CTL Model Checking

Paul Jackson
Paul.Jackson@ed.ac.uk

University of Edinburgh

Formal Verification
Autumn 2023
CTL satisfaction using formula denotations

- In CTL model checking we ask the question: does

\[ M, s \models \phi \]

hold for all initial states \( S_0 \)?

- CTL model checking algorithms usually fix \( M = \langle S, \rightarrow, L \rangle \) and \( \phi \) and compute

\[ \lbrack \phi \rbrack_M = \{ s \in S \mid M, s \models \phi \} \]

"The denotation of \( \phi \) in model \( M \)"

- The CTL model checking question now becomes:

\[ S_0 \subseteq \lbrack \phi \rbrack_M \ ? \]

- Often \( M \) is implicit and we write \( \lbrack \phi \rbrack \) rather than \( \lbrack \phi \rbrack_M \)
Denotational semantics for CTL

Instead of defining $\llbracket \phi \rrbracket$ in terms of $\models \phi$, we can define it directly – recursively on the structure of $\phi$

\[
\begin{align*}
\llbracket \top \rrbracket & = S \\
\llbracket \bot \rrbracket & = \emptyset \\
\llbracket p \rrbracket & = \{ s \in S \mid p \in L(s) \} \\
\llbracket \neg \phi \rrbracket & = S - \llbracket \phi \rrbracket \\
\llbracket \phi \land \psi \rrbracket & = \llbracket \phi \rrbracket \cap \llbracket \psi \rrbracket \\
\llbracket \phi \lor \psi \rrbracket & = \llbracket \phi \rrbracket \cup \llbracket \psi \rrbracket
\end{align*}
\]

Since $\llbracket \phi \rrbracket$ is always a finite set, these are computable.
Denotational semantics for CTL: temporal connectives

\[
\begin{align*}
\llbracket \text{EX} \; \phi \rrbracket &= \text{pre}_\exists(\llbracket \phi \rrbracket) \\
\llbracket \text{AX} \; \phi \rrbracket &= \text{pre}_\forall(\llbracket \phi \rrbracket)
\end{align*}
\]

where

\[
\begin{align*}
\text{pre}_\exists(Y) &= \{ s \in S \mid \exists s' \in S. s \rightarrow s' \land s' \in Y \} \\
\text{pre}_\forall(Y) &= \{ s \in S \mid \forall s' \in S. s \rightarrow s' \Rightarrow s' \in Y \}
\end{align*}
\]

These are computable.

But what about the rest? E.g.

\[
\llbracket \text{EF} \; \phi \rrbracket = \{ s \in S \mid \exists \text{ path } \pi \text{ s.t. } s_0 = s. \exists i. s_i \models \phi \}
\]

does not suggest how to compute \( \llbracket \text{EF} \; \phi \rrbracket \)
Approximating $[[\text{EF } \phi]]$

Define

$$\text{EF}_0 \phi = \bot$$

$$\text{EF}_{i+1} \phi = \phi \lor \text{EX} \text{EF}_i \phi$$

Then

$$\text{EF}_1 \phi = \phi$$

$$\text{EF}_2 \phi = \phi \lor \text{EX} \phi$$

$$\text{EF}_3 \phi = \phi \lor \text{EX} (\phi \lor \text{EX} \phi)$$

$$\ldots$$

$s \in [[\text{EF}_i \phi]]$ if there exists a finite path $i$ states long starting from $s$ such that $\phi$ holds at some point on the path.

Fix a model $\mathcal{M}$ and let $n = |S|$. If there is a finite path with $k > n$ states on which $\phi$ holds somewhere, then there also is a finite path of $n$ states or fewer where $\phi$ holds somewhere. (*Proof: if $\phi$ occurs at position $\geq n$, repeatedly cut out segments between repeated states*)

Therefore, for all $k > n$, $[[\text{EF}_k \phi]] = [[\text{EF}_n \phi]]$
Computing $\llbracket \text{EF} \phi \rrbracket$

By a similar argument

$$\llbracket \text{EF} \phi \rrbracket = \llbracket \text{EF}_n \phi \rrbracket$$

Consider $\llbracket \text{EF}_n \rrbracket$ when the definition of $\text{EF}_n$ is expanded:

$$\llbracket \text{EF}_0 \phi \rrbracket = \emptyset$$

$$\llbracket \text{EF}_{i+1} \phi \rrbracket = \llbracket \phi \rrbracket \cup \text{pre}_\exists(\llbracket \text{EF}_i \phi \rrbracket)$$

We have here a way of computing $\llbracket \text{EF} \phi \rrbracket$.

In general, we can stop computing the recurrence as soon as we find

$$\llbracket \text{EF}_{k+1} \phi \rrbracket = \llbracket \text{EF}_k \phi \rrbracket$$

for $k \leq n$.

For efficient computation $k \ll n$ is desirable.
Approximating $\llbracket EG \phi \rrbracket$

Define

\[
\begin{align*}
\text{EG}_0 \phi & = T \\
\text{EG}_{i+1} \phi & = \phi \land \text{EX} \text{EG}_i \phi
\end{align*}
\]

Then

\[
\begin{align*}
\text{EG}_1 \phi & = \phi \\
\text{EG}_2 \phi & = \phi \land \text{EX} \phi \\
\text{EG}_3 \phi & = \phi \land \text{EX} (\phi \land \text{EX} \phi)
\end{align*}
\]

\[\vdots\]

$s \in \llbracket \text{EG}_i \phi \rrbracket$ if there exists a finite path $i$ states long starting from $s$ such that $\phi$ holds at every point on the path.

One can show $\forall k > n. \, \llbracket \text{EG}_k \phi \rrbracket = \llbracket \text{EG}_n \phi \rrbracket = \llbracket \text{EG} \phi \rrbracket$ (exercise) and so we can compute $\llbracket \text{EG} \phi \rrbracket$ using

\[
\begin{align*}
\llbracket \text{EG}_0 \phi \rrbracket & = S \\
\llbracket \text{EG}_{i+1} \phi \rrbracket & = \llbracket \phi \rrbracket \cap \text{pre}_\exists (\llbracket \text{EG}_i \phi \rrbracket)
\end{align*}
\]
Fixed-point theory

What is happening here is that we are computing fixed-points.

A set $X \subseteq S$ is a fixed point of a function $F \in \mathcal{P}(S) \rightarrow \mathcal{P}(S)$ iff $F(X) = X$.

We have that

$$\llbracket EF_n \phi \rrbracket = \llbracket EF_{n+1} \phi \rrbracket$$
$$= \llbracket \phi \lor EX EF_n \phi \rrbracket$$
$$= \llbracket \phi \rrbracket \cup \text{pre}_\exists(\llbracket EF_n \phi \rrbracket)$$

so $\llbracket EF_n \phi \rrbracket$ is a fixed point of

$$F(Y) \equiv \llbracket \phi \rrbracket \cup \text{pre}_\exists(Y) .$$

Also $\llbracket EF \phi \rrbracket$ is a fixed-point of $F$, since $\llbracket EF_n \phi \rrbracket = \llbracket EF \phi \rrbracket$.

More specifically, $\llbracket EF_n \phi \rrbracket$ and $\llbracket EF \phi \rrbracket$ are the least fixed point of $F$. 
Fixed-point theorem

A function $F \in \mathcal{P}(S) \to \mathcal{P}(S)$ is monotone iff $X \subseteq Y$ implies $F(X) \subseteq F(Y)$ of $S$.

Let $F^i(X) = F(F^{i-1}(X))$ for $i > 0$ and $F^0(X) = X$.

Given a collection of sets $C \subseteq \mathcal{P}(S)$, a set $X \in C$ is
- the least element of $C$ iff $\forall Y \in C. \ X \subseteq Y$,
- the greatest element of $C$ iff $\forall Y \in C. \ X \supseteq Y$.

Knaster-Tarski Theorem (special case)
Let $S$ be a set with $n$ elements and $F \in \mathcal{P}(S) \to \mathcal{P}(S)$ be a monotone function. Then
- $F^n(\emptyset)$ is the least fixed point of $F$, and
- $F^n(S)$ is the greatest fixed point of $F$

Proof: See p241 H&R

This theorem justifies $F^n(\emptyset)$ and $F^n(S)$ being fixed points of $F$ without the need, as before, to appeal to further details about $F$. 
Denotational semantics of temporal connectives

When $F \in \mathcal{P}(S) \to \mathcal{P}(S)$ is a monotone function, let us write

- $\mu Y. \ F(Y)$ for the least fixed point of $F$, and
- $\nu Y. \ F(Y)$ for the greatest fixed point of $F$.

With this notation, we can make the definitions

\[
\begin{align*}
\llbracket EF \phi \rrbracket &= \mu Y. [\phi] \cup \text{pre}_\exists(Y) \\
\llbracket EG \phi \rrbracket &= \nu Y. [\phi] \cap \text{pre}_\exists(Y) \\
\llbracket AF \phi \rrbracket &= \mu Y. [\phi] \cup \text{pre}_\forall(Y) \\
\llbracket AG \phi \rrbracket &= \nu Y. [\phi] \cap \text{pre}_\forall(Y) \\
\llbracket E[\phi \ U \psi] \rrbracket &= \mu Y. [\psi] \cup ([\phi] \cap \text{pre}_\exists(Y)) \\
\llbracket A[\phi \ U \psi] \rrbracket &= \mu Y. [\psi] \cup ([\phi] \cap \text{pre}_\forall(Y))
\end{align*}
\]

In every case the $F(Y)$ is monotone, so the Knaster-Tarski theorem assures us the fixed point exists and can be computed.
Further CTL Equivalences

The fixed-point characterisations of the CTL temporal operators justify the CTL equivalences

\[
\begin{align*}
EF \phi & \equiv \phi \lor EX EF \phi \\
EG \phi & \equiv \phi \land EX EG \phi \\
AF \phi & \equiv \phi \lor AX AF \phi \\
AG \phi & \equiv \phi \land AX EG \phi \\
E[\phi U \psi] & \equiv \psi \lor (\phi \land EX E[\phi U \psi]) \\
A[\phi U \psi] & \equiv \psi \lor (\phi \land AX A[\phi U \psi])
\end{align*}
\]
Fair CTL model checking

A fair version $E_\psi G \phi$ of $EG \phi$ holds in a state $s$ if there exists a path from $s$ such that

1. $\phi$ holds in every state of the path, and
2. $\psi$ holds infinitely often along the path.

We can define it using a greatest fixed-point operator $\nu$

$$E_\psi G \phi = \nu Z. \phi \land EX E[\phi U (\psi \land Z)]$$

How do we compute it?

- Its definition has nested fixed-points as there is a $\mu$ least fixed-point operator in the $E[-U-]$ definition
- In each iteration of the computation of the $\nu$ fixed-point, we have to complete a full set of iterations of the $\mu$ fixed-point computation.

Fair CTL model checking is useful both because of the extra expressivity it brings to CTL and because LTL model checking can be reduced to it.