Introduction to Software Verification

Orna Grumberg

Lectures Material
winter 2015-16
Lecture 4
H* Proof System for Proving Termination (full correctness)

Termination rule for while-statement:

\[
\begin{align*}
p(\bar{x}, n) \land n > 0 & \rightarrow B(\bar{x}) \\
< p(\bar{x}, n) \land n > 0 > S < p(\bar{x}, n-1) > \\
p(\bar{x}, 0) & \rightarrow \neg B(\bar{x}) \\
\hline
< \exists n. p(\bar{x}, n) > \text{ while } B \text{ do } S \text{ od } < p(\bar{x}, 0) >
\end{align*}
\]
Completeness proof sketch for the termination rule for while-statement:

If \( \vdash \exists n. p(x,n) \) \( S \) \( \langle true \rangle \)
then \( \vdash_{H^*} \exists n. p(x,n) \) \( S \) \( \langle true \rangle \)

Full completeness proof in [Francez, program verification]
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 \( \neg (cs_1 \land cs_2) \)

• **non starvation:**
  
  always (request \( \Rightarrow \) eventually granted)

• **communication protocols:**
  
  (\( \neg \) 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
Example to demonstrate:

- Building a model from a program
- Properties
- Model checking
Mutual Exclusion Example

• Two processes with a joint Boolean signal \textit{sem}

• Each process $P_i$ has a local variable $v_i$ describing its state:
  - $v_i = \text{N}$ Non critical
  - $v_i = \text{T}$ Trying
  - $v_i = \text{C}$ Critical
Mutual Exclusion Example

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

- $v_1=N$, $v_2=N$, sem
- $v_1=T$, $v_2=N$, sem
- $v_1=N$, $v_2=T$, sem
- $v_1=C$, $v_2=N$, $\neg$sem
- $v_1=T$, $v_2=T$, sem
- $v_1=N$, $v_2=C$, $\neg$sem
- $v_1=C$, $v_2=T$, $\neg$sem
- $v_1=T$, $v_2=C$, $\neg$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: $AG_{\downarrow}(C_1 \land C_2)$ ✅
• Property 2: $AG((T_1 \rightarrow FC_1) \land (T_2 \rightarrow FC_2))$ ❌
• Property 1: $AG^{-1}(C_1 \land C_2)$
• Property 1: $AG^{-1}(C_1 \land C_2)$
• Property 1: $AG_\downarrow(C_1 \wedge C_2)$
• Property 1: $\text{AG}_{\downarrow}(C_1 \land C_2)$

$S_2$
• Property 1: $AG_{\downarrow}(C_1 \land C_2)$

$S_3$
\[ M \models AG \rightarrow (C_1 \land C_2) \]

\[ S_4 \subseteq S_0 \cup S_1 \cup S_2 \cup S_3 \]
• Property 2: $AG^\perp(T_1 \land T_2)$
• Property 2: $AG_r(T_1 \land T_2)$
• Property 2: $AG\neg(T_1 \land T_2)$
• M |≠ AG ⊢ (T_1 ∧ T_2)
• A violating state has been found
• $M \not\models AG \rightarrow (T_1 \land T_2)$

Model checker returns a counterexample
Property 3: $AG((T_1 \rightarrow FC_1) \land (T_2 \rightarrow FC_2))$