantics in Pictures – Combining Connectives

"For All Always" followed by "Something Implies For All Eventually"

$$\forall\Box(\varphi \to \forall\Diamond\,\psi)$$

for all current and future states were φ holds

and for all paths starting from there, ψ eventually holds



Formal Semantics – The Satisfaction Relation

Given an LTS $\mathcal{M} = (S, \rightarrow, L)$, a state $s \in S$ and a CTL formula φ, we define

"$\mathcal{M}$ in state s satisfies φ" or *"φ holds for $\mathcal{M}$ in state s"*

$$\mathcal{M}, s \models \varphi$$

by structural recursion on φ:

Formal Semantics – The Satisfaction Relation

Given an LTS $\mathcal{M} = (S, \rightarrow, L)$, a state $s \in S$ and a CTL formula φ, we define

"$\mathcal{M}$ in state s satisfies φ" or *"φ holds for $\mathcal{M}$ in state s"*

$$\mathcal{M}, s \models \varphi$$

by structural recursion on φ:

$\mathcal{M}, s \models a$ iff $a \in L(s)$

Formal Semantics – The Satisfaction Relation

Given an LTS $\mathcal{M} = (S, \rightarrow, L)$, a state $s \in S$ and a CTL formula φ, we define

"$\mathcal{M}$ in state s satisfies φ" or *"φ holds for $\mathcal{M}$ in state s"*

$$\mathcal{M}, s \models \varphi$$

by structural recursion on φ:

$\mathcal{M}, s \models a$	iff	$a \in L(s)$
$\mathcal{M}, s \models \neg\, \varphi$	iff	$\mathcal{M}, s \not\models \varphi$
$\mathcal{M}, s \models \varphi \vee \psi$	iff	$\mathcal{M}, s \models \varphi$ or $\mathcal{M}, s \models \psi$
$\mathcal{M}, s \models \varphi \wedge \psi$	iff	$\mathcal{M}, s \models \varphi$ and $\mathcal{M}, s \models \psi$
$\mathcal{M}, s \models \varphi \rightarrow \psi$	iff	$\mathcal{M}, s \models \varphi$ implies $\mathcal{M}, s \models \psi$

Formal Semantics – The Satisfaction Relation

Given an LTS $\mathcal{M} = (S, \to, L)$, a state $s \in S$ and a CTL formula φ, we define

"$\mathcal{M}$ in state s satisfies φ" or *"φ holds for $\mathcal{M}$ in state s"*

$$\mathcal{M}, s \models \varphi$$

by structural recursion on φ:

$\mathcal{M}, s \models a$ iff $a \in L(s)$

$\mathcal{M}, s \models \neg\, \varphi$ iff $\mathcal{M}, s \not\models \varphi$

$\mathcal{M}, s \models \varphi \vee \psi$ iff $\mathcal{M}, s \models \varphi$ or $\mathcal{M}, s \models \psi$

$\mathcal{M}, s \models \varphi \wedge \psi$ iff $\mathcal{M}, s \models \varphi$ and $\mathcal{M}, s \models \psi$

$\mathcal{M}, s \models \varphi \to \psi$ iff $\mathcal{M}, s \models \varphi$ implies $\mathcal{M}, s \models \psi$

$\mathcal{M}, s \models \forall\bigcirc\ \varphi$ iff for all $s' \in S$ such that $s \to s'$ we have $\mathcal{M}, s' \models \varphi$

Formal Semantics – The Satisfaction Relation

Given an LTS $\mathcal{M} = (S, \rightarrow, L)$, a state $s \in S$ and a CTL formula φ, we define

"$\mathcal{M}$ in state s satisfies φ" or *"φ holds for $\mathcal{M}$ in state s"*

$$\mathcal{M}, s \models \varphi$$

by structural recursion on φ:

$\mathcal{M}, s \models a$ iff $a \in L(s)$

$\mathcal{M}, s \models \neg\, \varphi$ iff $\mathcal{M}, s \not\models \varphi$

$\mathcal{M}, s \models \varphi \vee \psi$ iff $\mathcal{M}, s \models \varphi$ or $\mathcal{M}, s \models \psi$

$\mathcal{M}, s \models \varphi \wedge \psi$ iff $\mathcal{M}, s \models \varphi$ and $\mathcal{M}, s \models \psi$

$\mathcal{M}, s \models \varphi \rightarrow \psi$ iff $\mathcal{M}, s \models \varphi$ implies $\mathcal{M}, s \models \psi$

$\mathcal{M}, s \models \forall\bigcirc\ \varphi$ iff for all $s' \in S$ such that $s \rightarrow s'$ we have $\mathcal{M}, s' \models \varphi$

$\mathcal{M}, s \models \exists\bigcirc\ \varphi$ iff there exists $s' \in S$ such that $s \rightarrow s'$ and $\mathcal{M}, s' \models \varphi$

Formal Semantics – The Satisfaction Relation

Assume that paths π have the form $s_0 s_1 s_2 \ldots$

Formal Semantics – The Satisfaction Relation

Assume that paths π have the form $s_0 s_1 s_2 \dots$

$\mathcal{M}, s \models \forall\Diamond\varphi$ iff for all paths π such that $s_0 = s$
there exists i such that $\mathcal{M}, s_i \models \varphi$

Formal Semantics – The Satisfaction Relation

Assume that paths π have the form $s_0 s_1 s_2 \ldots$

$\mathcal{M}, s \models \forall\Diamond\varphi$ iff for all paths π such that $s_0 = s$
there exists i such that $\mathcal{M}, s_i \models \varphi$

$\mathcal{M}, s \models \exists\Diamond\varphi$ iff there exists a path π such that $s_0 = s$ and
there exists i such that $\mathcal{M}, s_i \models \varphi$

Formal Semantics – The Satisfaction Relation

Assume that paths π have the form $s_0 s_1 s_2 \ldots$

$\mathcal{M}, s \models \forall\Diamond\varphi$ iff for all paths π such that $s_0 = s$
there exists i such that $\mathcal{M}, s_i \models \varphi$

$\mathcal{M}, s \models \exists\Diamond\varphi$ iff there exists a path π such that $s_0 = s$ and
there exists i such that $\mathcal{M}, s_i \models \varphi$

$\mathcal{M}, s \models \forall\Box\varphi$ iff for all paths π such that $s_0 = s$
for all i, $\mathcal{M}, s_i \models \varphi$

$\mathcal{M}, s \models \exists\Box\varphi$ iff there exists a path π such that $s_0 = s$ and
for all i, $\mathcal{M}, s_i \models \varphi$

Formal Semantics – The Satisfaction Relation

Assume that paths π have the form $s_0 s_1 s_2 \ldots$

$\mathcal{M}, s \models \varphi \; \forall \mathsf{U} \; \psi$ iff for all paths π such that $s_0 = s$
there exists i such that $\mathcal{M}, s_i \models \psi$ and
for all $j < i$, $\mathcal{M}, s_j \models \varphi$

Formal Semantics – The Satisfaction Relation

Assume that paths π have the form $s_0 s_1 s_2 \ldots$

$\mathcal{M}, s \models \varphi \ \forall\mathsf{U}\ \psi$	iff	for all paths π such that $s_0 = s$ there exists i such that $\mathcal{M}, s_i \models \psi$ and for all $j < i$, $\mathcal{M}, s_j \models \varphi$
$\mathcal{M}, s \models \varphi \ \exists\mathsf{U}\ \psi$	iff	there exists a path π such that $s_0 = s$ and there exists i such that $\mathcal{M}, s_i \models \psi$ and for all $j < i$, $\mathcal{M}, s_j \models \varphi$

Formula Equivalence in CTL

Two CTL formulas φ and ψ are equivalent, denoted $\varphi \equiv \psi$, if they are satisfied by (i.e., hold for) exactly the same LTSs in the same states

Formula Equivalence in CTL

Two CTL formulas φ and ψ are equivalent, denoted $\varphi \equiv \psi$, if they are satisfied by (i.e., hold for) exactly the same LTSs in the same states:

Given any LTS $\mathcal{M} = (S, \rightarrow, L)$ and any $s \in S$, we have that

$$\mathcal{M}, s \models \varphi \text{ iff } \mathcal{M}, s \models \psi.$$

Some CTL Formula Equivalences

Just like for LTL, $\top$ (read "True") abbreviates $a \rightarrow a$ for some atom a.

$$
\begin{aligned}
\exists\Box\forall\Box\varphi &\equiv \forall\Box\forall\Box\varphi \equiv \forall\Box\varphi \\
\neg\exists\bigcirc\varphi &\equiv \forall\bigcirc\neg\varphi \\
\neg\exists\Diamond\varphi &\equiv \forall\Box\neg\varphi \\
\neg\exists\Box\varphi &\equiv \forall\Diamond\neg\varphi \\
\forall\Diamond\varphi &\equiv \top\ \forall\mathsf{U}\ \varphi \\
\exists\Diamond\varphi &\equiv \top\ \exists\mathsf{U}\ \varphi \\
\varphi\ \forall\mathsf{U}\ \psi &\equiv \neg((\neg\psi\ \exists\mathsf{U}\ (\neg\varphi \wedge \neg\psi)) \vee \exists\Box\neg\psi)
\end{aligned}
$$

Homework Exercise. Apply the CTL semantics to prove the above equivalences.

Proving CTL Formula Equivalence

For example, let's prove $\exists\Box\forall\Box\varphi \equiv \forall\Box\varphi$.

Proving CTL Formula Equivalence

For example, let's prove $\exists\Box\forall\Box\varphi \equiv \forall\Box\varphi$.

Let $\mathcal{M} = (S, \rightarrow, L)$ be an LTS and $s_0 \in S$. We must prove

$$\mathcal{M}, s_0 \models \exists\Box\forall\Box\varphi \text{ iff } \mathcal{M}, s_0 \models \forall\Box\varphi$$

Proving CTL Formula Equivalence

For example, let's prove $\exists\Box\forall\Box\varphi \equiv \forall\Box\varphi$.

Let $\mathcal{M} = (S, \to, L)$ be an LTS and $s_0 \in S$. We must prove

$$\mathcal{M}, s_0 \models \exists\Box\forall\Box\varphi \quad \text{iff} \quad \mathcal{M}, s_0 \models \forall\Box\varphi$$

To this end, we rewrite both sides applying the semantics of CTL connectives.

Proving CTL Formula Equivalence

For example, let's prove $\exists\Box\forall\Box\varphi \equiv \forall\Box\varphi$.

Let $\mathcal{M} = (S, \rightarrow, L)$ be an LTS and $s_0 \in S$. We must prove

$$\mathcal{M}, s_0 \models \exists\Box\forall\Box\varphi \quad \text{iff} \quad \mathcal{M}, s_0 \models \forall\Box\varphi$$

To this end, we rewrite both sides applying the semantics of CTL connectives.

On the one hand, we have the following chain of equivalent statements:

$\mathcal{M}, s_0 \models \exists\Box\forall\Box\varphi$

Proving CTL Formula Equivalence

For example, let's prove $\exists\Box\forall\Box\varphi \equiv \forall\Box\varphi$.

Let $\mathcal{M} = (S, \to, L)$ be an LTS and $s_0 \in S$. We must prove

$$\mathcal{M}, s_0 \models \exists\Box\forall\Box\varphi \quad \text{iff} \quad \mathcal{M}, s_0 \models \forall\Box\varphi$$

To this end, we rewrite both sides applying the semantics of CTL connectives.

On the one hand, we have the following chain of equivalent statements:

$\mathcal{M}, s_0 \models \exists\Box\forall\Box\varphi$
iff (by the semantics of $\exists\Box$)
there exists $\pi = s_0 s_1 s_2 \ldots \in Paths_{s_0}(\mathcal{M})$ such that, for all i,
$\mathcal{M}, s_i \models \forall\Box\varphi$

Proving CTL Formula Equivalence

For example, let's prove $\exists\Box\forall\Box\varphi \equiv \forall\Box\varphi$.

Let $\mathcal{M} = (S, \to, L)$ be an LTS and $s_0 \in S$. We must prove

$$\mathcal{M}, s_0 \models \exists\Box\forall\Box\varphi \ \text{ iff } \ \mathcal{M}, s_0 \models \forall\Box\varphi$$

To this end, we rewrite both sides applying the semantics of CTL connectives.

On the one hand, we have the following chain of equivalent statements:

$\mathcal{M}, s_0 \models \exists\Box\forall\Box\varphi$
iff (by the semantics of $\exists\Box$)
there exists $\pi = s_0 s_1 s_2 \ldots \in Paths_{s_0}(\mathcal{M})$ such that, for all i,
$\mathcal{M}, s_i \models \forall\Box\varphi$
iff (by the semantics of $\forall\Box$)
there exists $\pi = s_0 s_1 s_2 \ldots \in Paths_{s_0}(\mathcal{M})$ such that, for all i,
for all $\pi' = s'_0 s'_1 s'_2 \ldots \in Paths_{s_i}(\mathcal{M})$, for all j, $\mathcal{M}, s'_j \models \varphi$

Proving CTL Formula Equivalence

For example, let's prove $\exists\Box\forall\Box\varphi \equiv \forall\Box\varphi$.

Let $\mathcal{M} = (S, \rightarrow, L)$ be an LTS and $s_0 \in S$. We must prove

$$\mathcal{M}, s_0 \models \exists\Box\forall\Box\varphi \text{ iff } \mathcal{M}, s_0 \models \forall\Box\varphi$$

To this end, we rewrite both sides applying the semantics of CTL connectives.

On the one hand, we have the following chain of equivalent statements:

$\mathcal{M}, s_0 \models \exists\Box\forall\Box\varphi$
  iff (by the semantics of $\exists\Box$)
there exists $\pi = s_0 s_1 s_2 \ldots \in Paths_{s_0}(\mathcal{M})$ such that, for all i,
  $\mathcal{M}, s_i \models \forall\Box\varphi$
  iff (by the semantics of $\forall\Box$)
there exists $\pi = s_0 s_1 s_2 \ldots \in Paths_{s_0}(\mathcal{M})$ such that, for all i,
  for all $\pi' = s'_0 s'_1 s'_2 \ldots \in Paths_{s_i}(\mathcal{M})$, for all j, $\mathcal{M}, s'_j \models \varphi$

Proving CTL Formula Equivalence

For example, let's prove $\exists\Box\forall\Box\varphi \equiv \forall\Box\varphi$.

Let $\mathcal{M} = (S, \to, L)$ be an LTS and $s_0 \in S$. We must prove

$$\mathcal{M}, s_0 \models \exists\Box\forall\Box\varphi \;\text{ iff }\; \mathcal{M}, s_0 \models \forall\Box\varphi$$

To this end, we rewrite both sides applying the semantics of CTL connectives.

Proving CTL Formula Equivalence

For example, let's prove $\exists\Box\forall\Box\varphi \equiv \forall\Box\varphi$.

Let $\mathcal{M} = (S, \rightarrow, L)$ be an LTS and $s_0 \in S$. We must prove

$$\mathcal{M}, s_0 \models \exists\Box\forall\Box\varphi \text{ iff } \mathcal{M}, s_0 \models \forall\Box\varphi$$

To this end, we rewrite both sides applying the semantics of CTL connectives.

On the one hand, we have the following equivalent statements:

$\mathcal{M}, s_0 \models \forall\Box\varphi$

Proving CTL Formula Equivalence

For example, let's prove $\exists\Box\forall\Box\varphi \equiv \forall\Box\varphi$.

Let $\mathcal{M} = (S, \rightarrow, L)$ be an LTS and $s_0 \in S$. We must prove

$$\mathcal{M}, s_0 \models \exists\Box\forall\Box\varphi \ \text{ iff } \ \mathcal{M}, s_0 \models \forall\Box\varphi$$

To this end, we rewrite both sides applying the semantics of CTL connectives.

On the one hand, we have the following equivalent statements:

$\mathcal{M}, s_0 \models \forall\Box\varphi$
  iff (by the semantics of $\forall\Box$)
for all $\pi = s_0 s_1 s_2 \ldots \in Paths_{s_0}(\mathcal{M})$, for all i, $\mathcal{M}, s_i \models \varphi$

Proving CTL Formula Equivalence

For example, let's prove $\exists\Box\forall\Box\varphi \equiv \forall\Box\varphi$.

So, given an LTS $\mathcal{M} = (S, \rightarrow, L)$ and $s_0 \in S$, we are left the prove that the following statements are equivalent:

(1) there exists $\pi = s_0 s_1 s_2 \ldots \in \mathit{Paths}_{s_0}(\mathcal{M})$ such that, for all i, for all $\pi' = s'_0 s'_1 s'_2 \ldots \in \mathit{Paths}_{s_i}(\mathcal{M})$, for all j, $\mathcal{M}, s'_j \models \varphi$

(2) for all $\pi = s_0 s_1 s_2 \ldots \in \mathit{Paths}_{s_0}(\mathcal{M})$, for all i, $\mathcal{M}, s_i \models \varphi$

Proving CTL Formula Equivalence

For example, let's prove $\exists\Box\forall\Box\varphi \equiv \forall\Box\varphi$.

So, given an LTS $\mathcal{M} = (S, \rightarrow, L)$ and $s_0 \in S$, we are left the prove that the following statements are equivalent:

(1) there exists $\pi = s_0 s_1 s_2 \ldots \in \mathit{Paths}_{s_0}(\mathcal{M})$ such that, for all i,
for all $\pi' = s'_0 s'_1 s'_2 \ldots \in \mathit{Paths}_{s_i}(\mathcal{M})$, for all j, $\mathcal{M}, s'_j \models \varphi$

(2) for all $\pi = s_0 s_1 s_2 \ldots \in \mathit{Paths}_{s_0}(\mathcal{M})$, for all i, $\mathcal{M}, s_i \models \varphi$

First, assume (1) and let's prove (2):

Proving CTL Formula Equivalence

For example, let's prove $\exists\Box\forall\Box\varphi \equiv \forall\Box\varphi$.

So, given an LTS $\mathcal{M} = (S, \rightarrow, L)$ and $s_0 \in S$, we are left the prove that the following statements are equivalent:

(1) there exists $\pi = s_0 s_1 s_2 \ldots \in \mathit{Paths}_{s_0}(\mathcal{M})$ such that, for all i,
for all $\pi' = s'_0 s'_1 s'_2 \ldots \in \mathit{Paths}_{s_i}(\mathcal{M})$, for all j, $\mathcal{M}, s'_j \models \varphi$

(2) for all $\pi = s_0 s_1 s_2 \ldots \in \mathit{Paths}_{s_0}(\mathcal{M})$, for all i, $\mathcal{M}, s_i \models \varphi$

First, assume (1) and let's prove (2):
Let $\pi' = s'_0 s'_1 s'_2 \ldots \in \mathit{Paths}_{s_0}(\mathcal{M})$ (so $s'_0 = s_0$) and j. We must prove $\mathcal{M}, s'_j \models \varphi$.

Proving CTL Formula Equivalence

For example, let's prove $\exists\Box\forall\Box\varphi \equiv \forall\Box\varphi$.

So, given an LTS $\mathcal{M} = (S, \rightarrow, L)$ and $s_0 \in S$, we are left the prove that the following statements are equivalent:

(1) there exists $\pi = s_0 s_1 s_2 \ldots \in Paths_{s_0}(\mathcal{M})$ such that, for all i,
for all $\pi' = s'_0 s'_1 s'_2 \ldots \in Paths_{s_i}(\mathcal{M})$, for all j, $\mathcal{M}, s'_j \models \varphi$

(2) for all $\pi = s_0 s_1 s_2 \ldots \in Paths_{s_0}(\mathcal{M})$, for all i, $\mathcal{M}, s_i \models \varphi$

First, assume (1) and let's prove (2):
Let $\pi' = s'_0 s'_1 s'_2 \ldots \in Paths_{s_0}(\mathcal{M})$ (so $s'_0 = s_0$) and j. We must prove $\mathcal{M}, s'_j \models \varphi$.

Proving CTL Formula Equivalence

For example, let's prove $\exists\Box\forall\Box\varphi \equiv \forall\Box\varphi$.

So, given an LTS $\mathcal{M} = (S, \rightarrow, L)$ and $s_0 \in S$, we are left the prove that the following statements are equivalent:

(1) there exists $\pi = s_0 s_1 s_2 \ldots \in Paths_{s_0}(\mathcal{M})$ such that, for all i,
for all $\pi' = s'_0 s'_1 s'_2 \ldots \in Paths_{s_i}(\mathcal{M})$, for all j, $\mathcal{M}, s'_j \models \varphi$

(2) for all $\pi' = s'_0 s'_1 s'_2 \ldots \in Paths_{s_0}(\mathcal{M})$, for all j, $\mathcal{M}, s'_j \models \varphi$

First, assume (1) and let's prove (2):
Let $\pi' = s'_0 s'_1 s'_2 \ldots \in Paths_{s_0}(\mathcal{M})$ (so $s'_0 = s_0$) and j. We must prove $\mathcal{M}, s'_j \models \varphi$.

Proving CTL Formula Equivalence

For example, let's prove $\exists\Box\forall\Box\varphi \equiv \forall\Box\varphi$.

So, given an LTS $\mathcal{M} = (S, \rightarrow, L)$ and $s_0 \in S$, we are left the prove that the following statements are equivalent:

(1) there exists $\pi = s_0 s_1 s_2 \ldots \in Paths_{s_0}(\mathcal{M})$ such that, for all i, for all $\pi' = s'_0 s'_1 s'_2 \ldots \in Paths_{s_i}(\mathcal{M})$, for all j, $\mathcal{M}, s'_j \models \varphi$

(2) for all $\pi' = s'_0 s'_1 s'_2 \ldots \in Paths_{s_0}(\mathcal{M})$, for all j, $\mathcal{M}, s'_j \models \varphi$

First, assume (1) and let's prove (2):
Let $\pi' = s'_0 s'_1 s'_2 \ldots \in Paths_{s_0}(\mathcal{M})$ (so $s'_0 = s_0$) and j. We must prove $\mathcal{M}, s'_j \models \varphi$.

Proving CTL Formula Equivalence

For example, let's prove $\exists\Box\forall\Box\varphi \equiv \forall\Box\varphi$.

So, given an LTS $\mathcal{M} = (S, \rightarrow, L)$ and $s_0 \in S$, we are left the prove that the following statements are equivalent:

(1) there exists $\pi = s_0 s_1 s_2 \ldots \in \mathit{Paths}_{s_0}(\mathcal{M})$ such that, for all i,
for all $\pi' = s'_0 s'_1 s'_2 \ldots \in \mathit{Paths}_{s_i}(\mathcal{M})$, for all j, $\mathcal{M}, s'_j \models \varphi$

(2) for all $\pi' = s'_0 s'_1 s'_2 \ldots \in \mathit{Paths}_{s_0}(\mathcal{M})$, for all j, $\mathcal{M}, s'_j \models \varphi$

First, assume (1) and let's prove (2):
Let $\pi' = s'_0 s'_1 s'_2 \ldots \in \mathit{Paths}_{s_0}(\mathcal{M})$ (so $s'_0 = s_0$) and j. We must prove $\mathcal{M}, s'_j \models \varphi$.
Let $\pi = s_0 s_1 s_2 \ldots \in \mathit{Paths}_{s_0}(\mathcal{M})$ be the path obtained from (1).

Proving CTL Formula Equivalence

For example, let's prove $\exists\Box\forall\Box\varphi \equiv \forall\Box\varphi$.

So, given an LTS $\mathcal{M} = (S, \rightarrow, L)$ and $s_0 \in S$, we are left the prove that the following statements are equivalent:

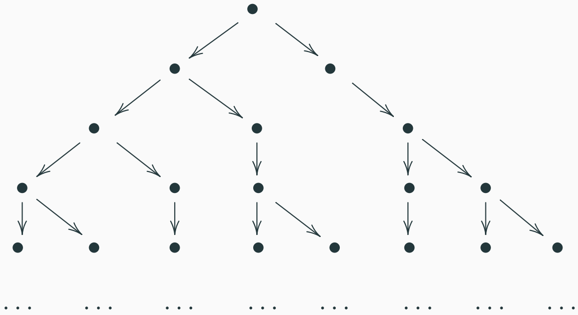
(1) there exists $\pi = s_0 s_1 s_2 \ldots \in \mathit{Paths}_{s_0}(\mathcal{M})$ such that, for all i,
for all $\pi' = s'_0 s'_1 s'_2 \ldots \in \mathit{Paths}_{s_i}(\mathcal{M})$, for all j, $\mathcal{M}, s'_j \models \varphi$

(2) for all $\pi' = s'_0 s'_1 s'_2 \ldots \in \mathit{Paths}_{s_0}(\mathcal{M})$, for all j, $\mathcal{M}, s'_j \models \varphi$

First, assume (1) and let's prove (2):
Let $\pi' = s'_0 s'_1 s'_2 \ldots \in \mathit{Paths}_{s_0}(\mathcal{M})$ (so $s'_0 = s_0$) and j. We must prove $\mathcal{M}, s'_j \models \varphi$.
Let $\pi = s_0 s_1 s_2 \ldots \in \mathit{Paths}_{s_0}(\mathcal{M})$ be the path obtained from (1).
Applying (1) to 0, π' and j, we obtain $\mathcal{M}, s'_j \models \varphi$, as desired.

We know:

(1) there exists $\pi = s_0 s_1 s_2 \ldots \in \mathit{Paths}_{s_0}(\mathcal{M})$ such that, for all i,
for all $\pi' = s'_0 s'_1 s'_2 \ldots \in \mathit{Paths}_{s_i}(\mathcal{M})$, for all j, $\mathcal{M}, s'_j \models \varphi$



We show:

(2) for all $\pi' = s'_0 s'_1 s'_2 \ldots \in \mathit{Paths}_{s_0}(\mathcal{M})$, for all j, $\mathcal{M}, s'_j \models \varphi$

Proving CTL Formula Equivalence

For example, let's prove $\exists\Box\forall\Box\varphi \equiv \forall\Box\varphi$.

So, given an LTS $\mathcal{M} = (S, \rightarrow, L)$ and $s_0 \in S$, we are left the prove that the following statements are equivalent:

(1) there exists $\pi = s_0 s_1 s_2 \ldots \in \mathit{Paths}_{s_0}(\mathcal{M})$ such that, for all i,
for all $\pi' = s'_0 s'_1 s'_2 \ldots \in \mathit{Paths}_{s_i}(\mathcal{M})$, for all j, $\mathcal{M}, s'_j \models \varphi$

(2) for all $\pi = s_0 s_1 s_2 \ldots \in \mathit{Paths}_{s_0}(\mathcal{M})$, for all i, $\mathcal{M}, s_i \models \varphi$

Finally, assume (2) and let's prove (1):

Proving CTL Formula Equivalence

For example, let's prove $\exists\Box\forall\Box\varphi \equiv \forall\Box\varphi$.

So, given an LTS $\mathcal{M} = (S, \rightarrow, L)$ and $s_0 \in S$, we are left the prove that the following statements are equivalent:

(1) there exists $\pi = s_0 s_1 s_2 \ldots \in Paths_{s_0}(\mathcal{M})$ such that, for all i,
for all $\pi' = s'_0 s'_1 s'_2 \ldots \in Paths_{s_i}(\mathcal{M})$, for all j, $\mathcal{M}, s'_j \models \varphi$

(2) for all $\pi = s_0 s_1 s_2 \ldots \in Paths_{s_0}(\mathcal{M})$, for all i, $\mathcal{M}, s_i \models \varphi$

Finally, assume (2) and let's prove (1):
Take $\pi = s_0 s_1 s_2 \ldots \in Paths_{s_0}(\mathcal{M})$ to be any path starting in s_0 (such a path exists because every state has an outward transition).

Proving CTL Formula Equivalence

For example, let's prove $\exists\Box\forall\Box\varphi \equiv \forall\Box\varphi$.

So, given an LTS $\mathcal{M} = (S, \rightarrow, L)$ and $s_0 \in S$, we are left the prove that the following statements are equivalent:

(1) there exists $\pi = s_0 s_1 s_2 \ldots \in Paths_{s_0}(\mathcal{M})$ such that, for all i,
for all $\pi' = s'_0 s'_1 s'_2 \ldots \in Paths_{s_i}(\mathcal{M})$, for all j, $\mathcal{M}, s'_j \models \varphi$

(2) for all $\pi = s_0 s_1 s_2 \ldots \in Paths_{s_0}(\mathcal{M})$, for all i, $\mathcal{M}, s_i \models \varphi$

Finally, assume (2) and let's prove (1):
Take $\pi = s_0 s_1 s_2 \ldots \in Paths_{s_0}(\mathcal{M})$ to be any path starting in s_0 (such a path exists because every state has an outward transition).
Let i, $\pi' = s'_0 s'_1 s'_2 \ldots \in Paths_{s_i}(\mathcal{M})$ and j. (Note that $s'_0 = s_i$.)
We must prove $\mathcal{M}, s'_j \models \varphi$.

Proving CTL Formula Equivalence

For example, let's prove $\exists\Box\forall\Box\varphi \equiv \forall\Box\varphi$.

So, given an LTS $\mathcal{M} = (S, \rightarrow, L)$ and $s_0 \in S$, we are left the prove that the following statements are equivalent:

(1) there exists $\pi = s_0 s_1 s_2 \ldots \in Paths_{s_0}(\mathcal{M})$ such that, for all i,
for all $\pi' = s'_0 s'_1 s'_2 \ldots \in Paths_{s_i}(\mathcal{M})$, for all j, $\mathcal{M}, s'_j \models \varphi$

(2) for all $\pi = s_0 s_1 s_2 \ldots \in Paths_{s_0}(\mathcal{M})$, for all i, $\mathcal{M}, s_i \models \varphi$

Finally, assume (2) and let's prove (1):
Take $\pi = s_0 s_1 s_2 \ldots \in Paths_{s_0}(\mathcal{M})$ to be any path starting in s_0 (such a path exists because every state has an outward transition).
Let i, $\pi' = s'_0 s'_1 s'_2 \ldots \in Paths_{s_i}(\mathcal{M})$ and j. (Note that $s'_0 = s_i$.)
We must prove $\mathcal{M}, s'_j \models \varphi$.
To this end, we apply (2) to the path $\pi'' = s''_0 s''_1 s''_2 \ldots \in Paths_{s_0}(\mathcal{M})$ defined as $s_0 s_1 \ldots s_{i-1} s'_0 s'_1 s'_2 \ldots$ and to the number $i + j$, obtaining $\mathcal{M}, s''_{i+j} \models \varphi$.

Proving CTL Formula Equivalence

For example, let's prove $\exists\Box\forall\Box\varphi \equiv \forall\Box\varphi$.

So, given an LTS $\mathcal{M} = (S, \rightarrow, L)$ and $s_0 \in S$, we are left the prove that the following statements are equivalent:

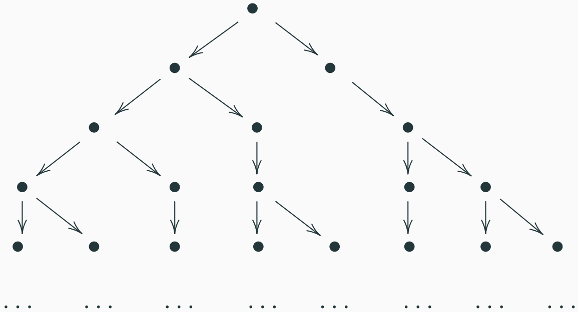
(1) there exists $\pi = s_0 s_1 s_2 \ldots \in Paths_{s_0}(\mathcal{M})$ such that, for all i,
for all $\pi' = s'_0 s'_1 s'_2 \ldots \in Paths_{s_i}(\mathcal{M})$, for all j, $\mathcal{M}, s'_j \models \varphi$

(2) for all $\pi = s_0 s_1 s_2 \ldots \in Paths_{s_0}(\mathcal{M})$, for all i, $\mathcal{M}, s_i \models \varphi$

Finally, assume (2) and let's prove (1):
Take $\pi = s_0 s_1 s_2 \ldots \in Paths_{s_0}(\mathcal{M})$ to be any path starting in s_0 (such a path exists because every state has an outward transition).
Let i, $\pi' = s'_0 s'_1 s'_2 \ldots \in Paths_{s_i}(\mathcal{M})$ and j. (Note that $s'_0 = s_i$.)
We must prove $\mathcal{M}, s'_j \models \varphi$.
To this end, we apply (2) to the path $\pi'' = s''_0 s''_1 s''_2 \ldots \in Paths_{s_0}(\mathcal{M})$ defined as $s_0 s_1 \ldots s_{i-1} s'_0 s'_1 s'_2 \ldots$ and to the number $i + j$, obtaining $\mathcal{M}, s''_{i+j} \models \varphi$.
And since $s''_{i+j} = s'_j$, we obtain $\mathcal{M}, s'_j \models \varphi$, as desired.

We know:

(2) for all $\pi = s_0 s_1 s_2 \ldots \in \mathit{Paths}_{s_0}(\mathcal{M})$, for all i, $\mathcal{M}, s_i \models \varphi$

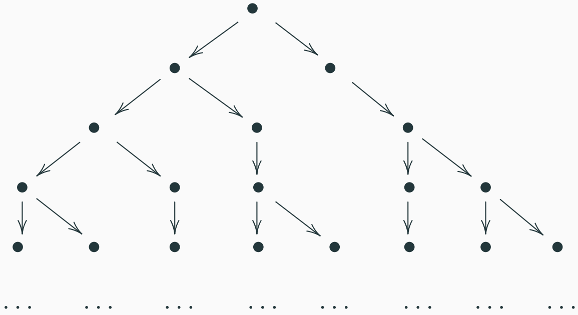


We show:

(1) there exists $\pi = s_0 s_1 s_2 \ldots \in \mathit{Paths}_{s_0}(\mathcal{M})$ such that, for all i,
for all $\pi' = s'_0 s'_1 s'_2 \ldots \in \mathit{Paths}_{s_i}(\mathcal{M})$, for all j, $\mathcal{M}, s'_j \models \varphi$

We know:

(2) for all $\pi = s_0 s_1 s_2 \ldots \in Paths_{s_0}(\mathcal{M})$, for all i, $\mathcal{M}, s_i \models \varphi$



We show:

(1) there exists $\pi = s_0 s_1 s_2 \ldots \in Paths_{s_0}(\mathcal{M})$ such that, for all i,
for all $\pi' = s'_0 s'_1 s'_2 \ldots \in Paths_{s_i}(\mathcal{M})$, for all j, $\mathcal{M}, s'_j \models \varphi$

This concludes our proof of (2) implies (1),
and also our entire proof of $\exists\Box\forall\Box\varphi \equiv \forall\Box\varphi$.

CTL Versus LTL

Differences between LTL and CTL

LTL is limited to questions of the form:

- For all paths, does the LTL formula φ hold?
- Does there exist a path on which the LTL formula φ holds?
  (by asking the opposite: whether $\neg\varphi$ holds on all paths)

Differences between LTL and CTL

LTL is limited to questions of the form:

- For all paths, does the LTL formula φ hold?
- Does there exist a path on which the LTL formula φ holds?
  (by asking the opposite: whether $\neg\varphi$ holds on all paths)

By contrast, CTL allows mixing in path quantifiers. For example:

$\forall\Box(a \rightarrow \exists\Box b)$
(For all paths, if a is true on some state then there is a path starting from that state on which b is always true.) This is not expressible in LTL.

Differences between LTL and CTL

LTL is limited to questions of the form:

- For all paths, does the LTL formula φ hold?
- Does there exist a path on which the LTL formula φ holds?
  (by asking the opposite: whether $\neg\varphi$ holds on all paths)

By contrast, CTL allows mixing in path quantifiers. For example:

$\forall\Box(a \to \exists\Box b)$
(For all paths, if a is true on some state then there is a path starting from that state on which b is always true.) This is not expressible in LTL. In particular, the LTL formula $\Box(a \to \Box b)$ does not express the same thing.

Differences between LTL and CTL

LTL is limited to questions of the form:

- For all paths, does the LTL formula φ hold?
- Does there exist a path on which the LTL formula φ holds?
  (by asking the opposite: whether $\neg\varphi$ holds on all paths)

By contrast, CTL allows mixing in path quantifiers. For example:

$\forall\Box\,(a \to \exists\Box\, b)$
(For all paths, if a is true on some state then there is a path starting from that state on which b is always true.) This is not expressible in LTL. In particular, the LTL formula $\Box(a \to \Box b)$ does not express the same thing. (As we'll see, the latter is equivalent to $\forall\Box\,(a \to \forall\Box\, b)$.)

Differences between LTL and CTL

On the other hand, some path properties are impossible to express in CTL.

Differences between LTL and CTL

On the other hand, some path properties are impossible to express in CTL. Intuitively, in CTL we cannot fix a path π and talk about it.

Differences between LTL and CTL

On the other hand, some path properties are impossible to express in CTL. Intuitively, in CTL we cannot fix a path π and talk about it.

The following formulas do not express the same thing:

- LTL: $(\Box\Diamond a) \rightarrow (\Box\Diamond b)$
  for all paths π, if a holds infinitely often on π, then b holds infinitely often on the same π
- CTL: $(\forall\Box\forall\Diamond a) \rightarrow (\forall\Box\forall\Diamond b)$
  if a holds infinitely often on all paths, then b holds infinitely often on all paths.

Differences between LTL and CTL

On the other hand, some path properties are impossible to express in CTL. Intuitively, in CTL we cannot fix a path π and talk about it.

The following formulas do not express the same thing:

- LTL: $(\Box\Diamond a) \to (\Box\Diamond b)$
  for all paths π, if a holds infinitely often on π, then b holds infinitely often on the same π
- CTL: $(\forall\Box\forall\Diamond\, a) \to (\forall\Box\forall\Diamond\, b)$
  if a holds infinitely often on all paths, then b holds infinitely often on all paths.

Neither CTL can express the first, nor LTL can express the second property.

Differences between LTL and CTL

Consider an LTS describing two parallel processes, where, for $i \in \{1, 2\}$:

n_i denotes "process i **n**ot in critical section"; r_i denotes "process i **r**equesting to enter critical section"; c_i denotes "process i in **c**ritical section"

Differences between LTL and CTL

Consider an LTS describing two parallel processes, where, for $i \in \{1, 2\}$:

n_i denotes "process i **n**ot in critical section"; r_i denotes "process i **r**equesting to enter critical section"; c_i denotes "process i in **c**ritical section"

Consider the following properties (which may or may not hold for that LTS):

- The safety property: Only one process at a time may execute critical section code.
- The liveness property: Whenever a process, say process 1, requests to enter its critical section, it will eventually be allowed to do so.
- The non-blocking property: A process, say process 1, can always request to enter its critical section

Differences between LTL and CTL

Consider an LTS describing two parallel processes, where, for $i \in \{1, 2\}$:

> n_i denotes "process i **n**ot in critical section"; r_i denotes "process i **r**equesting to enter critical section"; c_i denotes "process i in **c**ritical section"

Consider the following properties (which may or may not hold for that LTS):

- The safety property: Only one process at a time may execute critical section code. $\Box\,(\neg(c_1 \wedge c_2))$
- The liveness property: Whenever a process, say process 1, requests to enter its critical section, it will eventually be allowed to do so. $\Box\,(r_1 \rightarrow \Diamond c_1)$
- The non-blocking property: A process, say process 1, can always request to enter its critical section

Recall: The first two properties are expressible in LTL whereas the third is not.

Differences between LTL and CTL

Consider an LTS describing two parallel processes, where, for $i \in \{1, 2\}$:

> n_i denotes "process i **n**ot in critical section"; r_i denotes "process i **r**equesting to enter critical section"; c_i denotes "process i in **c**ritical section"

Consider the following properties (which may or may not hold for that LTS):

- The safety property: Only one process at a time may execute critical section code. $\Box(\neg(c_1 \wedge c_2))$
- The liveness property: Whenever a process, say process 1, requests to enter its critical section, it will eventually be allowed to do so. $\Box(r_1 \rightarrow \Diamond c_1)$
- The non-blocking property: A process, say process 1, can always request to enter its critical section

Recall: The first two properties are expressible in LTL whereas the third is not.
But can the third be expressed in CTL?

Differences between LTL and CTL

Consider an LTS describing two parallel processes, where, for $i \in \{1, 2\}$:

n_i denotes "process i **n**ot in critical section"; r_i denotes "process i **r**equesting to enter critical section"; c_i denotes "process i in **c**ritical section"

Consider the following properties (which may or may not hold for that LTS):

- The safety property: Only one process at a time may execute critical section code. $\Box(\neg(c_1 \wedge c_2))$
- The liveness property: Whenever a process, say process 1, requests to enter its critical section, it will eventually be allowed to do so. $\Box(r_1 \rightarrow \Diamond c_1)$
- The non-blocking property: A process, say process 1, can always request to enter its critical section, i.e.: For all states s reachable from the initial state such that $c_1 \notin L(s)$, there exists a state t reachable from s such that $r_1 \in L(t)$.

Recall: The first two properties are expressible in LTL whereas the third is not. But can the third be expressed in CTL?

Differences between LTL and CTL

Consider an LTS describing two parallel processes, where, for $i \in \{1, 2\}$:

n_i denotes "process i **n**ot in critical section"; r_i denotes "process i **r**equesting to enter critical section"; c_i denotes "process i in **c**ritical section"

Consider the following properties (which may or may not hold for that LTS):

- The safety property: Only one process at a time may execute critical section code. $\Box(\neg(c_1 \wedge c_2))$
- The liveness property: Whenever a process, say process 1, requests to enter its critical section, it will eventually be allowed to do so. $\Box(r_1 \rightarrow \Diamond c_1)$
- The non-blocking property: A process, say process 1, can always request to enter its critical section, i.e.: For all states s reachable from the initial state such that $c_1 \notin L(s)$, there exists a state t reachable from s such that $r_1 \in L(t)$.

Recall: The first two properties are expressible in LTL whereas the third is not. But can the third be expressed in CTL? Yes, it can: $\forall\Box(\neg c_1 \rightarrow \exists\Diamond r_1)$.

Differences between LTL and CTL

Consider an LTS describing two parallel processes, where, for $i \in \{1, 2\}$:

> n_i denotes "process i **n**ot in critical section"; r_i denotes "process i **r**equesting to enter critical section"; c_i denotes "process i in **c**ritical section"

Consider the following properties (which may or may not hold for that LTS):

- The safety property: Only one process at a time may execute critical section code. $\Box\,(\neg(c_1 \wedge c_2))$
- The liveness property: Whenever a process, say process 1, requests to enter its critical section, it will eventually be allowed to do so. $\Box\,(r_1 \rightarrow \Diamond c_1)$
- The non-blocking property: A process, say process 1, can always request to enter its critical section, i.e.: For all states s reachable from the initial state such that $c_1 \notin L(s)$, there exists a state t reachable from s such that $r_1 \in L(t)$.

Recall: The first two properties are expressible in LTL whereas the third is not.
But can the third be expressed in CTL? Yes, it can: $\forall\Box\,(\neg\, c_1 \rightarrow \exists\Diamond\, r_1)$.
How about the other two properties – can they be expressed in CTL?

Differences between LTL and CTL

Consider an LTS describing two parallel processes, where, for $i \in \{1, 2\}$:

n_i denotes "process i **n**ot in critical section"; r_i denotes "process i **r**equesting to enter critical section"; c_i denotes "process i in **c**ritical section"

Consider the following properties (which may or may not hold for that LTS):

- The safety property: Only one process at a time may execute critical section code. $\Box(\neg(c_1 \wedge c_2))$
- The liveness property: Whenever a process, say process 1, requests to enter its critical section, it will eventually be allowed to do so. $\Box(r_1 \rightarrow \Diamond c_1)$
- The non-blocking property: A process, say process 1, can always request to enter its critical section, i.e.: For all states s reachable from the initial state such that $c_1 \notin L(s)$, there exists a state t reachable from s such that $r_1 \in L(t)$.

Recall: The first two properties are expressible in LTL whereas the third is not.

But can the third be expressed in CTL? Yes, it can: $\forall\Box(\neg c_1 \rightarrow \exists\Diamond r_1)$.

How about the other two properties – can they be expressed in CTL?

Yes, they can.

For safety: the CTL $\forall\Box(\neg(c_1 \wedge c_2))$ is equivalent to the LTL $\Box(\neg(c_1 \wedge c_2))$.

Differences between LTL and CTL

Consider an LTS describing two parallel processes, where, for $i \in \{1, 2\}$:

n_i denotes "process i **n**ot in critical section"; r_i denotes "process i **r**equesting to enter critical section"; c_i denotes "process i in **c**ritical section"

Consider the following properties (which may or may not hold for that LTS):

- The safety property: Only one process at a time may execute critical section code. $\Box(\neg(c_1 \wedge c_2))$
- The liveness property: Whenever a process, say process 1, requests to enter its critical section, it will eventually be allowed to do so. $\Box(r_1 \to \Diamond c_1)$
- The non-blocking property: A process, say process 1, can always request to enter its critical section, i.e.: For all states s reachable from the initial state such that $c_1 \notin L(s)$, there exists a state t reachable from s such that $r_1 \in L(t)$.

Recall: The first two properties are expressible in LTL whereas the third is not.
But can the third be expressed in CTL? Yes, it can: $\forall\Box(\neg c_1 \to \exists\Diamond r_1)$.
How about the other two properties – can they be expressed in CTL?
Yes, they can.

For safety: the CTL $\forall\Box(\neg(c_1 \wedge c_2))$ is equivalent to the LTL $\Box(\neg(c_1 \wedge c_2))$.
For liveness: the CTL $\forall\Box(r_1 \to \forall\Diamond c_1)$ is equivalent to the LTL $\Box(r_1 \to \Diamond c_1)$.

Equivalence Relation between CTL and LTL Formulas

Defined similarly to CTL formula equivalence.

Let φ be a CTL formula, and let ψ be an LTL formula.

Equivalence Relation between CTL and LTL Formulas

Defined similarly to CTL formula equivalence.

Let φ be a CTL formula, and let ψ be an LTL formula. φ and ψ are said to be equivalent if they are satisfied by (i.e., hold for) exactly the same LTSs in the same states

Equivalence Relation between CTL and LTL Formulas

Defined similarly to CTL formula equivalence.

Let φ be a CTL formula, and let ψ be an LTL formula. φ and ψ are said to be equivalent if they are satisfied by (i.e., hold for) exactly the same LTSs in the same states: Given any LTS $\mathcal{M} = (S, \rightarrow, L)$ and any $s \in S$, we have that

$$\mathcal{M}, s \models \varphi \text{ (in CTL) iff } \mathcal{M}, s \models \psi \text{ (in LTL)}.$$

Proving CTL–LTL Formula Equivalence

Let's prove that the CTL formula $\forall\Box\,(\neg(c_1 \wedge c_2))$ is equivalent to (i.e., is satisfied by the same LTSs in the same states as) the LTL formula $\Box\,(\neg(c_1 \wedge c_2))$.

Proving CTL–LTL Formula Equivalence

Let's prove that the CTL formula $\forall\Box\,(\neg(c_1 \wedge c_2))$ is equivalent to (i.e., is satisfied by the same LTSs in the same states as) the LTL formula $\Box\,(\neg(c_1 \wedge c_2))$.

Let $\mathcal{M} = (S, \to, L)$ be an LTS and $s_0 \in S$. We have the following chain of equivalent statements:

$\mathcal{M}, s_0 \models \forall\Box\,(\neg(c_1 \wedge c_2))$ in CTL

Proving CTL–LTL Formula Equivalence

Let's prove that the CTL formula $\forall\Box\,(\neg(c_1 \wedge c_2))$ is equivalent to (i.e., is satisfied by the same LTSs in the same states as) the LTL formula $\Box\,(\neg(c_1 \wedge c_2))$.

Let $\mathcal{M} = (S, \rightarrow, L)$ be an LTS and $s_0 \in S$. We have the following chain of equivalent statements:

$\mathcal{M}, s_0 \models \forall\Box\,(\neg(c_1 \wedge c_2))$ in CTL

iff (by the CTL semantics of $\forall\Box$)

for all $\pi = s_0 s_1 s_2 \ldots \in Paths_{s_0}(\mathcal{M})$, for all i, $\mathcal{M}, s_i \models \neg(c_1 \wedge c_2)$ in CTL

Proving CTL–LTL Formula Equivalence

Let's prove that the CTL formula $\forall\Box\,(\neg(c_1 \wedge c_2))$ is equivalent to (i.e., is satisfied by the same LTSs in the same states as) the LTL formula $\Box\,(\neg(c_1 \wedge c_2))$.

Let $\mathcal{M} = (S, \rightarrow, L)$ be an LTS and $s_0 \in S$. We have the following chain of equivalent statements:

$\mathcal{M}, s_0 \models \forall\Box\,(\neg(c_1 \wedge c_2))$ in CTL

  iff (by the CTL semantics of $\forall\Box$)

for all $\pi = s_0 s_1 s_2 \ldots \in Paths_{s_0}(\mathcal{M})$, for all i, $\mathcal{M}, s_i \models \neg(c_1 \wedge c_2)$ in CTL

  iff (by the CTL semantics of $\neg$, $\wedge$ and atoms)

for all $\pi = s_0 s_1 s_2 \ldots \in Paths_{s_0}(\mathcal{M})$, for all i, not $[c_1 \in L(s_i)$ and $c_2 \in L(s_i)]$

Proving CTL–LTL Formula Equivalence

Let's prove that the CTL formula $\forall\Box\,(\neg(c_1 \wedge c_2))$ is equivalent to (i.e., is satisfied by the same LTSs in the same states as) the LTL formula $\Box\,(\neg(c_1 \wedge c_2))$.

Let $\mathcal{M} = (S, \rightarrow, L)$ be an LTS and $s_0 \in S$. We have the following chain of equivalent statements:

$\mathcal{M}, s_0 \models \forall\Box\,(\neg(c_1 \wedge c_2))$ in CTL

  iff (by the CTL semantics of $\forall\Box$)

for all $\pi = s_0 s_1 s_2 \ldots \in Paths_{s_0}(\mathcal{M})$, for all i, $\mathcal{M}, s_i \models \neg(c_1 \wedge c_2)$ in CTL

  iff (by the CTL semantics of $\neg$, $\wedge$ and atoms)

for all $\pi = s_0 s_1 s_2 \ldots \in Paths_{s_0}(\mathcal{M})$, for all i, not $[c_1 \in L(s_i)$ and $c_2 \in L(s_i)]$

  iff (by the LTL semantics of $\neg$, $\wedge$ and atoms, on sequences)

for all $\pi = s_0 s_1 s_2 \ldots \in Paths_{s_0}(\mathcal{M})$, for all i, $\pi^i \models_L \neg(c_1 \wedge c_2)$ in LTL

Proving CTL–LTL Formula Equivalence

Let's prove that the CTL formula $\forall\Box\,(\neg(c_1 \wedge c_2))$ is equivalent to (i.e., is satisfied by the same LTSs in the same states as) the LTL formula $\Box\,(\neg(c_1 \wedge c_2))$.

Let $\mathcal{M} = (S, \rightarrow, L)$ be an LTS and $s_0 \in S$. We have the following chain of equivalent statements:

$\mathcal{M}, s_0 \models \forall\Box\,(\neg(c_1 \wedge c_2))$ in CTL

iff (by the CTL semantics of $\forall\Box$)

for all $\pi = s_0 s_1 s_2 \ldots \in \mathit{Paths}_{s_0}(\mathcal{M})$, for all i, $\mathcal{M}, s_i \models \neg(c_1 \wedge c_2)$ in CTL

iff (by the CTL semantics of $\neg$, $\wedge$ and atoms)

for all $\pi = s_0 s_1 s_2 \ldots \in \mathit{Paths}_{s_0}(\mathcal{M})$, for all i, not $[c_1 \in L(s_i)$ and $c_2 \in L(s_i)]$

iff (by the LTL semantics of $\neg$, $\wedge$ and atoms, on sequences)

for all $\pi = s_0 s_1 s_2 \ldots \in \mathit{Paths}_{s_0}(\mathcal{M})$, for all i, $\pi^i \models_L \neg(c_1 \wedge c_2)$ in LTL

Remember: π^i is the i'th suffix of π.

Proving CTL–LTL Formula Equivalence

Let's prove that the CTL formula $\forall\Box\,(\neg(c_1 \wedge c_2))$ is equivalent to (i.e., is satisfied by the same LTSs in the same states as) the LTL formula $\Box\,(\neg(c_1 \wedge c_2))$.

Let $\mathcal{M} = (S, \rightarrow, L)$ be an LTS and $s_0 \in S$. We have the following chain of equivalent statements:

$\mathcal{M}, s_0 \models \forall\Box\,(\neg(c_1 \wedge c_2))$ in CTL
  iff (by the CTL semantics of $\forall\Box$)
for all $\pi = s_0 s_1 s_2 \ldots \in Paths_{s_0}(\mathcal{M})$, for all i, $\mathcal{M}, s_i \models \neg(c_1 \wedge c_2)$ in CTL
  iff (by the CTL semantics of $\neg$, $\wedge$ and atoms)
for all $\pi = s_0 s_1 s_2 \ldots \in Paths_{s_0}(\mathcal{M})$, for all i, not $[c_1 \in L(s_i)$ and $c_2 \in L(s_i)]$
  iff (by the LTL semantics of $\neg$, $\wedge$ and atoms, on sequences)
for all $\pi = s_0 s_1 s_2 \ldots \in Paths_{s_0}(\mathcal{M})$, for all i, $\pi^i \models_L \neg(c_1 \wedge c_2)$ in LTL
  iff (by the LTL semantics of $\Box$, on sequences)
for all $\pi = s_0 s_1 s_2 \ldots \in Paths_{s_0}(\mathcal{M})$, $\pi \models_L \Box\,(\neg(c_1 \wedge c_2))$ in LTL

Proving CTL–LTL Formula Equivalence

Let's prove that the CTL formula $\forall\Box\,(\neg(c_1 \wedge c_2))$ is equivalent to (i.e., is satisfied by the same LTSs in the same states as) the LTL formula $\Box\,(\neg(c_1 \wedge c_2))$.

Let $\mathcal{M} = (S, \rightarrow, L)$ be an LTS and $s_0 \in S$. We have the following chain of equivalent statements:

$\mathcal{M}, s_0 \models \forall\Box\,(\neg(c_1 \wedge c_2))$ in CTL

iff (by the CTL semantics of $\forall\Box$)

for all $\pi = s_0 s_1 s_2 \ldots \in \mathit{Paths}_{s_0}(\mathcal{M})$, for all i, $\mathcal{M}, s_i \models \neg(c_1 \wedge c_2)$ in CTL

iff (by the CTL semantics of $\neg$, $\wedge$ and atoms)

for all $\pi = s_0 s_1 s_2 \ldots \in \mathit{Paths}_{s_0}(\mathcal{M})$, for all i, not $[c_1 \in L(s_i)$ and $c_2 \in L(s_i)]$

iff (by the LTL semantics of $\neg$, $\wedge$ and atoms, on sequences)

for all $\pi = s_0 s_1 s_2 \ldots \in \mathit{Paths}_{s_0}(\mathcal{M})$, for all i, $\pi^i \models_L \neg(c_1 \wedge c_2)$ in LTL

iff (by the LTL semantics of $\Box$, on sequences)

for all $\pi = s_0 s_1 s_2 \ldots \in \mathit{Paths}_{s_0}(\mathcal{M})$, $\pi \models_L \Box\,(\neg(c_1 \wedge c_2))$ in LTL

iff (by the LTL semantics on LTSs)

$\mathcal{M}, s_0 \models \Box\,(\neg(c_1 \wedge c_2))$ in LTL

Proving CTL–LTL Formula Equivalence

Let's prove that the CTL formula $\forall\Box\,(\neg(c_1 \wedge c_2))$ is equivalent to (i.e., is satisfied by the same LTSs in the same states as) the LTL formula $\Box\,(\neg(c_1 \wedge c_2))$.

Let $\mathcal{M} = (S, \rightarrow, L)$ be an LTS and $s_0 \in S$. We have the following chain of equivalent statements:

$\mathcal{M}, s_0 \models \forall\Box\,(\neg(c_1 \wedge c_2))$ in CTL
  iff (by the CTL semantics of $\forall\Box$)
for all $\pi = s_0 s_1 s_2 \ldots \in Paths_{s_0}(\mathcal{M})$, for all i, $\mathcal{M}, s_i \models \neg(c_1 \wedge c_2)$ in CTL
  iff (by the CTL semantics of $\neg$, $\wedge$ and atoms)
for all $\pi = s_0 s_1 s_2 \ldots \in Paths_{s_0}(\mathcal{M})$, for all i, not $[c_1 \in L(s_i)$ and $c_2 \in L(s_i)]$
  iff (by the LTL semantics of $\neg$, $\wedge$ and atoms, on sequences)
for all $\pi = s_0 s_1 s_2 \ldots \in Paths_{s_0}(\mathcal{M})$, for all i, $\pi^i \models_L \neg(c_1 \wedge c_2)$ in LTL
  iff (by the LTL semantics of $\Box$, on sequences)
for all $\pi = s_0 s_1 s_2 \ldots \in Paths_{s_0}(\mathcal{M})$, $\pi \models_L \Box\,(\neg(c_1 \wedge c_2))$ in LTL
  iff (by the LTL semantics on LTSs)
$\mathcal{M}, s_0 \models \Box\,(\neg(c_1 \wedge c_2))$ in LTL

Proving CTL–LTL Formula Equivalence

Now let's prove that the CTL formula $\forall\Box\,(r_1 \rightarrow \forall\Diamond\, c_1)$ is equivalent to (i.e., is satisfied by the same LTSs in the same states as) the LTL formula $\Box\,(r_1 \rightarrow \Diamond\, c_1)$.

Proving CTL–LTL Formula Equivalence

Now let's prove that the CTL formula $\forall\Box\,(r_1 \to \forall\Diamond\,c_1)$ is equivalent to (i.e., is satisfied by the same LTSs in the same states as) the LTL formula $\Box\,(r_1 \to \Diamond\,c_1)$.

Let $\mathcal{M} = (S, \to, L)$ be an LTS and $s_0 \in S$.

Proving CTL–LTL Formula Equivalence

Now let's prove that the CTL formula $\forall\Box\,(r_1 \to \forall\Diamond\, c_1)$ is equivalent to (i.e., is satisfied by the same LTSs in the same states as) the LTL formula $\Box\,(r_1 \to \Diamond\, c_1)$.

Let $\mathcal{M} = (S, \to, L)$ be an LTS and $s_0 \in S$.

On the one hand, we have the following chain of equivalent statements:

$\mathcal{M}, s_0 \models \forall\Box\,(r_1 \to \forall\Diamond\, c_1)$ in CTL

Proving CTL–LTL Formula Equivalence

Now let's prove that the CTL formula $\forall\Box\,(r_1 \rightarrow \forall\Diamond\, c_1)$ is equivalent to (i.e., is satisfied by the same LTSs in the same states as) the LTL formula $\Box\,(r_1 \rightarrow \Diamond\, c_1)$.

Let $\mathcal{M} = (S, \rightarrow, L)$ be an LTS and $s_0 \in S$.

On the one hand, we have the following chain of equivalent statements:

$\mathcal{M}, s_0 \models \forall\Box\,(r_1 \rightarrow \forall\Diamond\, c_1)$ in CTL
  iff (by the CTL semantics of $\forall\Box$)
for all $\pi = s_0 s_1 s_2 \ldots \in Paths_{s_0}(\mathcal{M})$, for all i, $\mathcal{M}, s_i \models r_1 \rightarrow \forall\Diamond\, c_1$ in CTL

Proving CTL–LTL Formula Equivalence

Now let's prove that the CTL formula $\forall\Box\,(r_1 \rightarrow \forall\Diamond\, c_1)$ is equivalent to (i.e., is satisfied by the same LTSs in the same states as) the LTL formula $\Box\,(r_1 \rightarrow \Diamond c_1)$.

Let $\mathcal{M} = (S, \rightarrow, L)$ be an LTS and $s_0 \in S$.

On the one hand, we have the following chain of equivalent statements:

$\mathcal{M}, s_0 \models \forall\Box\,(r_1 \rightarrow \forall\Diamond\, c_1)$ in CTL
  iff (by the CTL semantics of $\forall\Box$)
for all $\pi = s_0 s_1 s_2 \ldots \in Paths_{s_0}(\mathcal{M})$, for all i, $\mathcal{M}, s_i \models r_1 \rightarrow \forall\Diamond\, c_1$ in CTL
  iff (by the CTL semantics of $\rightarrow$ and atoms)
for all $\pi = s_0 s_1 s_2 \ldots \in Paths_{s_0}(\mathcal{M})$, for all i,
    if $r_1 \in L(s_i)$ then $\mathcal{M}, s_i \models \forall\Diamond\, c_1$ in CTL

Proving CTL–LTL Formula Equivalence

Now let's prove that the CTL formula $\forall\Box\,(r_1 \to \forall\Diamond\,c_1)$ is equivalent to (i.e., is satisfied by the same LTSs in the same states as) the LTL formula $\Box\,(r_1 \to \Diamond\,c_1)$.

Let $\mathcal{M} = (S, \to, L)$ be an LTS and $s_0 \in S$.

On the one hand, we have the following chain of equivalent statements:

$\mathcal{M}, s_0 \models \forall\Box\,(r_1 \to \forall\Diamond\,c_1)$ in CTL
  iff (by the CTL semantics of $\forall\Box$)
for all $\pi = s_0 s_1 s_2 \ldots \in Paths_{s_0}(\mathcal{M})$, for all i, $\mathcal{M}, s_i \models r_1 \to \forall\Diamond\,c_1$ in CTL
  iff (by the CTL semantics of $\to$ and atoms)
for all $\pi = s_0 s_1 s_2 \ldots \in Paths_{s_0}(\mathcal{M})$, for all i,
    if $r_1 \in L(s_i)$ then $\mathcal{M}, s_i \models \forall\Diamond\,c_1$ in CTL
  iff (by the CTL semantics of $\forall\Diamond$ and atoms)
for all $\pi = s_0 s_1 s_2 \ldots \in Paths_{s_0}(\mathcal{M})$, for all i,
    if $r_1 \in L(s_i)$ then
    for all $\pi' = s'_0 s'_1 s'_2 \ldots \in Paths_{s_i}(\mathcal{M})$, there exists j such that $c_1 \in L(s'_j)$

Proving CTL–LTL Formula Equivalence

Now let's prove that the CTL formula $\forall\Box\,(r_1 \to \forall\Diamond\, c_1)$ is equivalent to (i.e., is satisfied by the same LTSs in the same states as) the LTL formula $\Box\,(r_1 \to \Diamond c_1)$.

Let $\mathcal{M} = (S, \to, L)$ be an LTS and $s_0 \in S$.

On the one hand, we have the following chain of equivalent statements:

$\mathcal{M}, s_0 \models \forall\Box\,(r_1 \to \forall\Diamond\, c_1)$ in CTL
  iff (by the CTL semantics of $\forall\Box$)
for all $\pi = s_0 s_1 s_2 \ldots \in \mathit{Paths}_{s_0}(\mathcal{M})$, for all i, $\mathcal{M}, s_i \models r_1 \to \forall\Diamond\, c_1$ in CTL
  iff (by the CTL semantics of $\to$ and atoms)
for all $\pi = s_0 s_1 s_2 \ldots \in \mathit{Paths}_{s_0}(\mathcal{M})$, for all i,
    if $r_1 \in L(s_i)$ then $\mathcal{M}, s_i \models \forall\Diamond\, c_1$ in CTL
  iff (by the CTL semantics of $\forall\Diamond$ and atoms)
for all $\pi = s_0 s_1 s_2 \ldots \in \mathit{Paths}_{s_0}(\mathcal{M})$, for all i,
    if $r_1 \in L(s_i)$ then
    for all $\pi' = s'_0 s'_1 s'_2 \ldots \in \mathit{Paths}_{s_i}(\mathcal{M})$, there exists j such that $c_1 \in L(s'_j)$

Proving CTL–LTL Formula Equivalence

Let's prove that the CTL formula $\forall\Box\,(r_1 \rightarrow \forall\Diamond\, c_1)$ is equivalent to (i.e., is satisfied by the same LTSs in the same states as) the LTL formula $\Box\,(r_1 \rightarrow \Diamond\, c_1)$.

Let $\mathcal{M} = (S, \rightarrow, L)$ be an LTS and $s_0 \in S$.

Proving CTL–LTL Formula Equivalence

Let's prove that the CTL formula $\forall\Box\,(r_1 \to \forall\Diamond\, c_1)$ is equivalent to (i.e., is satisfied by the same LTSs in the same states as) the LTL formula $\Box\,(r_1 \to \Diamond c_1)$.

Let $\mathcal{M} = (S, \to, L)$ be an LTS and $s_0 \in S$.

On the other hand, we have the following chain of equivalent statements:

$\mathcal{M}, s_0 \models \Box\,(r_1 \to \Diamond c_1)$ in LTL

Proving CTL–LTL Formula Equivalence

Let's prove that the CTL formula $\forall\Box\,(r_1 \rightarrow \forall\Diamond\, c_1)$ is equivalent to (i.e., is satisfied by the same LTSs in the same states as) the LTL formula $\Box\,(r_1 \rightarrow \Diamond c_1)$.

Let $\mathcal{M} = (S, \rightarrow, L)$ be an LTS and $s_0 \in S$.

On the other hand, we have the following chain of equivalent statements:

$\mathcal{M}, s_0 \models \Box\,(r_1 \rightarrow \Diamond c_1)$ in LTL
  iff (by the LTL semantics on LTSs)
for all $\pi = s_0 s_1 s_2 \ldots \in Paths_{s_0}(\mathcal{M})$, $\pi \models_L \Box\,(r_1 \rightarrow \Diamond c_1)$ in LTL

Proving CTL–LTL Formula Equivalence

Let's prove that the CTL formula $\forall\Box\,(r_1 \rightarrow \forall\Diamond\, c_1)$ is equivalent to (i.e., is satisfied by the same LTSs in the same states as) the LTL formula $\Box\,(r_1 \rightarrow \Diamond c_1)$.

Let $\mathcal{M} = (S, \rightarrow, L)$ be an LTS and $s_0 \in S$.

On the other hand, we have the following chain of equivalent statements:

$\mathcal{M}, s_0 \models \Box\,(r_1 \rightarrow \Diamond c_1)$ in LTL
  iff (by the LTL semantics on LTSs)
for all $\pi = s_0 s_1 s_2 \ldots \in Paths_{s_0}(\mathcal{M})$, $\pi \models_L \Box\,(r_1 \rightarrow \Diamond c_1)$ in LTL
  iff (by the LTL semantics of $\Box$, on sequences)
for all $\pi = s_0 s_1 s_2 \ldots \in Paths_{s_0}(\mathcal{M})$, for all i, $\pi^i \models_L r_1 \rightarrow \Diamond c_1$ in LTL

Proving CTL–LTL Formula Equivalence

Let's prove that the CTL formula $\forall\Box\,(r_1 \rightarrow \forall\Diamond\, c_1)$ is equivalent to (i.e., is satisfied by the same LTSs in the same states as) the LTL formula $\Box\,(r_1 \rightarrow \Diamond c_1)$.

Let $\mathcal{M} = (S, \rightarrow, L)$ be an LTS and $s_0 \in S$.

On the other hand, we have the following chain of equivalent statements:

$\mathcal{M}, s_0 \models \Box\,(r_1 \rightarrow \Diamond c_1)$ in LTL
  iff (by the LTL semantics on LTSs)
for all $\pi = s_0 s_1 s_2 \ldots \in Paths_{s_0}(\mathcal{M})$, $\pi \models_L \Box\,(r_1 \rightarrow \Diamond c_1)$ in LTL
  iff (by the LTL semantics of $\Box$, on sequences)
for all $\pi = s_0 s_1 s_2 \ldots \in Paths_{s_0}(\mathcal{M})$, for all i, $\pi^i \models_L r_1 \rightarrow \Diamond c_1$ in LTL
  iff (by the LTL semantics of $\rightarrow$ and atoms, on sequences)
for all $\pi = s_0 s_1 s_2 \ldots \in Paths_{s_0}(\mathcal{M})$, for all i,
    if $r_1 \in L(s_i)$ then $\pi^i \models_L \Diamond c_1$ in LTL

Proving CTL–LTL Formula Equivalence

Let's prove that the CTL formula $\forall\Box\,(r_1 \rightarrow \forall\Diamond\, c_1)$ is equivalent to (i.e., is satisfied by the same LTSs in the same states as) the LTL formula $\Box\,(r_1 \rightarrow \Diamond c_1)$.

Let $\mathcal{M} = (S, \rightarrow, L)$ be an LTS and $s_0 \in S$.

On the other hand, we have the following chain of equivalent statements:

$\mathcal{M}, s_0 \models \Box\,(r_1 \rightarrow \Diamond c_1)$ in LTL
  iff (by the LTL semantics on LTSs)
for all $\pi = s_0 s_1 s_2 \ldots \in Paths_{s_0}(\mathcal{M})$, $\pi \models_L \Box\,(r_1 \rightarrow \Diamond c_1)$ in LTL
  iff (by the LTL semantics of $\Box$, on sequences)
for all $\pi = s_0 s_1 s_2 \ldots \in Paths_{s_0}(\mathcal{M})$, for all i, $\pi^i \models_L r_1 \rightarrow \Diamond c_1$ in LTL
  iff (by the LTL semantics of $\rightarrow$ and atoms, on sequences)
for all $\pi = s_0 s_1 s_2 \ldots \in Paths_{s_0}(\mathcal{M})$, for all i,
    if $r_1 \in L(s_i)$ then $\pi^i \models_L \Diamond c_1$ in LTL
  iff (by the LTL semantics of $\Diamond$ and atoms, on sequences)
for all $\pi = s_0 s_1 s_2 \ldots \in Paths_{s_0}(\mathcal{M})$, for all i,
    if $r_1 \in L(s_i)$ then
    there exists j such that $c_1 \in L(s_{i+j})$

Proving CTL–LTL Formula Equivalence

Let's prove that the CTL formula $\forall\Box\,(r_1 \to \forall\Diamond\, c_1)$ is equivalent to (i.e., is satisfied by the same LTSs in the same states as) the LTL formula $\Box\,(r_1 \to \Diamond c_1)$.

Let $\mathcal{M} = (S, \to, L)$ be an LTS and $s_0 \in S$.

On the other hand, we have the following chain of equivalent statements:

$\mathcal{M}, s_0 \models \Box\,(r_1 \to \Diamond c_1)$ in LTL

iff (by the LTL semantics on LTSs)

for all $\pi = s_0 s_1 s_2 \ldots \in Paths_{s_0}(\mathcal{M})$, $\pi \models_L \Box\,(r_1 \to \Diamond c_1)$ in LTL

iff (by the LTL semantics of $\Box$, on sequences)

for all $\pi = s_0 s_1 s_2 \ldots \in Paths_{s_0}(\mathcal{M})$, for all i, $\pi^i \models_L r_1 \to \Diamond c_1$ in LTL

iff (by the LTL semantics of $\to$ and atoms, on sequences)

for all $\pi = s_0 s_1 s_2 \ldots \in Paths_{s_0}(\mathcal{M})$, for all i,
if $r_1 \in L(s_i)$ then $\pi^i \models_L \Diamond c_1$ in LTL

iff (by the LTL semantics of $\Diamond$ and atoms, on sequences)

for all $\pi = s_0 s_1 s_2 \ldots \in Paths_{s_0}(\mathcal{M})$, for all i,
if $r_1 \in L(s_i)$ then
there exists j such that $c_1 \in L(s_{i+j})$

Proving CTL–LTL Formula Equivalence

Let's prove that the CTL formula $\forall\Box\,(r_1 \to \forall\Diamond\, c_1)$ is equivalent to (i.e., is satisfied by the same LTSs in the same states as) the LTL formula $\Box\,(r_1 \to \Diamond c_1)$.

So, given an LTS $\mathcal{M} = (S, \to, L)$ and $s_0 \in S$, we are left the prove that the following statements are equivalent:

(1) for all $\pi = s_0 s_1 s_2 \ldots \in Paths_{s_0}(\mathcal{M})$, for all i,
if $r_1 \in L(s_i)$ then
for all $\pi' = s'_0 s'_1 s'_2 \ldots \in Paths_{s_i}(\mathcal{M})$, there exists j such that $c_1 \in L(s'_j)$

(2) for all $\pi = s_0 s_1 s_2 \ldots \in Paths_{s_0}(\mathcal{M})$, for all i,
if $r_1 \in L(s_i)$ then
there exists j such that $c_1 \in L(s_{i+j})$

Proving CTL–LTL Formula Equivalence

Let's prove that the CTL formula $\forall\Box\,(r_1 \to \forall\Diamond\, c_1)$ is equivalent to (i.e., is satisfied by the same LTSs in the same states as) the LTL formula $\Box\,(r_1 \to \Diamond c_1)$.

So, given an LTS $\mathcal{M} = (S, \to, L)$ and $s_0 \in S$, we are left the prove that the following statements are equivalent:

(1) for all $\pi = s_0 s_1 s_2 \ldots \in Paths_{s_0}(\mathcal{M})$, for all i,
 if $r_1 \in L(s_i)$ then
 for all $\pi' = s'_0 s'_1 s'_2 \ldots \in Paths_{s_i}(\mathcal{M})$, there exists j such that $c_1 \in L(s'_j)$

(2) for all $\pi = s_0 s_1 s_2 \ldots \in Paths_{s_0}(\mathcal{M})$, for all i,
 if $r_1 \in L(s_i)$ then
 there exists j such that $c_1 \in L(s_{i+j})$

First, assume (1) and let's prove (2):

Proving CTL–LTL Formula Equivalence

Let's prove that the CTL formula $\forall\Box\,(r_1 \to \forall\Diamond\, c_1)$ is equivalent to (i.e., is satisfied by the same LTSs in the same states as) the LTL formula $\Box\,(r_1 \to \Diamond c_1)$.

So, given an LTS $\mathcal{M} = (S, \to, L)$ and $s_0 \in S$, we are left the prove that the following statements are equivalent:

(1) for all $\pi = s_0 s_1 s_2 \ldots \in Paths_{s_0}(\mathcal{M})$, for all i,
if $r_1 \in L(s_i)$ then
for all $\pi' = s'_0 s'_1 s'_2 \ldots \in Paths_{s_i}(\mathcal{M})$, there exists j such that $c_1 \in L(s'_j)$

(2) for all $\pi = s_0 s_1 s_2 \ldots \in Paths_{s_0}(\mathcal{M})$, for all i,
if $r_1 \in L(s_i)$ then
there exists j such that $c_1 \in L(s_{i+j})$

First, assume (1) and let's prove (2):
Let $\pi = s_0 s_1 s_2 \ldots \in Paths_{s_0}(\mathcal{M})$ and i, and assume $r_1 \in L(s_i)$.

Proving CTL–LTL Formula Equivalence

Let's prove that the CTL formula $\forall\Box\,(r_1 \rightarrow \forall\Diamond\, c_1)$ is equivalent to (i.e., is satisfied by the same LTSs in the same states as) the LTL formula $\Box\,(r_1 \rightarrow \Diamond c_1)$.

So, given an LTS $\mathcal{M} = (S, \rightarrow, L)$ and $s_0 \in S$, we are left the prove that the following statements are equivalent:

(1) for all $\pi = s_0 s_1 s_2 \ldots \in Paths_{s_0}(\mathcal{M})$, for all i,
if $r_1 \in L(s_i)$ then
for all $\pi' = s'_0 s'_1 s'_2 \ldots \in Paths_{s_i}(\mathcal{M})$, there exists j such that $c_1 \in L(s'_j)$

(2) for all $\pi = s_0 s_1 s_2 \ldots \in Paths_{s_0}(\mathcal{M})$, for all i,
if $r_1 \in L(s_i)$ then
there exists j such that $c_1 \in L(s_{i+j})$

First, assume (1) and let's prove (2):
Let $\pi = s_0 s_1 s_2 \ldots \in Paths_{s_0}(\mathcal{M})$ and i, and assume $r_1 \in L(s_i)$.
Take $\pi' = s'_0 s'_1 s'_2 \ldots$ to be π^i.

Proving CTL–LTL Formula Equivalence

Let's prove that the CTL formula $\forall\Box\,(r_1 \to \forall\Diamond\, c_1)$ is equivalent to (i.e., is satisfied by the same LTSs in the same states as) the LTL formula $\Box\,(r_1 \to \Diamond c_1)$.

So, given an LTS $\mathcal{M} = (S, \to, L)$ and $s_0 \in S$, we are left the prove that the following statements are equivalent:

(1) for all $\pi = s_0 s_1 s_2 \ldots \in Paths_{s_0}(\mathcal{M})$, for all i,
 if $r_1 \in L(s_i)$ then
 for all $\pi' = s'_0 s'_1 s'_2 \ldots \in Paths_{s_i}(\mathcal{M})$, there exists j such that $c_1 \in L(s'_j)$

(2) for all $\pi = s_0 s_1 s_2 \ldots \in Paths_{s_0}(\mathcal{M})$, for all i,
 if $r_1 \in L(s_i)$ then
 there exists j such that $c_1 \in L(s_{i+j})$

First, assume (1) and let's prove (2):
Let $\pi = s_0 s_1 s_2 \ldots \in Paths_{s_0}(\mathcal{M})$ and i, and assume $r_1 \in L(s_i)$.
Take $\pi' = s'_0 s'_1 s'_2 \ldots$ to be π^i. Then $\pi' \in Paths_{s_i}(\mathcal{M})$ and, for all j, $s'_j = s_{i+j}$.

Proving CTL–LTL Formula Equivalence

Let's prove that the CTL formula $\forall\Box\,(r_1 \to \forall\Diamond\, c_1)$ is equivalent to (i.e., is satisfied by the same LTSs in the same states as) the LTL formula $\Box\,(r_1 \to \Diamond c_1)$.

So, given an LTS $\mathcal{M} = (S, \to, L)$ and $s_0 \in S$, we are left the prove that the following statements are equivalent:

(1) for all $\pi = s_0 s_1 s_2 \ldots \in \mathit{Paths}_{s_0}(\mathcal{M})$, for all i,
 if $r_1 \in L(s_i)$ then
 for all $\pi' = s'_0 s'_1 s'_2 \ldots \in \mathit{Paths}_{s_i}(\mathcal{M})$, there exists j such that $c_1 \in L(s'_j)$

(2) for all $\pi = s_0 s_1 s_2 \ldots \in \mathit{Paths}_{s_0}(\mathcal{M})$, for all i,
 if $r_1 \in L(s_i)$ then
 there exists j such that $c_1 \in L(s_{i+j})$

First, assume (1) and let's prove (2):
Let $\pi = s_0 s_1 s_2 \ldots \in \mathit{Paths}_{s_0}(\mathcal{M})$ and i, and assume $r_1 \in L(s_i)$.
Take $\pi' = s'_0 s'_1 s'_2 \ldots$ to be π^i. Then $\pi' \in \mathit{Paths}_{s_i}(\mathcal{M})$ and, for all j, $s'_j = s_{i+j}$.
By (1) applied to π, i and π', we obtain j such that $c_1 \in L(s'_j)$,

Proving CTL–LTL Formula Equivalence

Let's prove that the CTL formula $\forall\Box\,(r_1 \to \forall\Diamond\, c_1)$ is equivalent to (i.e., is satisfied by the same LTSs in the same states as) the LTL formula $\Box\,(r_1 \to \Diamond c_1)$.

So, given an LTS $\mathcal{M} = (S, \to, L)$ and $s_0 \in S$, we are left the prove that the following statements are equivalent:

(1) for all $\pi = s_0 s_1 s_2 \ldots \in Paths_{s_0}(\mathcal{M})$, for all i,
 if $r_1 \in L(s_i)$ then
 for all $\pi' = s'_0 s'_1 s'_2 \ldots \in Paths_{s_i}(\mathcal{M})$, there exists j such that $c_1 \in L(s'_j)$

(2) for all $\pi = s_0 s_1 s_2 \ldots \in Paths_{s_0}(\mathcal{M})$, for all i,
 if $r_1 \in L(s_i)$ then
 there exists j such that $c_1 \in L(s_{i+j})$

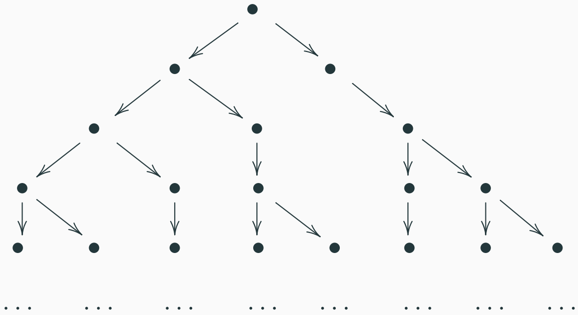
First, assume (1) and let's prove (2):
Let $\pi = s_0 s_1 s_2 \ldots \in Paths_{s_0}(\mathcal{M})$ and i, and assume $r_1 \in L(s_i)$.
Take $\pi' = s'_0 s'_1 s'_2 \ldots$ to be π^i. Then $\pi' \in Paths_{s_i}(\mathcal{M})$ and, for all j, $s'_j = s_{i+j}$.
By (1) applied to π, i and π', we obtain j such that $c_1 \in L(s'_j)$, i.e., $c_1 \in L(s_{i+j})$.

We know:

(1) for all $\pi = s_0 s_1 s_2 \ldots \in Paths_{s_0}(\mathcal{M})$, for all i,

if $r_1 \in L(s_i)$ then

for all $\pi' = s'_0 s'_1 s'_2 \ldots \in Paths_{s_i}(\mathcal{M})$, there exists j such that $c_1 \in L(s'_j)$



We show:

(2) for all $\pi = s_0 s_1 s_2 \ldots \in Paths_{s_0}(\mathcal{M})$, for all i,

if $r_1 \in L(s_i)$ then

there exists j such that $c_1 \in L(s_{i+j})$

Proving CTL–LTL Formula Equivalence

Let's prove that the CTL formula $\forall\Box\,(r_1 \to \forall\Diamond\, c_1)$ is equivalent to (i.e., is satisfied by the same LTSs in the same states as) the LTL formula $\Box\,(r_1 \to \Diamond c_1)$.

So, given an LTS $\mathcal{M} = (S, \to, L)$ and $s_0 \in S$, we are left the prove that the following statements are equivalent:

(1) for all $\pi = s_0 s_1 s_2 \ldots \in \mathit{Paths}_{s_0}(\mathcal{M})$, for all i,
 if $r_1 \in L(s_i)$ then
 for all $\pi' = s'_0 s'_1 s'_2 \ldots \in \mathit{Paths}_{s_i}(\mathcal{M})$, there exists j such that $c_1 \in L(s'_j)$

(2) for all $\pi = s_0 s_1 s_2 \ldots \in \mathit{Paths}_{s_0}(\mathcal{M})$, for all i,
 if $r_1 \in L(s_i)$ then
 there exists j such that $c_1 \in L(s_{i+j})$

Finally, assume (2) and let's prove (1):

Proving CTL–LTL Formula Equivalence

Let's prove that the CTL formula $\forall\Box\,(r_1 \to \forall\Diamond\, c_1)$ is equivalent to (i.e., is satisfied by the same LTSs in the same states as) the LTL formula $\Box\,(r_1 \to \Diamond c_1)$.

So, given an LTS $\mathcal{M} = (S, \to, L)$ and $s_0 \in S$, we are left the prove that the following statements are equivalent:

(1) for all $\pi = s_0 s_1 s_2 \ldots \in \mathit{Paths}_{s_0}(\mathcal{M})$, for all i,
    if $r_1 \in L(s_i)$ then
    for all $\pi' = s'_0 s'_1 s'_2 \ldots \in \mathit{Paths}_{s_i}(\mathcal{M})$, there exists j such that $c_1 \in L(s'_j)$

(2) for all $\pi = s_0 s_1 s_2 \ldots \in \mathit{Paths}_{s_0}(\mathcal{M})$, for all i,
    if $r_1 \in L(s_i)$ then
    there exists j such that $c_1 \in L(s_{i+j})$

Finally, assume (2) and let's prove (1):
Let $\pi = s_0 s_1 s_2 \ldots \in \mathit{Paths}_{s_0}(\mathcal{M})$ and i, and assume $r_1 \in L(s_i)$.

Proving CTL–LTL Formula Equivalence

Let's prove that the CTL formula $\forall\Box\,(r_1 \rightarrow \forall\Diamond\, c_1)$ is equivalent to (i.e., is satisfied by the same LTSs in the same states as) the LTL formula $\Box\,(r_1 \rightarrow \Diamond c_1)$.

So, given an LTS $\mathcal{M} = (S, \rightarrow, L)$ and $s_0 \in S$, we are left the prove that the following statements are equivalent:

(1) for all $\pi = s_0 s_1 s_2 \ldots \in Paths_{s_0}(\mathcal{M})$, for all i,
if $r_1 \in L(s_i)$ then
for all $\pi' = s'_0 s'_1 s'_2 \ldots \in Paths_{s_i}(\mathcal{M})$, there exists j such that $c_1 \in L(s'_j)$

(2) for all $\pi = s_0 s_1 s_2 \ldots \in Paths_{s_0}(\mathcal{M})$, for all i,
if $r_1 \in L(s_i)$ then
there exists j such that $c_1 \in L(s_{i+j})$

Finally, assume (2) and let's prove (1):
Let $\pi = s_0 s_1 s_2 \ldots \in Paths_{s_0}(\mathcal{M})$ and i, and assume $r_1 \in L(s_i)$.
Let $\pi' = s'_0 s'_1 s'_2 \ldots \in Paths_{s_i}(\mathcal{M})$.

Proving CTL–LTL Formula Equivalence

Let's prove that the CTL formula $\forall\Box\,(r_1 \rightarrow \forall\Diamond\, c_1)$ is equivalent to (i.e., is satisfied by the same LTSs in the same states as) the LTL formula $\Box\,(r_1 \rightarrow \Diamond c_1)$.

So, given an LTS $\mathcal{M} = (S, \rightarrow, L)$ and $s_0 \in S$, we are left the prove that the following statements are equivalent:

(1) for all $\pi = s_0 s_1 s_2 \ldots \in Paths_{s_0}(\mathcal{M})$, for all i,
if $r_1 \in L(s_i)$ then
for all $\pi' = s'_0 s'_1 s'_2 \ldots \in Paths_{s_i}(\mathcal{M})$, there exists j such that $c_1 \in L(s'_j)$

(2) for all $\pi = s_0 s_1 s_2 \ldots \in Paths_{s_0}(\mathcal{M})$, for all i,
if $r_1 \in L(s_i)$ then
there exists j such that $c_1 \in L(s_{i+j})$

Finally, assume (2) and let's prove (1):
Let $\pi = s_0 s_1 s_2 \ldots \in Paths_{s_0}(\mathcal{M})$ and i, and assume $r_1 \in L(s_i)$.
Let $\pi' = s'_0 s'_1 s'_2 \ldots \in Paths_{s_i}(\mathcal{M})$. (Note that $s'_0 = s_i$.)

Proving CTL–LTL Formula Equivalence

Let's prove that the CTL formula $\forall\Box\,(r_1 \to \forall\Diamond\, c_1)$ is equivalent to (i.e., is satisfied by the same LTSs in the same states as) the LTL formula $\Box\,(r_1 \to \Diamond c_1)$.

So, given an LTS $\mathcal{M} = (S, \to, L)$ and $s_0 \in S$, we are left the prove that the following statements are equivalent:

(1) for all $\pi = s_0 s_1 s_2 \ldots \in \mathit{Paths}_{s_0}(\mathcal{M})$, for all i,
 if $r_1 \in L(s_i)$ then
 for all $\pi' = s'_0 s'_1 s'_2 \ldots \in \mathit{Paths}_{s_i}(\mathcal{M})$, there exists j such that $c_1 \in L(s'_j)$

(2) for all $\pi = s_0 s_1 s_2 \ldots \in \mathit{Paths}_{s_0}(\mathcal{M})$, for all i,
 if $r_1 \in L(s_i)$ then
 there exists j such that $c_1 \in L(s_{i+j})$

Finally, assume (2) and let's prove (1):
Let $\pi = s_0 s_1 s_2 \ldots \in \mathit{Paths}_{s_0}(\mathcal{M})$ and i, and assume $r_1 \in L(s_i)$.
Let $\pi' = s'_0 s'_1 s'_2 \ldots \in \mathit{Paths}_{s_i}(\mathcal{M})$. (Note that $s'_0 = s_i$.)
Take $\pi'' = s''_0 s''_1 s''_2 \ldots$ to be $s_0 s_1 \ldots s_{i-1} s'_0 s'_1 s'_2 \ldots$

Proving CTL–LTL Formula Equivalence

Let's prove that the CTL formula $\forall\Box\,(r_1 \to \forall\Diamond\, c_1)$ is equivalent to (i.e., is satisfied by the same LTSs in the same states as) the LTL formula $\Box\,(r_1 \to \Diamond c_1)$.

So, given an LTS $\mathcal{M} = (S, \to, L)$ and $s_0 \in S$, we are left the prove that the following statements are equivalent:

(1) for all $\pi = s_0 s_1 s_2 \ldots \in \mathit{Paths}_{s_0}(\mathcal{M})$, for all i,
 if $r_1 \in L(s_i)$ then
 for all $\pi' = s'_0 s'_1 s'_2 \ldots \in \mathit{Paths}_{s_i}(\mathcal{M})$, there exists j such that $c_1 \in L(s'_j)$

(2) for all $\pi = s_0 s_1 s_2 \ldots \in \mathit{Paths}_{s_0}(\mathcal{M})$, for all i,
 if $r_1 \in L(s_i)$ then
 there exists j such that $c_1 \in L(s_{i+j})$

Finally, assume (2) and let's prove (1):
Let $\pi = s_0 s_1 s_2 \ldots \in \mathit{Paths}_{s_0}(\mathcal{M})$ and i, and assume $r_1 \in L(s_i)$.
Let $\pi' = s'_0 s'_1 s'_2 \ldots \in \mathit{Paths}_{s_i}(\mathcal{M})$. (Note that $s'_0 = s_i$.)
Take $\pi'' = s''_0 s''_1 s''_2 \ldots$ to be $s_0 s_1 \ldots s_{i-1} s'_0 s'_1 s'_2 \ldots$
Then $\pi'' \in \mathit{Paths}_{s_0}(\mathcal{M})$, $r_1 \in L(s''_i)$ and, for all j, $s''_{i+j} = s'_j$.

Proving CTL–LTL Formula Equivalence

Let's prove that the CTL formula $\forall\Box\,(r_1 \to \forall\Diamond\, c_1)$ is equivalent to (i.e., is satisfied by the same LTSs in the same states as) the LTL formula $\Box\,(r_1 \to \Diamond c_1)$.

So, given an LTS $\mathcal{M} = (S, \to, L)$ and $s_0 \in S$, we are left the prove that the following statements are equivalent:

(1) for all $\pi = s_0 s_1 s_2 \ldots \in Paths_{s_0}(\mathcal{M})$, for all i,
if $r_1 \in L(s_i)$ then
for all $\pi' = s'_0 s'_1 s'_2 \ldots \in Paths_{s_i}(\mathcal{M})$, there exists j such that $c_1 \in L(s'_j)$

(2) for all $\pi = s_0 s_1 s_2 \ldots \in Paths_{s_0}(\mathcal{M})$, for all i,
if $r_1 \in L(s_i)$ then
there exists j such that $c_1 \in L(s_{i+j})$

Finally, assume (2) and let's prove (1):
Let $\pi = s_0 s_1 s_2 \ldots \in Paths_{s_0}(\mathcal{M})$ and i, and assume $r_1 \in L(s_i)$.
Let $\pi' = s'_0 s'_1 s'_2 \ldots \in Paths_{s_i}(\mathcal{M})$. (Note that $s'_0 = s_i$.)
Take $\pi'' = s''_0 s''_1 s''_2 \ldots$ to be $s_0 s_1 \ldots s_{i-1} s'_0 s'_1 s'_2 \ldots$
Then $\pi'' \in Paths_{s_0}(\mathcal{M})$, $r_1 \in L(s''_i)$ and, for all j, $s''_{i+j} = s'_j$.
By (2) applied to π'' and i, we obtain j such that $c_1 \in L(s''_{i+j})$,

Proving CTL–LTL Formula Equivalence

Let's prove that the CTL formula $\forall\Box\,(r_1 \to \forall\Diamond\, c_1)$ is equivalent to (i.e., is satisfied by the same LTSs in the same states as) the LTL formula $\Box\,(r_1 \to \Diamond c_1)$.

So, given an LTS $\mathcal{M} = (S, \to, L)$ and $s_0 \in S$, we are left the prove that the following statements are equivalent:

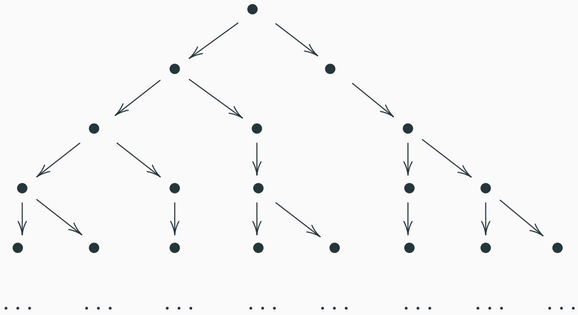
(1) for all $\pi = s_0 s_1 s_2 \ldots \in Paths_{s_0}(\mathcal{M})$, for all i,
if $r_1 \in L(s_i)$ then
for all $\pi' = s'_0 s'_1 s'_2 \ldots \in Paths_{s_i}(\mathcal{M})$, there exists j such that $c_1 \in L(s'_j)$

(2) for all $\pi = s_0 s_1 s_2 \ldots \in Paths_{s_0}(\mathcal{M})$, for all i,
if $r_1 \in L(s_i)$ then
there exists j such that $c_1 \in L(s_{i+j})$

Finally, assume (2) and let's prove (1):
Let $\pi = s_0 s_1 s_2 \ldots \in Paths_{s_0}(\mathcal{M})$ and i, and assume $r_1 \in L(s_i)$.
Let $\pi' = s'_0 s'_1 s'_2 \ldots \in Paths_{s_i}(\mathcal{M})$. (Note that $s'_0 = s_i$.)
Take $\pi'' = s''_0 s''_1 s''_2 \ldots$ to be $s_0 s_1 \ldots s_{i-1} s'_0 s'_1 s'_2 \ldots$
Then $\pi'' \in Paths_{s_0}(\mathcal{M})$, $r_1 \in L(s''_i)$ and, for all j, $s''_{i+j} = s'_j$.
By (2) applied to π'' and i, we obtain j such that $c_1 \in L(s''_{i+j})$, i.e., $c_1 \in L(s'_j)$.

We know:

(2) for all $\pi = s_0 s_1 s_2 \ldots \in \mathit{Paths}_{s_0}(\mathcal{M})$, for all i,
 if $r_1 \in L(s_i)$ then
 there exists j such that $c_1 \in L(s_{i+j})$



We show:

(1) for all $\pi = s_0 s_1 s_2 \ldots \in \mathit{Paths}_{s_0}(\mathcal{M})$, for all i,
 if $r_1 \in L(s_i)$ then
 for all $\pi' = s'_0 s'_1 s'_2 \ldots \in \mathit{Paths}_{s_i}(\mathcal{M})$, there exists j such that $c_1 \in L(s'_j)$

CTL Model Checking (Very Briefly)

CTL Model Checking Problem

Let $\mathcal{M} = (S, \rightarrow, L)$ be an LTS, $s_0 \in S$, and φ a CTL formula.

The CTL model checking problem is to determine whether $\mathcal{M}, s_0 \models \varphi$, i.e., whether $\mathcal{M}$ satisfies φ in state s_0.

CTL Model Checking Problem

Let $\mathcal{M} = (S, \rightarrow, L)$ be an LTS, $s_0 \in S$, and φ a CTL formula.

The CTL model checking problem is to determine whether $\mathcal{M}, s_0 \models \varphi$, i.e., whether $\mathcal{M}$ satisfies φ in state s_0.

Theorem. The CTL model checking problem is algorithmically decidable in time and space complexity $O(|\mathcal{M}| \times |\varphi|)$, where $|\mathcal{M}|$ is defined as $|S| + |\rightarrow|$.

CTL Model Checking Problem

Let $\mathcal{M} = (S, \rightarrow, L)$ be an LTS, $s_0 \in S$, and φ a CTL formula.

The CTL model checking problem is to determine whether $\mathcal{M}, s_0 \models \varphi$, i.e., whether $\mathcal{M}$ satisfies φ in state s_0.

Theorem. The CTL model checking problem is algorithmically decidable in time and space complexity $O(|\mathcal{M}| \times |\varphi|)$, where $|\mathcal{M}|$ is defined as $|S| + |\rightarrow|$.

Idea: Traverse the formula φ by structural recursion, labeling with φ all the states s for which φ holds (i.e., $\mathcal{M}, s \models \varphi$).

CTL Model Checking Problem

Let $\mathcal{M} = (S, \rightarrow, L)$ be an LTS, $s_0 \in S$, and φ a CTL formula.

The CTL model checking problem is to determine whether $\mathcal{M}, s_0 \models \varphi$, i.e., whether $\mathcal{M}$ satisfies φ in state s_0.

Theorem. The CTL model checking problem is algorithmically decidable in time and space complexity $O(|\mathcal{M}| \times |\varphi|)$, where $|\mathcal{M}|$ is defined as $|S| + |\rightarrow|$.

Idea: Traverse the formula φ by structural recursion, labeling with φ all the states s for which φ holds (i.e., $\mathcal{M}, s \models \varphi$). For example:

- If φ is an atom a, then we label s with φ just in case $a \in L(s)$.

CTL Model Checking Problem

Let $\mathcal{M} = (S, \to, L)$ be an LTS, $s_0 \in S$, and φ a CTL formula.

The CTL model checking problem is to determine whether $\mathcal{M}, s_0 \models \varphi$, i.e., whether $\mathcal{M}$ satisfies φ in state s_0.

Theorem. The CTL model checking problem is algorithmically decidable in time and space complexity $O(|\mathcal{M}| \times |\varphi|)$, where $|\mathcal{M}|$ is defined as $|S| + |\to|$.

Idea: Traverse the formula φ by structural recursion, labeling with φ all the states s for which φ holds (i.e., $\mathcal{M}, s \models \varphi$). For example:

- If φ is an atom a, then we label s with φ just in case $a \in L(s)$.
- If φ has the form $\varphi_1 \wedge \varphi_2$, then we label s with φ just in case s is already labeled with both φ_1 and φ_2.

CTL Model Checking Problem

Let $\mathcal{M} = (S, \rightarrow, L)$ be an LTS, $s_0 \in S$, and φ a CTL formula.

The CTL model checking problem is to determine whether $\mathcal{M}, s_0 \models \varphi$, i.e., whether $\mathcal{M}$ satisfies φ in state s_0.

Theorem. The CTL model checking problem is algorithmically decidable in time and space complexity $O(|\mathcal{M}| \times |\varphi|)$, where $|\mathcal{M}|$ is defined as $|S| + |\rightarrow|$.

Idea: Traverse the formula φ by structural recursion, labeling with φ all the states s for which φ holds (i.e., $\mathcal{M}, s \models \varphi$). For example:

- If φ is an atom a, then we label s with φ just in case $a \in L(s)$.
- If φ has the form $\varphi_1 \wedge \varphi_2$, then we label s with φ just in case s is already labeled with both φ_1 and φ_2.
- If φ has the form $\exists\Box\varphi$, then, by fixpoint iteration, we label all states that belong to the largest subset T of S with the following properties:
  - all states in T are already labeled with φ;
  - if $s \in T$, then there exists s' such as $s \rightarrow s'$ and $s' \in T$.

CTL Model Checking Problem

Let $\mathcal{M} = (S, \rightarrow, L)$ be an LTS, $s_0 \in S$, and φ a CTL formula.

The CTL model checking problem is to determine whether $\mathcal{M}, s_0 \models \varphi$, i.e., whether $\mathcal{M}$ satisfies φ in state s_0.

Theorem. The CTL model checking problem is algorithmically decidable in time and space complexity $O(|\mathcal{M}| \times |\varphi|)$, where $|\mathcal{M}|$ is defined as $|S| + |\rightarrow|$.

Idea: Traverse the formula φ by structural recursion, labeling with φ all the states s for which φ holds (i.e., $\mathcal{M}, s \models \varphi$). For example:

- If φ is an atom a, then we label s with φ just in case $a \in L(s)$.
- If φ has the form $\varphi_1 \wedge \varphi_2$, then we label s with φ just in case s is already labeled with both φ_1 and φ_2.
- If φ has the form $\exists\Box\varphi$, then, by fixpoint iteration, we label all states that belong to the largest subset T of S with the following properties:
  - all states in T are already labeled with φ;
  - if $s \in T$, then there exists s' such as $s \rightarrow s'$ and $s' \in T$.

Main lemma: Characterization of the CTL connectives' semantics by means of least and greatest fixpoints of suitable operators on $\mathcal{P}(S)$.

CTL Model Checking Problem

Let $\mathcal{M} = (S, \rightarrow, L)$ be an LTS, $s_0 \in S$, and φ a CTL formula.

The CTL model checking problem is to determine whether $\mathcal{M}, s_0 \models \varphi$, i.e., whether $\mathcal{M}$ satisfies φ in state s_0.

Theorem. The CTL model checking problem is algorithmically decidable in time and space complexity $O(|\mathcal{M}| \times |\varphi|)$, where $|\mathcal{M}|$ is defined as $|S| + |\rightarrow|$.

Idea: Traverse the formula φ by structural recursion, labeling with φ all the states s for which φ holds (i.e., $\mathcal{M}, s \models \varphi$). For example:

- If φ is an atom a, then we label s with φ just in case $a \in L(s)$.
- If φ has the form $\varphi_1 \wedge \varphi_2$, then we label s with φ just in case s is already labeled with both φ_1 and φ_2.
- If φ has the form $\exists\Box\varphi$, then, by fixpoint iteration, we label all states that belong to the largest subset T of S with the following properties:
  - all states in T are already labeled with φ;
  - if $s \in T$, then there exists s' such as $s \rightarrow s'$ and $s' \in T$.

Main lemma: Characterization of the CTL connectives' semantics by means of least and greatest fixpoints of suitable operators on $\mathcal{P}(S)$.

Not covered in these lectures – see Section 6.4 of Baier & Katoen's "Principles of Model Checking" (MIT Press 2008). Simpler than LTL model checking!

Ending

Summary of the Discussed Concepts

- CTL = Computation Tree Logic
- Syntax = formulas built from
  - atoms
  - propositional connectives
  - CTL connectives, each consisting of a path quantifier and a temporal operator
- Semantics = the satisfaction relation defined on LTSs
- Formula equivalence
- CTL versus LTL
- Brief sketch of CTL model checking

Further Reading

Section 6 of Baier & Katoen's "Principles of Model Checking" (MIT Press 2008)

- Distinguishes state formulas from path formulas
- Writes $\forall(\varphi \mathsf{U} \psi)$ instead of $\varphi \, \forall\mathsf{U} \, \psi$, and $\exists(\varphi \mathsf{U} \psi)$ instead of $\varphi \, \exists\mathsf{U} \, \psi$

Section 3.4 of Huth & Ryan's "Logic in Computer Science: Modelling and Reasoning about Systems" (Cambridge University Press 2004)

Uses different notations for the CTL connectives:

- X instead of $\bigcirc$, F instead of $\Diamond$, G instead of $\Box$ (just like it does for LTL)
- A instead of $\forall$, E instead of $\exists$
- Hence writes AF instead of $\forall\Diamond$, EG instead of $\exists\Box$, etc.
- Also, $\mathsf{A}(\varphi \mathsf{U} \psi)$ instead of $\varphi \, \forall\mathsf{U} \, \psi$, and $\mathsf{E}(\varphi \mathsf{U} \psi)$ instead of $\varphi \, \exists\mathsf{U} \, \psi$