Introduction to Software Verification

Orna Grumberg

Lectures Material
winter 2018-19
Lecture 4
Model Checking

Automated formal verification:

A different approach to formal verification
Formal Verification

Given
• a model of a (hardware or software) system and
• a formal specification

does the system model satisfy the specification?

Not decidable!

To enable automation, we restrict the problem to a decidable one:
• **Finite-state** reactive systems
• **Propositional** temporal logics
Properties in Propositional Temporal Logic - Examples

• mutual exclusion:
  always ¬( cs₁ ∧ cs₂)

• non starvation:
  always (request ⇒ eventually granted)

• communication protocols:
  (¬ get-message) until send-message
Finite State Systems - Examples

- Hardware designs
- Controllers (elevator, traffic-light)
- Communication protocols (when ignoring the message content)
- High level (abstracted) description of non finite state systems
Model Checking \([CE81,QS82]\)

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
Model Checking

• Given a system and a specification, does the system satisfy the specification.
Model of a system
Kripke structure / transition system

Labeled by atomic propositions AP
(critical section, variable value...)

Reactive Systems
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
- $L: S \rightarrow 2^{AP}$ - 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 \( i \in \mathbb{N} \)

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

\( \mathcal{AP} \) - a set of atomic propositions, \( p \in \mathcal{AP} \)

Temporal operators:
- \( Xp \)
- \( Gp \)
- \( Fp \)
- \( pUq \)

Path quantifiers:
- \( A \) for all path
- \( E \) there exists a path
Example to demonstrate:

- Building a model from a program
- Properties
- Model checking
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 :: \quad \text{while (true)} \{
    \quad \text{if (} v_i = N \text{)} \quad v_i = T; \\
    \quad \text{else if (} v_i = T \text{ and sem) } \quad \{ v_i = C; \text{ sem = 0; } \}
    \quad \text{else if (} v_i = C \text{)} \quad \{ v_i = N; \text{ sem = 1; } \}
  \}

• The full program is: \( P_1 || P_2 \)

• Initial state: \( (v_1 = N, v_2 = N, \text{ sem}) \)

• 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: \( AP=\{C_1, C_2, T_1, T_2\} \)

• A state is marked with \( T_i \) if \( v_i=T \)

• A state is marked with \( C_i \) if \( v_i=C \)
- Property 1: $AG^{-}(C_1 \land C_2)$ ✓
- Property 2: $AG((T_1 \rightarrow FC_1) \land (T_2 \rightarrow FC_2))$ ✗
• Property 1: $AG_\downarrow(C_1 \land C_2)$
Property 1: $\text{AG}(C_1 \land C_2)$

$S_0$
• Property 1: $AG_{\downarrow}(C_1 \land C_2)$
• Property 1: $AG^{-1}(C_1 \land C_2)$
• Property 1: $AG_\perp(C_1 \land C_2)$

$S_3$
\[
\begin{align*}
&M \models AG \not\rightarrow (C_1 \land C_2) \\
&S_4 \subseteq S_0 \cup S_1 \cup S_2 \cup S_3
\end{align*}
\]
• Property 2: $AG^{-}(T_{1} \land T_{2})$
• Property 2: $AG_{\downarrow}(T_1 \land T_2)$
• Property 2: $AG(T_1 \land T_2)$
• $M \not\models AG \leftarrow (T_1 \wedge T_2)$
• *A violating state has been found*
Model checker returns a counterexample

\[ M \models \neg (T_1 \land T_2) \]
• Property 3: $AG((T_1 \rightarrow FC_1) \land (T_2 \rightarrow FC_2))$
Mutual Exclusion Example

\[ M \models AG EF (N_1 \land N_2 \land S_0) \]

*No matter where you are there is always a way to get to the initial state (restart)*
Temporal logics

We present 3 (propositional) temporal logics:

• $\text{CTL}^*$
• $\text{CTL}$
• $\text{LTL}$

$\text{CTL}$ and $\text{LTL}$ can be described as sub-logics of $\text{CTL}^*$
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
• $E f, Af$ 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*

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

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

Path formulas:

- \[ M, \pi \models g, \text{ where } g \text{ is a state formula } \iff M, s_0 \models g \]
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 $\iff$ $M, s_0 \models g$
- $M, \pi \models Xf \iff M, \pi^1 \models f$

![Diagram](image-url)
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:

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

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

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

Path formulas:

- \[ M, \pi \models g, \text{ where } g \text{ is a state formula } \iff M, s_0 \models g \]
- \[ M, \pi \models Xf \iff M, \pi^1 \models f \]
- \[ M, \pi \models Gf \iff \text{ for every } k \geq 0, M, \pi^k \models f \]
- \[ M, \pi \models Ff \iff \text{ there exists } k \geq 0, \text{ s.t. } M, \pi^k \models f \]
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 \( s_i \).

- \( M, \pi \models Gf \iff \) for every \( k \geq 0 \), \( M, \pi^k \models f \)
- \( M, \pi \models Ff \iff \) there exists \( k \geq 0 \), s.t. \( M, \pi^k \models f \)
- \( M, \pi \models f_1 \lor f_2 \iff \) 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 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*

\( M \models g \iff \text{for every initial state } s: M,s \models g \)
Examples
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:
• \( A f \) where \( f \) is a path formula

Path formulas:
• \( p \in AP \)
• \( \neg f_1, f_1 \lor f_2, f_1 \land f_2, Xf_1, Gf_1, Ff_1, f_1Uf_2 \) where \( f_1, f_2 \) are path formulas

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

CTL - set of all *state* formulas

- \( p \in \text{AP} \)
- \( \neg g_1, \; g_1 \lor g_2, \; g_1 \land g_2 \)
- \( \text{AX} \; g_1, \; \text{AG} \; g_1, \; \text{AF} \; g_1, \; \text{A} \; g_1 \; \text{U} \; g_2, \)
- \( \text{EX} \; g_1, \; \text{EG} \; g_1, \; \text{EF} \; g_1, \; \text{E} \; g_1 \; \text{U} \; g_2, \)

where \( g_1, g_2 \) are *state* formulas
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:
• $Xf_1, Gf_1, Ff_1, f_1Uf_2$ where $f_1, f_2$ are state formulas

CTL - set of all state formulas
Semantics of CTL

Recall: path $\pi = s_0, s_1, ...$

- $M, s \models p \iff p \in L(s)$ for $p \in AP$

- $M, s \models \varphi_1 \lor \varphi_2 \iff M, s \models \varphi_1$ or $M, s \models \varphi_2$

- $M, s \models EX \varphi \iff$ there is $s'$ s.t. $R(s, s')$ and $M, s' \models \varphi$

- $M, s \models EG \varphi \iff$ there is a path $\pi$ from $s$, s.t. for every $i \geq 0$, $M, s_i \models \varphi$