Introduction to Formal Methods

Lecture 8

Propagating Preconditions and Postconditions

Hossein Hojjat & Fatemeh Ghassemi

October 16, 2018

Review of Key Definitions

Hoare triple:

$$\{P\}\ r\ \{Q\} \Leftrightarrow \forall s, s' \in S. \big((s \in P \wedge (s, s') \in r) \rightarrow s' \in Q\big)$$

$\{P\}$ does not denote a singleton set containing P but is just a notation for an "assertion" around a command. Likewise for $\{Q\}$.

Strongest postcondition:

$$\mathit{sp}(P, r) = \{s' \mid \exists s. s \in P \wedge (s, s') \in r\}$$

Weakest precondition:

$$\mathit{wp}(r, Q) = \{s \mid \forall s'. (s, s') \in r \rightarrow s' \in Q\}$$

Hoare Rules: Summary

$$\frac{}{\vdash \{A[x := e]\}\ x := e\ \{A\}} \qquad \frac{\vdash \{A \wedge b\}\ c_1\ \{B\} \qquad \vdash \{A \wedge \neg b\}\ c_2\ \{B\}}{\vdash \{A\}\ \texttt{if}\ b\ \texttt{then}\ c_1\ \texttt{else}\ c_2\ \{B\}}$$

$$\frac{\vdash \{A \wedge b\}\ c\ \{A\}}{\vdash\ \{A\}\ \texttt{while}\ b\ \texttt{do}\ c\ \{A \wedge \neg b\}} \qquad \frac{\vdash \{A\}\ c_1\ \{C\} \qquad \vdash \{C\}\ c_2\ \{B\}}{\vdash \{A\}\ c_1\ ;\ c_2\ \{B\}}$$

$$\frac{\vdash A' \to A \quad \vdash \{A\}\ c\ \{B\} \quad \vdash\ B \to B'}{\vdash \{A'\}\ c\ \{B'\}}$$

Automating Reasoning in Hoare Logic

- Manually proving correctness is tedious
- We'd like to automate the tedious parts of program verification
- Idea: Assume an oracle gives loop invariants - we can then automate the rest of the reasoning
- This oracle can either be a human or a static analysis tool
  - (e.g., abstract interpretation)

Generating VCs: Forwards vs. Backwards

$$\{A\} \quad \text{program} \quad \{B\}$$

Precondition   Postcondition

- Two ways to generate verification conditions: forwards or backwards
- A forwards analysis starts from precondition and generates formulas to prove postcondition
- Forwards technique computes strongest postconditions (sp)
- In contrast, backwards analysis starts from postcondition and tries to prove precondition
- Backwards technique computes weakest preconditions (wp)

Some Notations

- If P is a formula on states and c a command, let $\textsf{sp}_F(P, c)$ be the formula version of the strongest postcondition operator
- $\textsf{sp}_F(P, c)$ is the formula Q that describes the set of states that can result from executing c in a state satisfying P

  $$\textsf{sp}_F(P, c) = Q$$

  implies

  $$\textsf{sp}((\{\vec{x} \mid P\}, \rho(c)) = \{\vec{x} \mid Q\}$$

- We denote the set of states satisfying a predicate by underscore s, i.e. for a predicate P, let P_s be the set of states that satisfies it:

  $$P_s = \{\vec{x} \mid P\}$$

Forward VCG: Using Strongest Postcondition

- Remember: $\{P_s\}\ \rho(c)\ \{Q_s\}$ is equivalent to

$$\mathit{sp}(P_s, \rho(c)) \subseteq Q_s$$

- A syntactic form of Hoare triple is $\{P\}\ c\ \{Q\}$
- That syntactic form is therefore equivalent to proving

$$\forall \vec{x}.(\mathit{sp}_F(P, c) \rightarrow Q)$$

- We can use the sp_F operator to compute verification conditions such as the one above
- We next give rules to compute $\mathit{sp}_F(P, c)$ for our commands such that
  $(\mathit{sp}_F(P, c) = Q)$ implies $(\mathit{sp}(P_s, \rho(c)) = Q_s)$

Assume Statement

Consider

- a precondition P, with $\mathit{FV}(P)$ among $\vec{x}$ and
- a property F, also with $\mathit{FV}(F)$ among $\vec{x}$

Assume Statement

$$
\begin{aligned}
\mathit{sp}(P_s, \rho(\texttt{assume}(F))) &= \mathit{sp}(P_s, \Delta_{F_s}) \\
&= \{\vec{x'} \mid \exists \vec{x} \in P_s.((\vec{x}, \vec{x'}) \in \Delta_{F_s})\} \\
&= \{\vec{x'} \mid \exists \vec{x} \in P_s.(\vec{x} = \vec{x'} \wedge \vec{x} \in F_s)\} \\
&= \{\vec{x'} \mid \vec{x'} \in P_s \wedge \vec{x'} \in F_s\} \\
&= P_s \cap F_s
\end{aligned}
$$

So:

$$\mathit{sp}_F(P, \texttt{assume}(F)) = P \wedge F$$

Assignment Statement

- Consider (for simplicity) we have a single variable $V = \{x\}$
- Let $e(x)$ be an expression on x

$$
\begin{aligned}
&\textit{sp}(P_s, \rho(x = e)) \\
&= \{x' \mid \exists x.\ x \in P_s \wedge (x, x') \in \rho(x = e)\} \\
&= \{x' \mid \exists x_0.\ (P[x := x_0] \wedge (x' = e[x := x_0])\}
\end{aligned}
$$

In general:

$$
\textit{sp}_F(P, x = e) = \exists x_0.(P[x := x_0] \wedge x = e[x := x_0])
$$

Exercise

Precondition: $\{x \geq 10 \wedge y \geq 5\}$

Code: `x = x + y - 5`

Exercise

Precondition: $\{x \geq 10 \land y \geq 5\}$

Code: `x = x + y - 5`

$$\mathit{sp}(x \geq 10 \land y \geq 5, \texttt{x = x + y - 5}) =$$
$$\exists x_0 . x_0 \geq 10 \land y \geq 5 \land x = x_0 + y - 5$$
$$\leftrightarrow y \geq 5 \land x \geq y + 5$$

Rules for Computing Strongest Postcondition

Sequential Composition

For relations we can prove

$$\mathit{sp}(P_s, r_1 \circ r_2) = \mathit{sp}(\mathit{sp}(P_s, r_1), r_2)$$

Therefore, define

$$\mathit{sp}_F(P, c_1; c_2) = \mathit{sp}_F(\mathit{sp}_F(P, c_1), c_2)$$

Nondeterministic Choice (Branches)

For relations we can prove

$$\mathit{sp}(P_s, r_1 \cup r_2) = \mathit{sp}(P_s, r_1) \cup \mathit{sp}(P_s, r_2)$$

Therefore define:

$$\mathit{sp}_F(P, c_1 \mathbin{[\!]} c_2) = \mathit{sp}_F(P, c_1) \vee \mathit{sp}_F(P, c_2)$$

Size of Generated Formulas

The size of the formula can be exponential because each time we have a nondeterministic choice, we double formula size:

$$\mathsf{sp}_F(P, (c_1 \mathbin{[\!]} c2); (c_3 \mathbin{[\!]} c_4)) =$$
$$\mathsf{sp}_F(\mathsf{sp}_F(P, c_1 \mathbin{[\!]} c_2), c_3 \mathbin{[\!]} c_4) =$$
$$\mathsf{sp}_F(\mathsf{sp}_F(P, c_1) \vee \mathsf{sp}_F(P, c_2), c_3 \mathbin{[\!]} c_4) =$$
$$\mathsf{sp}_F(\mathsf{sp}_F(P, c_1) \vee \mathsf{sp}_F(P, c_2), c_3) \vee \mathsf{sp}_F(\mathsf{sp}_F(P, c_1) \vee \mathsf{sp}_F(P, c_2), c_4)$$

Another Useful Characterization of *sp*

For any relation $\sigma \subseteq S \times S$ we define its range by

$$\mathit{ran}(\sigma) = \{s' \mid \exists s \in S.(s, s') \in \sigma\}$$

Lemma: suppose that

- $A \subseteq S$ and $r \subseteq S \times S$
- $\Delta = \{(s, s) \mid s \in S\}$

Then

$$\mathit{sp}(A, r) = \mathit{ran}(\Delta_A \circ r)$$

Proof of the Previous Fact

$$
\begin{aligned}
\mathit{ran}(\Delta_A \circ r) &= \mathit{ran}(\{(x,z) \mid \exists y.(x,y) \in \Delta_A \wedge (y,z) \in r\}) \\
&= \mathit{ran}(\{(x,z) \mid \exists y.x = y \wedge x \in A \wedge (y,z) \in r\}) \\
&= \mathit{ran}(\{(x,z) \mid x \in A \wedge (x,z) \in r\}) \\
&= \{z \mid \exists x.x \in A \wedge (x,z) \in r\} \\
&= \mathit{sp}(A, r)
\end{aligned}
$$

Reducing *sp* to Relation Composition

The following identity holds for relations:

$$\mathit{sp}(P_s, r) = \mathit{ran}(\Delta_P \circ r)$$

Based on this, we can compute $\mathit{sp}(P_s, \rho(c))$ in two steps:

1. compute formula $R(\texttt{assume}(P); c)$
2. existentially quantify over initial (non-primed) variables

Indeed, if F_1 is a formula denoting relation r_1, that is,

$$r_1 = \{(\vec{x}, \vec{x'}) \mid F_1(\vec{x}, \vec{x'})\}$$

then $\exists \vec{x}. F_1(\vec{x}, \vec{x'})$ is formula denoting the range of r_1:

$$\mathit{ran}(r_1) = \{\vec{x'} \mid \exists \vec{x}. F_1(\vec{x}, \vec{x'})\}$$

The resulting approach does not have exponentially large formulas.

Backward VCG: Using Weakest Preconditions

We derive the rules below from the definition of weakest precondition on sets and relations

$$\mathit{wp}(r, Q) = \{s \mid \forall s'.(s, s') \in r \rightarrow s' \in Q\}$$

Assume Statement

Suppose we have one variable x, and identify the state with that variable.

Note that $\rho(\texttt{assume}(F)) = \Delta_{F_s}$

$$\begin{aligned}
\mathit{wp}(\Delta_{F_s}, Q_s) &= \{x \mid \forall x'.(x, x') \in \Delta_{F_s} \rightarrow x' \in Q_s\} \\
&= \{x \mid \forall x'.(x \in F_s \wedge x = x') \rightarrow x' \in Q_s\} \\
&= \{x \mid x \in F_s \rightarrow x \in Q_s\} = \{x \mid F \rightarrow Q\}
\end{aligned}$$

Changing from sets to formulas, we obtain the rule for wp on formulas:

$$\mathit{wp}_F(\texttt{assume}(F), Q) = (F \rightarrow Q)$$

Assignment Statement

Consider the case of two variables. Recall that the relation associated with the assignment $x = e$ is

$$x' = e \wedge y' = y$$

Then we have, for formula Q containing x and y:

$$\begin{aligned}
&\textit{wp}(\rho(x = e), \{(x, y) \mid Q\}) \\
&= \{(x, y) \mid \forall x'.\forall y'.x' = e \wedge y' = y \rightarrow Q[x := x', y := y']\} \\
&= \{(x, y) \mid Q[x := e]\}
\end{aligned}$$

From here we obtain a justification to define:

$$\textit{wp}_F(x = e, Q) = Q[x := e]$$

Rules for Computing Weakest Preconditions

Sequential Composition

$$\mathit{wp}(r_1 \circ r_2, Q_s) = \mathit{wp}(r_1, \mathit{wp}(r_2, Q_s))$$

Same for formulas:

$$\mathit{wp}_F(c_1; c_2, Q) = \mathit{wp}_F(c_1, \mathit{wp}_F(c_2, Q))$$

Nondeterministic Choice (Branches)

In terms of sets and relations

$$\mathit{wp}(r_1 \cup r_2, Q_s) = \mathit{wp}(r_1, Q_s) \cap \mathit{wp}(r_2, Q_s)$$

In terms of formulas

$$\mathit{wp}_F(c_1 \;[]\; c2, Q) = \mathit{wp}_F(c1, Q) \wedge \mathit{wp}_F(c_2, Q)$$

Reference

Mike Gordon and Hélène Collavizza, “Forward with Hoare”,
Reflections on the Work of C. A. R. Hoare, 101–121, 2010.