2

Model Checking

Computation Tree Logic (CTL)

[Baier & Katoen, Chapter 6.1–6.3]

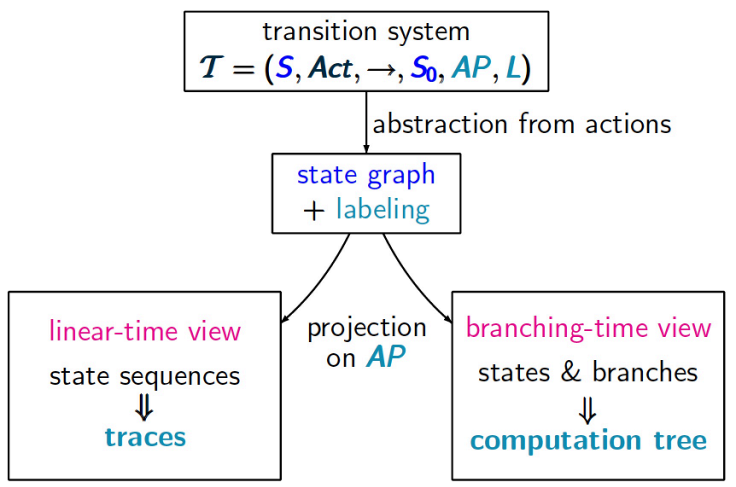
Prof. Dr. Nils Jansen
Radboud University, 2025

Credit to the slides: Prof. Dr. Dr.h.c. Joost-Pieter Katoen

Overview

Overview

Linear Time Versus Branching Time

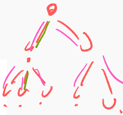


Linear Time Versus Branching Time

- Linear Temporal Logic (LTL) is interpreted over $\underbrace{\text{infinite sequences}}_{\text{traces}}$
- Traces are obtained from paths in a transition system.

- Computation Tree Logic (CTL) is interpreted over $\underbrace{\text{infinite trees}}_{\text{computation trees}}$
- Computation trees are infinite trees whose nodes are labelled with sets of propositions
  - They are obtained by unfolding a transition system
  - Such trees keep track of branching between several traces

Linear Time Versus Branching Time



	linear time	branching time
behavior	path based traces	state based computation tree
temporal logic	**LTL** path formulas	**CTL** state formulas
model checking	PSPACE-complete $\mathcal{O}(size(\mathcal{T}) \cdot \exp(\lvert\varphi\rvert))$	PTIME $\mathcal{O}(size(\mathcal{T}) \cdot \lvert\Phi\rvert)$
impl. relation	trace inclusion trace equivalence PSPACE-complete	simulation bisimulation PTIME

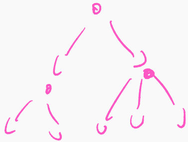
Overview

Computation Tree Logic

Edmund M. Clarke, Jr.
(1945–†2020)

E. Allen Emerson
(1954–†2024)

CTL Syntax



Definition: Syntax Computation Tree Logic

- CTL state-formulas with $a \in AP$ obey the grammar:

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

- and φ is a path-formula formed by the grammar:

$$\varphi \ ::= \ \bigcirc\Phi \ \Big| \ \Phi_1 \,\mathsf{U}\, \Phi_2.$$

Example CTL State-formulas

- $\forall\Box\exists\bigcirc a$
- $\exists(\forall\Box a)\,\mathsf{U}\, b$

CTL Syntax

Definition: Syntax Computation Tree Logic

- CTL state-formulas with $a \in AP$ obey the grammar:

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

- and φ is a path-formula formed by the grammar:

$$\varphi \;::=\; \bigcirc\,\Phi \;\Big|\; \Phi_1 \,\mathsf{U}\, \Phi_2.$$

Intuition

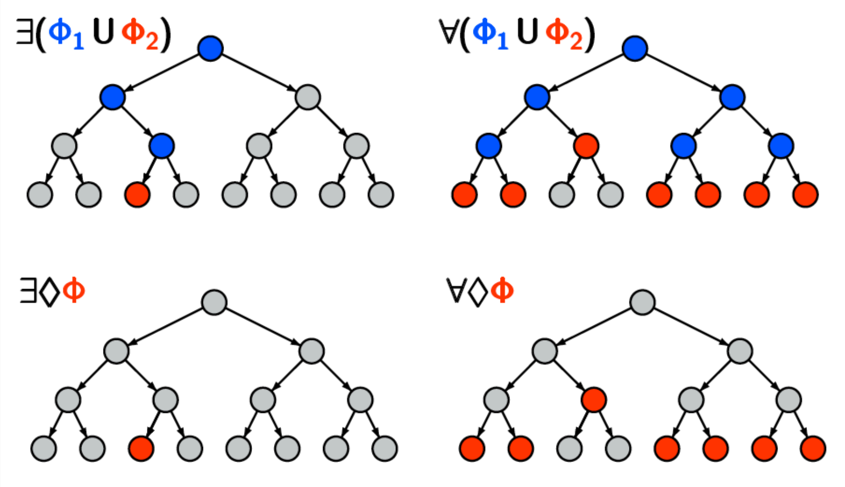
- $s \models \forall\varphi$ if all paths starting in s fulfill φ
- $s \models \exists\varphi$ if some path starting in s fulfill φ

Derived CTL Operators

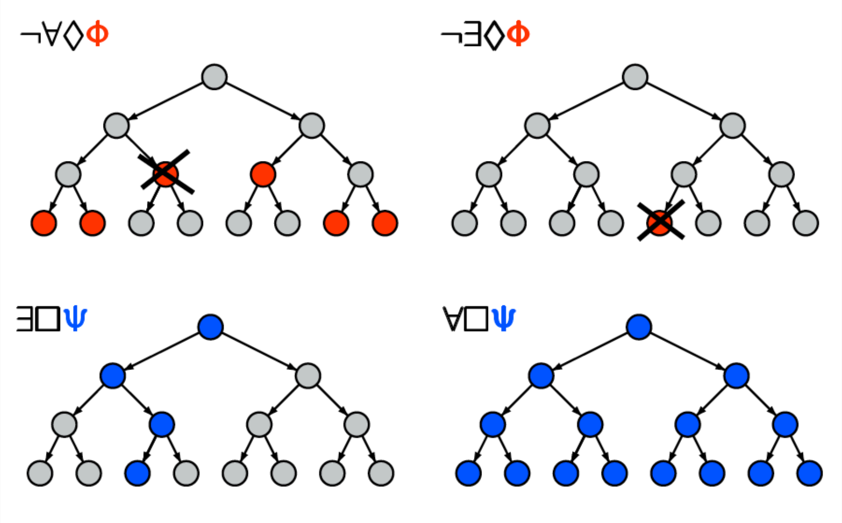
potentially Φ:	$\exists\Diamond\,\Phi$	$=$	$\exists(\mathrm{true}\,\mathsf{U}\,\Phi)$
inevitably Φ:	$\forall\Diamond\,\Phi$	$=$	$\forall(\mathrm{true}\,\mathsf{U}\,\Phi)$
potentially always Φ:	$\exists\Box\Phi$	$=$	$\neg\forall\Diamond\,\neg\Phi$
invariantly Φ:	$\forall\Box\Phi$	$=$	$\neg\exists\Diamond\,\neg\Phi$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

The Boolean connectives are derived as usual

Intuitive CTL Semantics



Intuitive CTL Semantics



Overview

CTL Semantics

Define a satisfaction relation for CTL-formulas over AP for a given transition system TS without terminal states.

Two parts:

- Interpretation of state-formulas over states of TS
- Interpretation of path-formulas over paths of TS

CTL Semantics (1)

Notation
$TS, s \vDash \Phi$ if and only if state-formula Φ holds in state s of transition system TS. As TS is known from the context we simply write $s \vDash \Phi$.

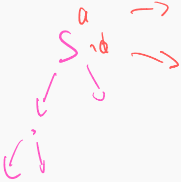
Definition: Satisfaction relation for CTL state-formulas
The satisfaction relation $\vDash$ is defined for CTL state-formulas by:

$$
\begin{array}{lll}
s \vDash a & \text{iff} & a \in L(s) \\
s \vDash \neg \Phi & \text{iff} & \text{not } (s \vDash \Phi) \\
s \vDash \Phi \wedge \Psi & \text{iff} & (s \vDash \Phi) \text{ and } (s \vDash \Psi)
\end{array}
$$

$s^{\{a,b\}}$

Notation
$TS, s \models \Phi$ if and only if state-formula Φ holds in state s of transition system TS. As TS is known from the context we simply write $s \models \Phi$.

Definition: Satisfaction relation for CTL state-formulas
The satisfaction relation $\models$ is defined for CTL state-formulas by:



$$
\begin{array}{lll}
s \models a & \text{iff} & a \in L(s) \\
s \models \neg\,\Phi & \text{iff} & \text{not } (s \models \Phi) \\
s \models \Phi \wedge \Psi & \text{iff} & (s \models \Phi) \text{ and } (s \models \Psi) \\
s \models \exists\varphi & \text{iff} & \text{there exists } \pi \in Paths(s).\ \pi \models \varphi \\
s \models \forall\varphi & \text{iff} & \text{for all } \pi \in Paths(s).\ \pi \models \varphi
\end{array}
$$

where the semantics of CTL path-formulas is defined on the next slide.

CTL Semantics (2)

Definition: satisfaction relation for CTL path-formulas
Given path π and CTL path-formula φ, the satisfaction relation $\models$ where $\pi \models \varphi$ if and only if path π satisfies φ is defined as follows:

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

$$\pi \models \Phi \, \mathsf{U} \, \Psi \qquad \text{iff } (\exists\, j \geq 0.\, \pi[j] \models \Psi \text{ and } (\forall\, 0 \leq i < j.\, \pi[i] \models \Phi))$$

where $\pi[i]$ denotes the state s_i in the path $\pi = s_0\, s_1\, s_2 \ldots$.

Transition System Semantics

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

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

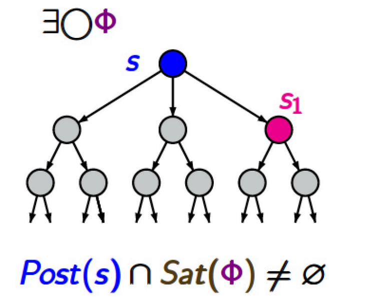
- TS satisfies CTL-formula Φ iff Φ holds in all its initial states:

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

- Point of attention: $TS \not\models \Phi$ is not equivalent to $TS \models \neg\Phi$

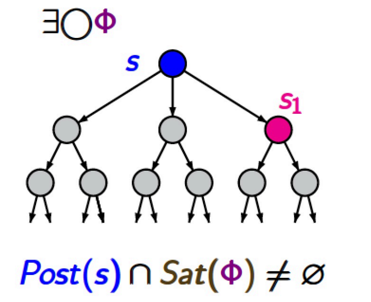
  because of several initial states, e.g., $s_0 \models \exists\Box\Phi$ and $s_0' \not\models \exists\Box\Phi$

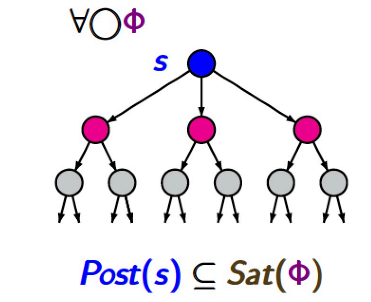
Semantics of $\bigcirc$-Operator



$s \vDash \exists \bigcirc \Phi$ iff $\exists \pi = s\, s_1\, s_2 \ldots \in Paths(s).\ \pi \vDash \bigcirc \Phi$, that is: $s_1 \vDash \Phi$

Semantics of $\bigcirc$-Operator

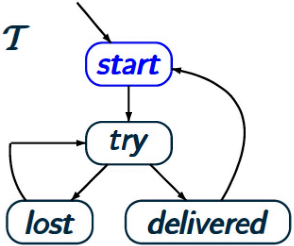




$s \vDash \exists\bigcirc \Phi$ iff $\exists \pi = s\, s_1\, s_2 \ldots \in \mathit{Paths}(s).\ \pi \vDash \bigcirc \Phi$, that is: $s_1 \vDash \Phi$

$s \vDash \forall\bigcirc \Phi$ iff $\forall \pi = s\, s_1\, s_2 \ldots \in \mathit{Paths}(s).\ \pi \vDash \bigcirc \Phi$, that is: $s_1 \vDash \Phi$

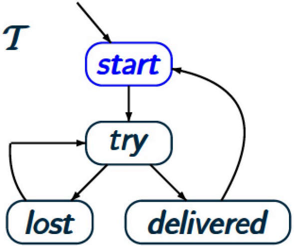
Example



$\mathcal{T} \not\models \forall\Box\forall\Diamond\mathit{start}$

Quiz

Example



$\mathcal{T} \not\models \forall\Box\forall\Diamond start$

CTL formula

$\Phi = \forall\Box\forall\Diamond start \quad \widehat{=} \quad \forall\Box(start \vee delivered)$

$$Sat(\forall\Diamond start) = \{start, delivered\}$$
$$Sat(\Phi) = \varnothing$$

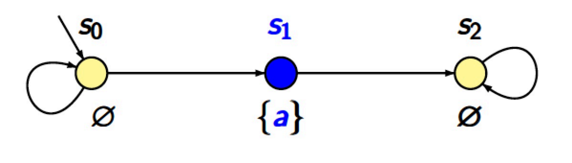
Infinitely Often

$s \models \forall\Box\forall\Diamond\, \Phi$ iff $\forall\pi \in \mathit{Paths}(s)$ a Φ-state is visited infinitely often.

For $\Phi = a \in AP$ we get

$$\underbrace{s \models \forall\Box\forall\Diamond\, a}_{\text{CTL}} \quad \text{iff} \quad \underbrace{s \models \Box\Diamond\, a}_{\text{LTL}}$$

Example



(1) Does $TS \models \exists \bigcirc \forall \Box \neg a$?

(2) Does $TS \models \forall \Box \exists \bigcirc \neg a$?

Overview

CTL Equivalence

Definition: CTL equivalence
CTL-formulas Φ and Ψ (both over AP) are equivalent:

$$\Phi \equiv_{CTL} \Psi \quad \text{if and only if} \quad Sat(\Phi) = Sat(\Psi) \quad \text{for any } TS \text{ (over } AP\text{)}$$

If it is clear from the context that we deal with CTL-formulas, we simply write $\Phi \equiv \Psi$.

Equivalently,

$$\Phi \equiv \Psi \quad \text{iff} \quad \big(\forall TS \ . \ TS \models \Phi \quad \text{iff} \quad TS \models \Psi\big)$$

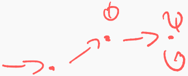
Duality

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

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

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

But: $\forall\Diamond(\Phi \vee \Psi) \quad \not\equiv \quad \forall\Diamond\Phi \vee \forall\Diamond\Psi$



Distributive Laws

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

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

But: $\forall\Diamond(\Phi \vee \Psi) \quad \not\equiv \quad \forall\Diamond\Phi \vee \forall\Diamond\Psi$

$$\exists\Box(\Phi \wedge \Psi) \quad \not\equiv \quad \exists\Box\Phi \wedge \exists\Box\Psi$$

Duality $\bigcirc$ and $\Box$ — Correct or Wrong?

$$\forall\bigcirc\forall\Box a \equiv \forall\Box\forall\bigcirc a$$

Duality $\bigcirc$ and $\square$ — Correct or Wrong?

$\forall\bigcirc\forall\square a \equiv \forall\square\forall\bigcirc a$

correct.

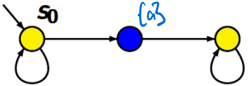
$\exists\bigcirc\exists\square a \equiv \exists\square\exists\bigcirc a$

$\forall\bigcirc\forall\Box a \equiv \forall\Box\forall\bigcirc a$

correct.

$\exists\bigcirc\exists\Box a \equiv \exists\Box\exists\bigcirc a$

wrong, e.g.,



$s_0 \not\models \exists\bigcirc\exists\Box a$

$s_0 \models \exists\Box\exists\bigcirc a$

$$
\begin{aligned}
& s_0 \models \exists\bigcirc a \\
\Longrightarrow\ & s_0\, s_0\, s_0 \ldots \models \Box\exists\bigcirc a \\
\Longrightarrow\ & s_0 \models \exists\Box\exists\bigcirc a
\end{aligned}
$$

Overview

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

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

Examples

- "Next a": $\forall\bigcirc a \equiv \bigcirc a$
- "Eventually a": $\forall\lozenge a \equiv \lozenge a$
- "Infinitely often a": $\cancel{\forall}\square\cancel{\forall}\lozenge a \equiv \square\lozenge a$

What about e.g. $\forall\lozenge(a \wedge \forall\bigcirc a) \equiv \ldots$?

$\lozenge(a \wedge \bigcirc a)$

LTL and CTL are Incomparable

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

  There does not exist an equivalent CTL formula

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

  There does not exist an equivalent LTL formula

How to prove this formally?

From CTL to LTL

[Clarke & Draghicescu]

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

either $\Phi \equiv \varphi$ or there is no LTL-formula equivalent to Φ.

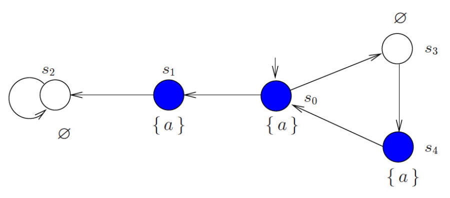
Examples

- $\forall \bigcirc a \quad \mapsto$
- $\forall \Diamond (a \wedge \forall \bigcirc a) \quad \mapsto$
- $\forall \Box \forall \Diamond a \quad \mapsto$
- $\forall \Box \exists \Diamond a \quad \mapsto$

From CTL To LTL (1)

CTL-formula $\forall\Diamond(a \wedge \forall\bigcirc a)$ cannot be expressed in LTL.

Proof.
We show that $\underbrace{\forall\Diamond(a \wedge \forall\bigcirc a)}_{\text{CTL-formula } \Phi} \not\equiv \underbrace{\Diamond(a \wedge \bigcirc a)}_{\Phi \text{ with path-quantifiers removed}}$.



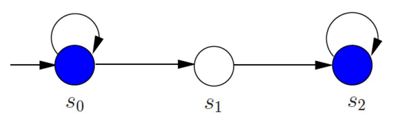
$s_0 \models \Diamond(a \wedge \bigcirc a)$ but $\underbrace{s_0 \not\models \forall\Diamond(a \wedge \forall\bigcirc a)}_{\text{path } s_0\, s_1\, (s_2)^\omega \text{ violates it}}$.

$\square$

From CTL To LTL (2)

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

Proof.
We show that: $\forall\Diamond\forall\Box a$ is not equivalent to $\Diamond\Box a$.



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

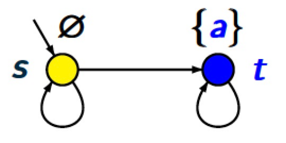
□

From CTL To LTL (3)

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

Proof.

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



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

$\square$

From LTL To CTL

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

Proof.
Provide two series of transition systems TS_n and TS'_n for $n = 0, 1, 2, \ldots$ (see next slide) such that:

- $TS_n \not\models \Diamond\Box a$ and $TS'_n \models \Diamond\Box\, a$ (*), and
- For any CTL-formula Φ with $|\Phi| \leq n$: $TS_n \models \Phi$ iff $TS'_n \models \Phi$ (**)

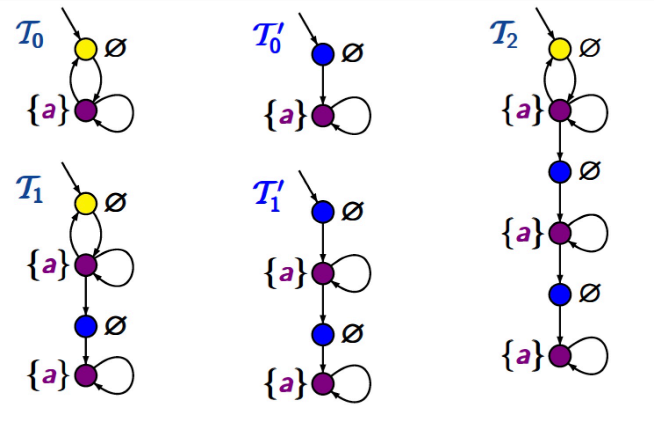
  proof by induction on n (omitted here)

Proof by contraposition.
<u>Assume</u> there is a CTL-formula $\Phi \equiv \Diamond\Box a$ with $|\Phi| = n$ for some n

- by (*), it follows $TS_n \not\models \Phi$ and $TS'_n \models \Phi$
- but this contradicts (**): $TS_n \models \Phi$ if and only if $TS'_n \models \Phi$

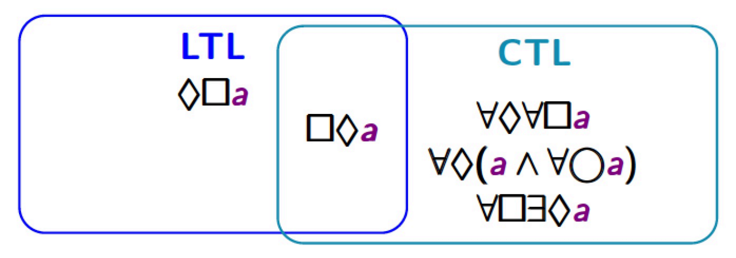
$\Box$

Proof



$\mathcal{T}_0$
$\emptyset$
$\{a\}$
$\mathcal{T}_0'$
$\emptyset$
$\{a\}$
$\mathcal{T}_2$
$\emptyset$
$\{a\}$
$\emptyset$
$\{a\}$
$\emptyset$
$\{a\}$
$\mathcal{T}_1$
$\emptyset$
$\{a\}$
$\emptyset$
$\{a\}$
$\mathcal{T}_1'$
$\emptyset$
$\{a\}$
$\emptyset$
$\{a\}$

LTL Versus CTL



Syntax of CTL*

Definition: Syntax CTL*

- CTL* state-formulas with $a \in AP$ obey the grammar:

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

- and φ is a CTL* path-formula formed by the grammar:

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

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

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

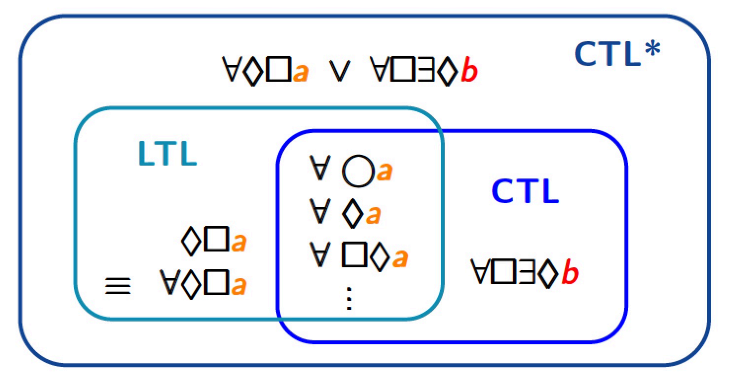
CTL* Is More Expressive Than LTL And CTL

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

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

can neither be expressed in LTL nor in CTL.

Relating LTL, CTL, and CTL*



Overview

Summary

- Computation tree logic (CTL) is a logic interpreted over infinite trees
- Path quantifiers in CTL alternate with temporal modalities
- CTL and LTL have an incomparable expressive power
- A CTL-formula Φ is equivalent to:
  - the LTL-formula obtained by removing all path quantifiers from Φ, or
  - there is no equivalent LTL-formula
- CTL^* is strictly more expressive than LTL and CTL