Model Checking and Games

Part V - Model checking CTL

Rüdiger Ehlers, Clausthal University of Technology

September 2019

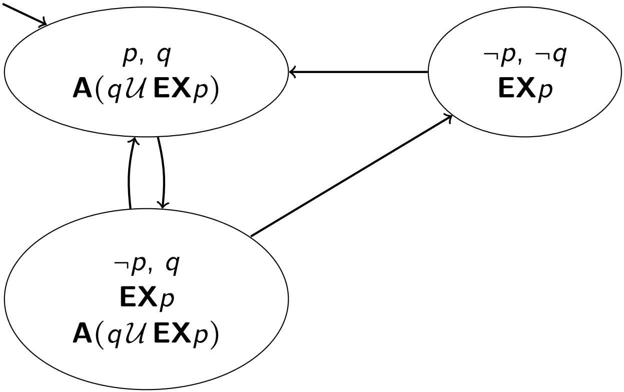
Model checking CTL vs. model checking LTL

Computation tree logic

- While LTL is a logic on *traces*, CTL is a logic on *states*
- This means that every node in the computation tree induced by a Kripke structure satisfies a CTL formula or not.
- Note that the subtrees of two nodes of the computation tree corresponding to the same Kripke structure are identical.
  $\rightarrow$ so we can label every node of a Kripke structure by the CTL (sub-)formulas that they satisfy!

Example using a simple Kripke structure

CTL formula of interest: $\mathbf{A}(q\,\mathcal{U}\,\mathbf{EX}p)$



How to find which states satisfy a CTL (sub-)formula?

$\neg p$ and p for $p \in \mathrm{AP}$

The Kripke structure is already labeled by the propositions holding from a state

$\mathbf{EX}\psi$

Label all states with $\mathbf{EX}\psi$ that have one successor state satifying ψ.

$\mathbf{AX}\psi$

Label all states with $\mathbf{AX}\psi$ for which *all* successor states satify ψ.

Labeling states with CTL subformulas (continued)

EFψ

Perform the following steps:

- Label every state satisfying ψ with **EF**ψ.
- Label every state with a successor state labeled by **EF**ψ by **EF**ψ as well.
- Repeat the previous step until no more states can be labeled with **EF**ψ.

AGψ

Let us use the fact that **AG**$\psi \equiv$ **EF**$\neg\psi$. Using this fact, we can execute the following approach:

- Initially, label every state satisfying ψ with **AG**ψ.
- Remove the **AG**ψ label of every state with a successor state not labelled by **AG**ψ.
- Repeat the previous step until no more state labels can be removed.

Labeling states with CTL subformulas (continued)

$\mathbf{E}(\psi \,\mathcal{U}\, \psi')$

Perform the following steps:

- Label every state satisfying ψ' with $\mathbf{E}(\psi \,\mathcal{U}\, \psi')$.
- Label every state satisfying ψ and having a successor labeled with $\mathbf{E}(\psi \,\mathcal{U}\, \psi')$ also as $\mathbf{E}(\psi \,\mathcal{U}\, \psi')$.
- Repeat the previous step until no more states can be labeled with $\mathbf{E}(\psi \,\mathcal{U}\, \psi')$.

$\mathbf{A}(\psi \,\mathcal{U}\, \psi')$

Perform the following steps:

- Label every state satisfying ψ' with $\mathbf{A}(\psi \,\mathcal{U}\, \psi')$.
- Label every state satisfying ψ and having only successor states labeled with $\mathbf{A}(\psi \,\mathcal{U}\, \psi')$ also as $\mathbf{A}(\psi \,\mathcal{U}\, \psi')$.
- Repeat the previous step until no more states can be labeled with $\mathbf{A}(\psi \,\mathcal{U}\, \psi')$.

Question

Main question

Rather than giving algorithms for each operator, can we somehow give a *uniform approach* that is parameterized by each temporal logic operator?

Answer

There is an encoding of each operator into *modal μ-calculus*, which gives a theoretical foundation to evaluating CTL formulas

Modal μ-calculus (short version!)

Syntax

Modal μ-calculus is an extension of propositional logic. For some given set of variable symbols $\mathcal{V}$, formulas in modal μ-calculus over some set of variables with defined values V and some set of atomic proposition AP is are defined as follows (for $p \in \mathrm{AP}$ and $x \in \mathcal{V} \setminus V$):

$$\begin{aligned}\psi(V,\mathrm{AP}) ::= \top \mid \bot \mid p \mid x \mid \Box\psi(V,\mathrm{AP}) \mid \Diamond\psi(V,\mathrm{AP}) \\ \mid \psi(V,\mathrm{AP}) \cup \psi(V,\mathrm{AP}) \mid \psi(V,\mathrm{AP}) \cap \psi(V,\mathrm{AP}) \\ \mid \mu x.\psi(V \cup \{x\},\mathrm{AP}) \mid \nu x.\psi(V \cup \{x\},\mathrm{AP})\end{aligned}$$

Semantics

Let a Kripke structure $\mathcal{K} = (S, S_0, T, \mathrm{AP}, L)$ be given. Let $V \subseteq \mathcal{V}$ be a subset of variables of $\mathcal{V}$ and $M : V \to 2^S$ be a valuation of the variables. A subformula for some variable assignment M always evaluates to some subset of S. We define the semantics of a subformula in *modal μ-calculus* (for $p \in \mathrm{AP}$ and $X \in \mathcal{V} \setminus V$): as follows:

$$\begin{aligned}
[\![\bot]\!]_M &= \varnothing \\
[\![\top]\!]_M &= S \\
[\![p]\!]_M &= \{s \in S \mid p \in L(s)\} \\
[\![x]\!]_M &= M(x) \\
[\![\Diamond\psi]\!]_M &= \{s \in S \mid \exists s' \in [\![\psi]\!]_M.(s,s') \in T\} \\
[\![\Box\psi]\!]_M &= \{s \in S \mid \forall s' \in S.(s,s') \in T \to s' \in [\![\psi]\!]_M\} \\
[\![\psi \cup \psi']\!]_M &= [\![\psi]\!]_M \cup [\![\psi']\!]_M \\
[\![\psi \cap \psi']\!]_M &= [\![\psi]\!]_M \cap [\![\psi']\!]_M \\
[\![\mu X.\psi]\!]_M &= \cup_{i=0}^{\infty}[\![\mu^i X.\psi]\!]_M \\
&\quad \text{for } \mu^0 X.\psi = \varnothing \text{ and} \\
&\quad \mu^i X.\psi = [\![\psi]\!]_{M \cup \{X \mapsto [\![\mu^{i-1}X.\psi]\!]\}} \text{ for } i > 0 \\
[\![\nu X.\psi]\!]_M &= \cup_{i=0}^{\infty}[\![\nu^i X.\psi]\!]_M \\
&\quad \text{for } \nu^0 X.\psi = S \text{ and} \\
&\quad \nu^i X.\psi = [\![X.\psi]\!]_{M \cup \{X \mapsto [\![\nu^{i-1}X.\psi]\!]\}} \text{ for } i > 0
\end{aligned}$$

Closed formulas

A *closed* mu-calculus formula is defined over the variables $V = \varnothing$ and hence can be evaluated on a Kripke structures. It can thus denote a specification.

A more thorough formalization of CTL operato

Using fixed point equation

We can formalize these rules as follows:

$\mathbf{AX}\psi$	$\Box\psi$
$\mathbf{EX}\psi$	$\Diamond\psi$
$\mathbf{AG}\psi$	$\nu X.\psi \cap \Box X$
$\mathbf{EG}\psi$	$\nu X.\psi \cap \Diamond X$
$\mathbf{AF}\psi$	$\mu X.\psi \cup \Box X$
$\mathbf{EF}\psi$	$\mu X.\psi \cup \Diamond X$
$\mathbf{A}(\psi \mathcal{U} \psi')$	$\mu X.\psi' \cup (\psi \cap \Box X)$
$\mathbf{E}(\psi \mathcal{U} \psi')$	$\mu X.\psi' \cup (\psi \cap \Diamond X)$
$\mathbf{A}(\psi \, \mathbf{R} \, \psi')$	$\nu X.\psi' \cap (\psi \cup \Box X)$
$\mathbf{E}(\psi \, \mathbf{R} \, \psi')$	$\nu X.\psi' \cap (\psi \cup \Diamond X)$

Summary / List of Concepts

- Model checking CTL by labeling Kripke structure states
- Simple algorithms for some sub-formulas
- Modal μ-calculus (a very short version!)
- Translating CTL to modal μ-calculus

References I