CTL: Computation tree logic

A logic based on paths

\[ A ::= At \mid A_0 \land A_1 \mid A_0 \lor A_1 \mid \neg A \mid T \mid F \mid \text{EX } A \mid \text{EG } A \mid E[A_0 \cup A_1] \]

A path from state \( s \) is a maximal sequence of states

\[ \pi = (\pi_0, \pi_1, \ldots, \pi_i \ldots) \]

such that \( s = \pi_0 \) and \( \pi_i \rightarrow \pi_{i+1} \) for all \( i \).

\( s \models \text{EX } A \) iff Exists a path from \( s \) along which the next state satisfies \( A \)

\( s \models \text{EG } A \) iff Exists a path from \( s \) along which Globally each state satisfies \( A \)

\( s \models E[A \cup B] \) iff Exists a path from \( s \) along which \( A \) holds Until \( B \) holds
Derived assertions

\[ AX \, B \equiv \neg \text{EX} \, \neg B \]
\[ EF \, B \equiv E[\, T \, U \, B\,] \]
\[ AG \, B \equiv \neg \text{EF} \, \neg B \]
\[ AF \, B \equiv \neg \text{EG} \, \neg B \]
\[ A[ B \, U \, C ] \equiv \neg E[ \neg C \, U \, \neg B \, \land \, \neg C \,] \, \land \, \neg \text{EG} \, \neg C \]

The *Until* operator is **strict**
Want a modal-$\mu$ assertion equivalent to $\text{EG } A$. 
Want a modal-$\mu$ assertion equivalent to $\text{EG } A$.

Begin by writing a fixed point equation:

$$X = \varphi(X) \quad \text{where} \quad \varphi(X) = A \land (\lnot F \lor \lnot X)$$

Least or greatest fixed point? Consider:

Alternatively, consider the approximants for finite-state systems.
Want a modal-$\mu$ assertion equivalent to $\text{EG} \ A$.

Begin by writing a fixed point equation:

$$X = \varphi(X) \text{ where } \varphi(X) = A \land ([\neg]F \lor \langle \rangle X)$$

Least or greatest fixed point? Consider:

$$\mu X. A \land ([\neg]F \lor \langle \rangle X) = \emptyset$$
Want a modal-$\mu$ assertion equivalent to $\text{EG } A$.

Begin by writing a fixed point equation:

$$X = \varphi(X) \quad \text{where} \quad \varphi(X) = A \land (\lnot F \lor \lnot X)$$

Least or greatest fixed point? Consider:

$$\mu X. A \land (\lnot F \lor \lnot X) = \emptyset$$

$$\nu X. A \land (\lnot F \lor \lnot X) = \{s, t\}$$
Want a modal-$\mu$ assertion equivalent to $\text{EG} \ A$.

Begin by writing a fixed point equation:

$$X = \varphi(X) \quad \text{where} \quad \varphi(X) = A \land (\neg F \lor \langle \rangle X)$$

Least or greatest fixed point? Consider:

$$\mu X. A \land (\neg F \lor \langle \rangle X) = \emptyset$$

$$\nu X. A \land (\neg F \lor \langle \rangle X) = \{s, t\}$$

Alternatively, consider the approximants for finite-state systems.
A translation into modal-$\mu$

\[
\begin{align*}
\text{EX } a & \equiv (^{\neg})A \\
\text{EG } a & \equiv \nu Y.A \land ([^{\neg}]F \lor (^{\neg})Y) \\
E[a \cup b] & \equiv \mu Z.B \lor (A \land (^{\neg})Z)
\end{align*}
\]

Based on this, we get a translation of CTL into the modal-$\mu$ calculus.
Proposition

$$s \models \nu Y.A \land ([-]F \lor \langle \rangle Y)$$

in a finite-state transition system iff there exists a path \( \pi \) from \( s \) such that \( \pi_i \models A \) for all \( i \).

Proof:
Take \( \varphi( Y) \) \( \overset{\text{def}}{=} A \land ([-]F \lor \langle \rangle Y) \).

$$\nu Y.\varphi( Y) = \bigcap_{n \in \omega} \varphi^n(T) \text{ where } T \supseteq \varphi(T) \supseteq \cdots$$

since \( \varphi \) is monotonic and \( \cap \)-continuous due to the set of states being finite.

By induction, for \( n \geq 1 \)

$$s \models \varphi^n(T) \text{ iff there is a path of length } \leq n \text{ from } s \text{ along which all states satisfy } A \text{ and the final state has no outward transition}$$

or

$$\text{there is a path of length } n \text{ from } s \text{ along which all states satisfy } A \text{ and the final state has some outward transition}$$
Assuming the number of states is $k$, we have

$$\varphi^k(T) = \varphi^{k+1}(T)$$

and hence $\nu Y.\varphi(Y) = \varphi^k(T)$.

$s \models \nu Y.\varphi(Y)$ iff $s \models \varphi^k(T)$

iff there exists a maximal $A$ path of length $\leq k$ from $s$

or there exists a necessarily looping $A$ path of length $k$ from $s$.

\qed
Model checking modal-$\mu$

Assume processes are finite-state

- Brute force (+ optimizations) computes each fixed point
- Local model checking [Larsen, Stirling and Walker, Winskel]
Model checking modal-$\mu$

Assume processes are finite-state

- Brute force (+ optimizations) computes each fixed point
- Local model checking [Larsen, Stirling and Walker, Winskel]
  “Silly idea”

\[ p \in \nu X. \varphi(X) \iff p \in \varphi(\nu X. \varphi(X)) \]
Model checking modal-$\mu$

Assume processes are finite-state

- Brute force (+ optimizations) computes each fixed point
- Local model checking [Larsen, Stirling and Walker, Winskel]

*Reduction Lemma*

\[ p \in \nu X. \varphi(X) \iff p \in \varphi(\nu X.\{p\} \lor \varphi(X)) \]
Modal-μ for model checking

Extend the syntax with defined basic assertions and adapt the fixed point operator:

\[ A ::= U | T | F | \neg A | A \land B | A \lor B | \langle a \rangle A | \langle \neg \rangle A | \nu X . A \]

Semantics identifies assertions with subsets of states:

- \( U \) is an arbitrary subset of states
- \( T = S \)
- \( F = \emptyset \)
- \( \neg A = S \setminus A \)
- \( A \land B = A \cap B \)
- \( A \lor B = A \cup B \)
- \( \langle a \rangle A = \{ p \in S \mid \exists q. p \xrightarrow{a} q \land q \in A \} \)
- \( \langle \neg \rangle A = \{ p \in S \mid \exists q, a. p \xrightarrow{a} q \land q \in A \} \)
- \( \nu X \{ p_1, \ldots, p_n \}. A = \bigcup \{ U \subseteq S \mid U \subseteq A[U/X] \} \)
Modal-\(\mu\) for model checking

Extend the syntax with defined basic assertions and adapt the fixed point operator:

\[
A ::= U \mid T \mid F \mid \neg A \mid A \land B \mid A \lor B \mid \langle a \rangle A \mid \langle - \rangle A \mid \nu X\{p_1, \ldots, p_n\}.A
\]

Semantics identifies assertions with subsets of states:

- \(U\) is an arbitrary subset of states
- \(T = S\)
- \(F = \emptyset\)
- \(\neg A = S \setminus A\)
- \(A \land B = A \cap B\)
- \(A \lor B = A \cup B\)
- \(\langle a \rangle A = \{p \in S \mid \exists q. p \xrightarrow{a} q \land q \in A\}\)
- \(\langle - \rangle A = \{p \in S \mid \exists q, a. p \xrightarrow{a} q \land q \in A\}\)
- \(\nu X\{p_1, \ldots, p_n\}.A = \bigcup\{U \subseteq S \mid U \subseteq \{p_1, \ldots, p_n\} \cup A[U/X]\}\)
Modal-$\mu$ for model checking

Extend the syntax with defined basic assertions and adapt the fixed point operator:

$$A ::= U \mid T \mid F \mid \neg A \mid A \land B \mid A \lor B \mid \langle a \rangle A \mid \langle \neg \rangle A \mid \nu X\{p_1, \ldots, p_n\}.A$$

Semantics identifies assertions with subsets of states:

- $U$ is an arbitrary subset of states
- $T = S$
- $F = \emptyset$
- $\neg A = S \setminus A$
- $A \land B = A \cap B$
- $A \lor B = A \cup B$
- $\langle a \rangle A = \{p \in S \mid \exists q. p \xrightarrow{a} q \land q \in A\}$
- $\langle \neg \rangle A = \{p \in S \mid \exists q, a. p \xrightarrow{a} q \land q \in A\}$
- $\nu X\{p_1, \ldots, p_n\}.A = \bigcup\{U \subseteq S \mid U \subseteq \{p_1, \ldots, p_n\} \cup A[U/X]\}$

As before, $\mu X.A \equiv \neg \nu X.\neg A[\neg X/X]$ and now

$$\nu X.A = \nu X\{\}.A$$
Lemma

Let $\varphi : \mathcal{P}(S) \rightarrow \mathcal{P}(S)$ be monotonic. For all $U \subseteq S$,

$$U \subseteq \nu X. \varphi(X) \iff U \subseteq \varphi(\nu X.(U \cup \varphi(X)))$$

In particular,

$$p \in \nu X. \varphi(X) \iff p \in \varphi(\nu X.(\{p\} \cup \varphi(X))).$$
Model checking algorithm

Given a transition system and a set of basic assertions \( \{U, V, \ldots\} \):

\[
\begin{align*}
p \models U & \quad \rightarrow \quad \text{true} \quad \text{if } p \in U \\
p \models U & \quad \rightarrow \quad \text{false} \quad \text{if } p \notin U \\
p \models T & \quad \rightarrow \quad \text{true} \\
p \models F & \quad \rightarrow \quad \text{false} \\
p \models \neg B & \quad \rightarrow \quad \text{not}(p \models B) \\
p \models A \land B & \quad \rightarrow \quad p \models A \quad \text{and} \quad p \models B \\
p \models A \lor B & \quad \rightarrow \quad p \models A \quad \text{or} \quad p \models B \\
p \models \langle a \rangle B & \quad \rightarrow \quad q_1 \models B \quad \text{or} \quad \ldots \quad \text{or} \quad q_n \models B \\
p \models \nu X\{\bar{r}\}.B & \quad \rightarrow \quad \text{true} \quad \text{if } p \in \{\bar{r}\} \\
p \models \nu X\{\bar{r}\}.B & \quad \rightarrow \quad p \models B[\nu X\{p, \bar{r}\}.B/X] \quad \text{if } p \notin \{\bar{r}\}
\end{align*}
\]

Can use any sensible reduction technique for \textit{not}, \textit{or} and \textit{and} and.
Define the pure CCS process

\[ P \overset{\text{def}}{=} a.(a.\textit{nil} + a.P) \]

Check

\[ P \vdash \nu X.\langle a \rangle X \]

and check

\[ P \vdash \mu Y.[-]F \lor \langle - \rangle Y \]

Note:

\[ \mu Y.[-]F \lor \langle - \rangle Y \equiv \neg \nu Y.\neg([-]F \lor \langle - \rangle \neg Y)) \]
Well-founded induction

A binary relation $<$ on a set $A$ is **well-founded** iff there are no infinite descending chains

$$\ldots < a_n < \ldots < a_1 < a_0$$

**The principle of well-founded induction:**
Let $<$ be a well-founded relation on a set $A$. Let $P$ be a property on $A$. Then

$$\forall a \in A. \: P(a)$$

iff

$$\forall a \in A. \: ((\forall b < a. \: P(b)) \implies P(a))$$
Correctness and termination of the algorithm

Write $(p \Vdash A) = \text{true}$ iff $p$ is in the set of states determined by $A$.

**Theorem**

Let $p \in \mathcal{P}$ be a finite-state process and $A$ be a closed assertion. For any truth value $t \in \{\text{true}, \text{false}\}$,

$$(p \Vdash A) \rightarrow^* t \iff (p \Vdash A) = t$$
Proof sketch

For assertions $A$ and $A'$, take

$A'$ is a proper subassertion of $A$

$A' < A \iff \text{ or } A \equiv \nu X\{\bar{r}\}B \land \exists p \ A' \equiv \nu X\{\bar{r}, p\}B \land p \notin \bar{r}$

Want, for all closed assertions $A$,

$Q(A) \iff \forall q \in P. \forall t. (q \vdash A) \rightarrow^* t \iff (q \vdash A) = t$

We show the following stronger property on open assertions by well-founded induction:

$Q^+(A) \iff \forall$ closed substitutions for free variables

$B_1/X_1, \ldots, B_n/X_n: Q(B_1) \land \ldots \land Q(B_n) \implies Q(A[B_1/X_1, \ldots, B_n/X_n])$

The proof (presented in the lecture notes) centrally depends on the reduction lemma.