Introduction to Software Verification

Orna Grumberg

Lectures Material
winter 2015-16
Lecture 4
Model Checking

Automated formal verification:

A different approach to formal verification
Model Checking \cite{CE81,QS82}

An efficient procedure that receives:
- A finite-state model describing a system
- A temporal logic formula describing a property

It returns
- yes, if the system has the property
- no + Counterexample, otherwise
Building a model from a program
Back to the Mutual Exclusion Example

- Two processes with a joint Boolean signal `sem`
- Each process `P_i` has a variable `v_i` describing its state:
  - `v_i := N` Non critical
  - `v_i := T` Trying
  - `v_i := C` Critical
Mutual Exclusion Example

- Each process runs the following program:
  
P_i :: while (true) {
      if (v_i == N) v_i = T;
      else if (v_i == T && sem) { v_i = C; sem = 0; }
      else if (v_i == C) {v_i = N; sem = 1; }
  }

- The full program is: P_1||P_2
- Initial state: (v_1=N, v_2=N, sem)
- The execution is interleaving
Mutual Exclusion Example

- \( v_1 = N, v_2 = N, \text{sem} \)
- \( v_1 = T, v_2 = N, \text{sem} \)
- \( v_1 = N, v_2 = T, \text{sem} \)
- \( v_1 = C, v_2 = N, \neg \text{sem} \)
- \( v_1 = T, v_2 = T, \text{sem} \)
- \( v_1 = N, v_2 = C, \neg \text{sem} \)
- \( v_1 = C, v_2 = T, \neg \text{sem} \)
- \( v_1 = T, v_2 = C, \neg \text{sem} \)
We define atomic propositions: $\text{AP} = \{C_1, C_2, T_1, T_2\}$

A state is marked with $T_i$ if $v_i = T$

A state is marked with $C_i$ if $v_i = C$
• Property 1: $\text{AG}(C_1 \land C_2)$ ✓
• Property 2: $\text{AG}((T_1 \rightarrow FC_1) \land (T_2 \rightarrow FC_2))$ ✗
Temporal Logic
Kripke Structure $M = (S, R, L, S_0)$

Given $AP$ - finite set of atomic proposition

- $S$ - (finite) set of states
- $R \subseteq S \times S$ - total transition relation
  For every $s \in S$ there exists $s' \in S$ such that $(s, s') \in R$.
  Totality means that every path is infinite
- $L : S \to 2^{AP}$ - labeling function that associates every state with the atomic propositions true in that state
- $S_0 \subseteq S$ - set of initial states (optional)
\[ \pi = s_0, s_1, \ldots \text{ is a path in } M \text{ from a state } s \text{ if} \]

- \( s = s_0 \) and
- \( R(s_i, s_{i+1}) \) for every \( i \)

\[ \pi^i \text{ - the suffix of } \pi \text{ starting at } s_i \]
Temporal Logics

Express properties of event orderings in time

- **Linear Time**
  - Every moment has a unique successor
  - Infinite sequences (words)
  - Linear Time Temporal Logic (LTL)

- **Branching Time**
  - Every moment has several successors
  - Infinite tree
  - Computation Tree Logic (CTL)
Propositional Temporal Logic

$AP$ - a set of atomic propositions, $p \in AP$

Temporal operators:

- $Xp$
- $Gp$
- $Fp$
- $pUq$

Path quantifiers: $A$ for all path, $E$ there exists a path
CTL*

**State formulas:**
- \( p \in AP \)
- \( \neg g_1, g_1 \lor g_2, g_1 \land g_2 \) where \( g_1, g_2 \) are state formulas
- \( Ef, Af \) where \( f \) is a path formula

**Path formulas:**
- Every state formula \( g \) is a path formula
- \( \neg f_1, f_1 \lor f_2, f_1 \land f_2, Xf_1, Gf_1, Ff_1, f_1 U f_2 \) where \( f_1, f_2 \) are path formulas

**CTL* - set of all state formulas**
Semantics of CTL*

\( \pi = s_0, s_1, \ldots \) is a path in \( M \) if \( R(s_i, s_{i+1}) \) for every \( i \).

\( \pi^i \) - the suffix of \( \pi \) starting at \( s_i \).

State formulas:

- \( M, s \models p \iff p \in L(s) \)
- \( M, s \models Ef \iff \) there is a path \( \pi \) from \( s \) s.t. \( M, \pi \models f \)
- \( M, s \models Af \iff \) for every path \( \pi \) from \( s \), \( M, \pi \models f \)
Semantics of CTL*

\( \pi = s_0, s_1, \ldots \) is a path in \( M \) if \( R(s_i, s_{i+1}) \) for every \( i \).

\( \pi^i \) - the suffix of \( \pi \) starting at \( s_i \).

Path formulas:

- \( M, \pi \models g \), where \( g \) is a state formula \( \iff M, s_0 \models g \)
Semantics of CTL*

\[ \pi = s_0, s_1, \ldots \text{ is a path in } M \text{ if } R(s_i, s_{i+1}) \text{ for every } i. \]

\[ \pi^i \text{ - the suffix of } \pi \text{ starting at } s_i. \]

Path formulas:

- \( M, \pi \models g \), where \( g \) is a state formula \( \iff M, s_0 \models g \)
- \( M, \pi \models Xf \iff M, \pi^1 \models f \)
Semantics of CTL*

\[ \pi = s_0, s_1, \ldots \text{ is a path in } M \text{ if } R(s_i, s_{i+1}) \text{ for every } i. \]

\[ \pi^i - \text{the suffix of } \pi \text{ starting at } s_i. \]

Path formulas:
- \( M, \pi \models g \), where \( g \) is a state formula \( \iff M, s_0 \models g \)
- \( M, \pi \models Xf \iff M, \pi^1 \models f \)
- \( M, \pi \models Gf \iff \text{for every } k \geq 0, M, \pi^k \models f \)
Semantics of CTL*

\[ \pi = s_0, s_1, \ldots \] is a path in \( M \) if \( R(s_i, s_{i+1}) \) for every \( i \).
\( \pi^i \) - the suffix of \( \pi \) starting at \( s_i \).

Path formulas:

- \( M, \pi \models g \), where \( g \) is a state formula \( \iff \) \( M, s_0 \models g \)
- \( M, \pi \models Xf \) \( \iff \) \( M, \pi^1 \models f \)
- \( M, \pi \models Gf \) \( \iff \) for every \( k \geq 0 \), \( M, \pi^k \models f \)
- \( M, \pi \models Ff \) \( \iff \) there exists \( k \geq 0 \), s.t. \( M, \pi^k \models f \)
Semantics of CTL*

\( \pi = s_0, s_1, \ldots \) is a path in \( M \) if \( R(s_i, s_{i+1}) \) for every \( i \).

\( \pi^i \) - the suffix of \( \pi \) starting at \( s_i \).

\[
\begin{align*}
\textbullet & \ M, \pi \vdash G f \iff \text{for every } k \geq 0, M, \pi^k \models f \\
\textbullet & \ M, \pi \vdash F f \iff \text{there exists } k \geq 0, \text{s.t. } M, \pi^k \models f \\
\textbullet & \ M, \pi \vdash f_1 U f_2 \iff \text{there exists } k \geq 0, \text{s.t. } M, \pi^k \models f_2 \\
& \quad \text{and for every } 0 \leq j < k, M, \pi^j \models f_1
\end{align*}
\]
Semantics of CTL*

$\pi = s_0, s_1, \ldots$ is a path in $M$ if $R(s_i, s_{i+1})$ for every $i$.

$\pi^i$ - the suffix of $\pi$ starting at $s_i$.

Path formulas:

- $M, \pi \models g$, where $g$ is a state formula $\iff M, s_0 \models g$
- $M, \pi \models Xf \iff M, \pi^1 \models f$
- $M, \pi \models Gf \iff$ for every $k \geq 0$, $M, \pi^k \models f$
- $M, \pi \models Ff \iff$ there exists $k \geq 0$, s.t. $M, \pi^k \models f$
- $M, \pi \models f_1 U f_2 \iff$ there exists $k \geq 0$, s.t. $M, \pi^k \models f_2$
  and for every $0 \leq j < k$, $M, \pi^j \models f_1$
Semantics of Path Formulas

If $p,q$ are state formulas then:

- $Xp$
- $Gp$
- $Fp$
- $pUq$

But in the general case, they can be path formulas.
Semantics of CTL*

\[ M \models g \iff \text{for every initial state } s: \ M, s \models g \]
LTL/CTL/CTL*

**LTL** - state formulas of the form $A\psi$

- $\psi$ - path formula, contains **no** path quantifiers
- interpreted over infinite computation paths

**CTL** - state formulas where path quantifiers and temporal operators appear in pairs:

- $AG, AU, AF, AX, EG, EU, EF, EX$
- interpreted over infinite computation trees

**CTL** - Allows any combination of temporal operators and path quantifiers. Includes both LTL and CTL
LTL

State formulas:
• \( Af \) where \( f \) is a path formula

Path formulas:
• \( p \in AP \)
• \( \neg f_1, f_1 \lor f_2, f_1 \land f_2, Xf_1, Gf_1, Ff_1, f_1 U f_2 \) where \( f_1, f_2 \) are path formulas

LTL - set of all state formulas
CTL

CTL - set of all state formulas

• $p \in \text{AP}$

• $\neg g_1$, $g_1 \lor g_2$, $g_1 \land g_2$

• $\text{AG } g_1$, $\text{A } g_1 \cup g_2$, $\text{AF } g_1$, $\text{AX } g_1$

• $\text{EG } g_1$, $\text{E } g_1 \cup g_2$, $\text{EF } g_1$, $\text{EX } g_1$

where $g_1, g_2$ are state formulas
Examples (LTL)

• $AG \neg (\text{start} \wedge \neg \text{ready})$
• $AG (\text{req} \rightarrow \text{Fack})$
• A GF enabled
• A FG deadlock
• A (GF enabled $\rightarrow$ GF running)

Cannot express existential properties: “from any state the system can...”
Examples (CTL)

- EF (start ∧ ¬ ready)
- AG (req → AF ack)
- AG (AF enabled)
- AF (AG deadlock)
- AG (EF restart)
- AG (non_critical → EX trying)
- AG (try → A[try U succeed])
Illustration of CTL Semantics

EFp: "exists reachable state s.t."

EGp: "all reachable states...."

AFp:

AGp:
## Property types

<table>
<thead>
<tr>
<th></th>
<th>Universal</th>
<th>Existential</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Safety</strong></td>
<td>$AGp$</td>
<td>$EGp$</td>
</tr>
<tr>
<td><strong>Liveness</strong></td>
<td>$AFp$</td>
<td>$EFp$</td>
</tr>
</tbody>
</table>
Examples
Equivalence

• Path formulas $\psi_1, \psi_2$ are equivalent if:
  For every $M$ and path $\pi$
    $M, \pi \models \psi_1$ iff $M, \pi \models \psi_2$

• State formulas $\varphi_1, \varphi_2$ are equivalent if:
  For every $M$ and state $s$
    $M, s \models \varphi_1$ iff $M, s \models \varphi_2$
Expressiveness

¬, ∨, X, U, E suffice to express all CTL*:

• Ff ≡ true U f
• Gf ≡ ¬F (¬f)
• Af ≡ ¬E (¬f)

In CTL: EX, EG, EU are sufficient

• A [pUq] ≡ (¬EG ¬q) ∧ ¬E[¬q U (¬p ∧ ¬q)]
LTL vs. CTL

• **A (FG p)** has no equivalent in CTL
  “in all paths, p globally holds from some point on”

• Failed attempts:
  **AFAGP** : “in every path there is a point from which all reachable states satisfy p.”

All paths satisfy FGp
- $s_0,s_0,s_0,...$
- $s_0,s_0,...s_0,s_1,s_2,s_2,s_2,...$

But first one does not sat FAGp
LTL vs. CTL

• A (FG p) has no equivalent in CTL
  “in all paths, p globally holds from some point on”

• What about AFEGP?
  “in every path there is a point from which there is a path where p globally holds”

\[ s_0 \xrightarrow{p} s_1 \]

All paths satisfy FEGp
- since \( s_1 \) sat EGp
But \( s_0, s_1, s_0, s_1, s_0, s_1 \ldots \) does not sat FGp
LTL vs. CTL

- **AG (EFp)** has no equivalent in LTL
  - “all reachable states can reach p”

- Failed attempt:
  - \( AGFp \) : “in all paths, p holds infinitely many times.”

  All reachable states \((s_0, s_1)\) satisfy EFp

  But \(s_0, s_0, s_0, \ldots\) does not satisfy GFp
LTL and CTL vs. CTL*

• $E(GFp)$ has no equivalent in LTL or CTL