

Computation Tree Logic

IN5110 Model checking

Brage Joachim Andersen & Vegar Skaret

November 29, 2019

Main early contributions

- Linear Temporal Logic (LTL) proposed in 1977 (Pnueli) [2]
- Computation Tree Logic (CTL) proposed in 1980 (Clarke and Emerson)
- CTL also introduced by Sifakis in 1981 independently of Clarke and Emerson
- CTL* introduced in 1986 (Emerson and Halpern)



Pnueli



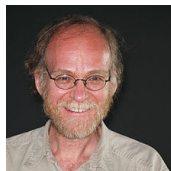
Clarke



Emerson



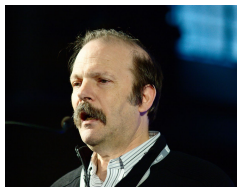
Sifakis



Halpern

"War" of the branching- and linear-time logics

- Papers in the 80's and 90's comparing LTL and CTL
- Consensus: while specifying is easier in LTL, verification is easier for CTL



- Moshe Vardi, 2001: Branching vs. Linear Time: Final Showdown
- Vardi concludes: LTL is usually preferred over CTL.
- CTL "is unintuitive and hard to use, it does not lend itself to compositional reasoning, and it is fundamentally incompatible with semi-formal verification."

The Computation Tree

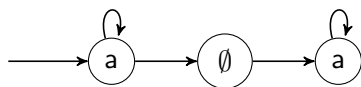


Figure: Transition system

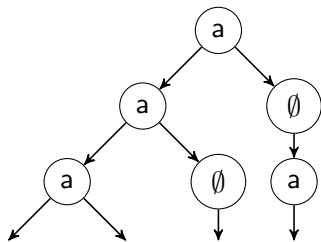


Figure: Start of the transition system's computation tree

CTL Syntax 1

Basic Syntax

$$\Phi ::= \top \mid a \mid \neg\Phi \mid \Phi_1 \wedge \Phi_2 \mid \exists\varphi \mid \forall\varphi$$
$$\varphi ::= \bigcirc\Phi \mid \Phi_1 U \Phi_2$$

Where a is an atomic proposition. Φ is called a state formula and φ is called a path formula. Note that a path formula must be preceded by a quantifier to be a legal CTL formula.

Abbreviations

$$\exists\Diamond\Phi \equiv \exists(\top U \Phi)$$
$$\forall\Diamond\Phi \equiv \forall(\top U \Phi)$$
$$\exists\Box\Phi \equiv \neg\forall\Diamond\neg\Phi$$
$$\forall\Box\Phi \equiv \neg\exists\Diamond\neg\Phi$$
$$\perp \equiv \neg\top$$
$$\Phi_1 \vee \Phi_2 \equiv \neg\Phi_1 \wedge \neg\Phi_2$$
$$\Phi_1 \rightarrow \Phi_2 \equiv \neg\Phi_1 \vee \Phi_2$$

Weak until and release can be defined similarly.

CTL Syntax 2

Some legal formulas

- $\forall \square black$
- $\forall (gray \cup black)$
- $\exists \bigcirc \forall \square black$

Some illegal formulas

- $\square black$
- $gray \cup black$
- $\exists \bigcirc \square black$

CTL Semantics 1: Intuition

- For path formulas: \bigcirc , \square , \diamond , and U have the same semantics as in LTL:
 - ▶ \bigcirc : "next"
 - ▶ U : "until"
 - ▶ \square : "always"
 - ▶ \diamond : "eventually"
- State formulas can quantify over paths beginning in the current state. $\forall\varphi$ means that φ is true for all paths from the current state. $\exists\varphi$ means that it is true for at least one path.
- CTL formulas are interpreted over the states s and paths π of a transition system TS
- Some CTL formulas verbalized:
 - ▶ $\exists\diamond\Phi$: " Φ holds potentially"
 - ▶ $\forall\diamond\Phi$: " Φ is inevitable"
 - ▶ $\exists\square\Phi$: "potentially always Φ "
 - ▶ $\forall\square\Phi$: "invariantly Φ "

CTL Semantics 2: Formal

State semantics

A CTL formula Φ is true relative to a state s , written $s \models \Phi$, in the following cases:

$s \models a$	iff	$a \in L(s)$
$s \models \neg\Phi$	iff	not $s \models \Phi$
$s \models \Phi \wedge \Psi$	iff	$(s \models \Phi)$ and $s \models \Psi$
$s \models \exists\varphi$	iff	$\pi \models \varphi$ for some $\pi \in Paths(s)$
$s \models \forall\varphi$	iff	$\pi \models \varphi$ for all $\pi \in Paths(s)$

Where:

- $L(s)$ is a labeling function that returns a set with all atomic propositions in s that are true
- $Paths(s)$ denotes the set of all maximal paths starting in the state s . A maximal path is a path that cannot be prolonged.

CTL Semantics 3: Formal

Path semantics

A CTL formula Φ is true relative to a path π , written $\pi \models \Phi$, in the following cases:

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

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

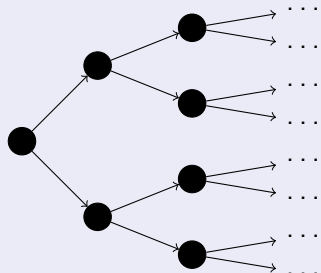
Where $\pi[1]$ means the next state, as in the second state on the path.

In English (kind of):

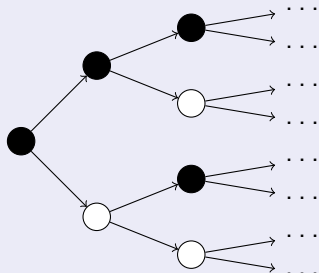
- π satisfies $\bigcirc\Phi$ iff the second state on the path satisfies Φ .
- π satisfies $\Phi U \Psi$ iff Φ is true for all the states before Ψ is true on the path.

CTL Semantics Example 1

$\forall \square black$

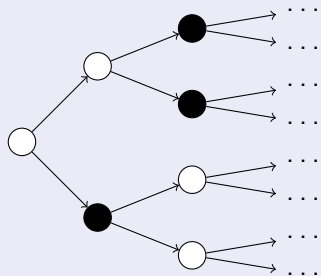


$\exists \square black$

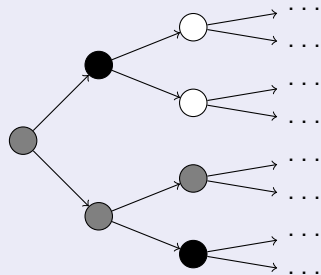


CTL Semantics Example 2

$\forall \diamond black$

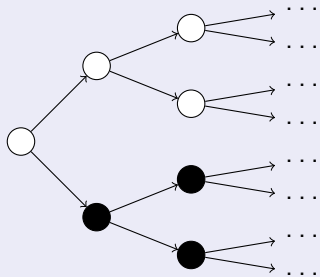


$\forall (gray \cup black)$

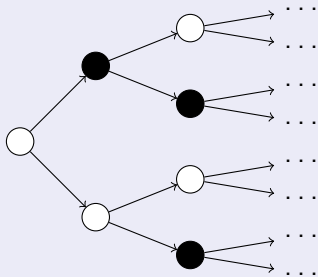


CTL Semantics Example 3

$\exists \bigcirc \forall \square \text{black}$



$\forall \square \exists \bigcirc \text{black}$



The Satisfaction Set $Sat(\Phi)$

The satisfaction set $Sat(\Phi)$ is the set of states in a transition system TS that satisfies Φ .

A transition system satisfies Φ , written $TS \models \Phi$, iff all the initial states of the TS satisfies Φ : $I \subseteq Sat(\Phi)$, where I is the set of initial states in TS .

Model-checking CTL

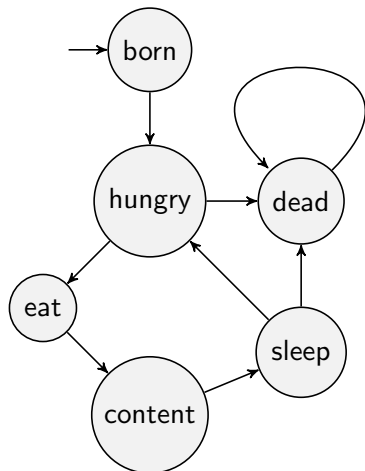
The task is to check whether a transition system TS satisfies a CTL formula Φ . This is the case when all the initial states I of the TS satisfy Φ .

Basic Algorithm

- 1 The set $Sat(\Phi)$ of all states satisfying Φ is computed recursively ("from inside and out")
- 2 $TS \models \Phi$ iff $I \subseteq Sat(\Phi)$

This can be achieved by a bottom-up traversal of the CTL formula's parse tree.

Model-checking Example 1: Is death inevitable?



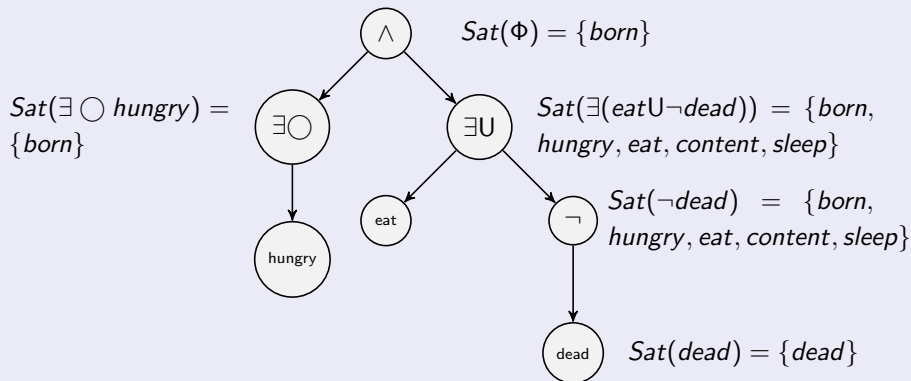
$\forall \diamond \text{dead}?$

$$\text{Sat}(\forall \diamond \text{dead}) = \{\text{dead}\}$$

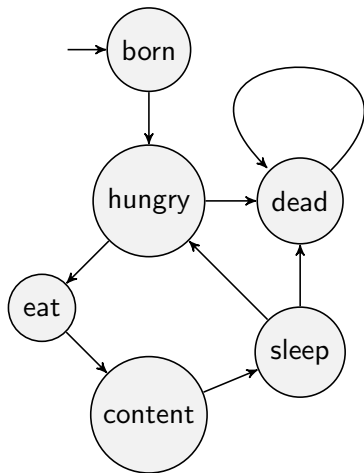
The initial state $\text{born} \notin \text{Sat}(\forall \diamond \text{dead})$,
so $TS \not\models \forall \diamond \text{dead}$

Model-checking Example 2

$$\Phi = \exists \bigcirc \text{hungry} \wedge \exists (\text{eat} \cup \neg \text{dead})$$



Model-checking Example 2



$$Sat(\Phi) = \{born\}$$

Because the only initial state is in the formula's satisfaction set, the transition system satisfies the formula.

LTL Comparison

CTL and LTL are not equally expressive, but neither is more expressive than the other.

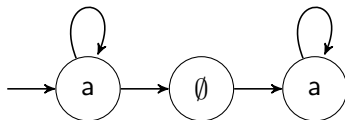
Theorem 6.18 [1]

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

$\Phi \equiv \varphi$ or there exists no LTL formula that is equivalent to Φ .

Lemma 6.19

$\forall \Diamond \forall \Box a \not\equiv \Diamond \Box a$



- The CTL* syntax is the same as CTL with the addition of allowing path formulas to appear without being prefixed by a quantifier.
- Any CTL or LTL formula is also a CTL* formula. But there are also CTL* formulas that aren't CTL or LTL formulas.

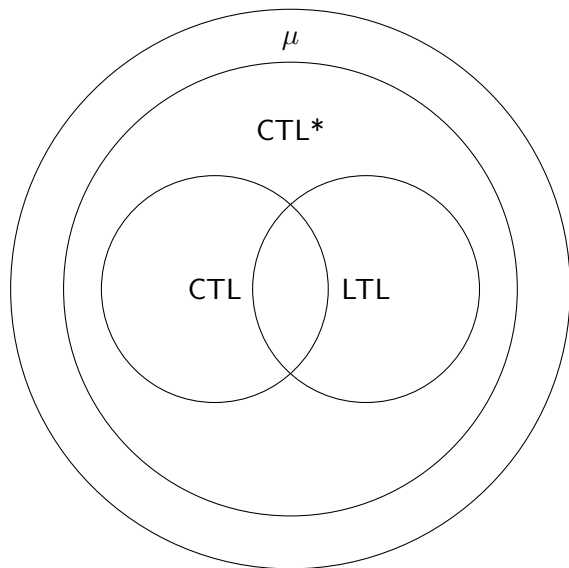
Example CTL* Formulas

$\forall \square a$




$\forall \diamond \square (a \cup b)$

$\forall \exists \square \diamond a$

Hierarchy of expressiveness



References

-  Christel Baier and Joost-Pieter Katoen. *Principles of Model Checking*. The MIT Press, 2008. ISBN: 978-0-262-02649-9.
-  Edmund M. Clarke. “The Birth of Model Checking”. In: *Lecture Notes in Computer Science* 5000 (2008), pp. 1–26.
-  Moshe Y. Vardi. “Branching vs. Linear Time: Final Showdown”. In: *Lecture Notes in Computer Science* 2031 (2001), pp. 1–22.