CS 6110 S11 Lecture 19 Hoare Logic 11 March 2011

As we have seen, calculating preconditions by hand is hard and not always tractable. We will now define a logic which allows us to reason about when assertions hold and therefore (hopefully) bypass most of these kinds of computations.

1 Partial vs. Total Correctness

Two approaches to program verification are:

- *Partial correctness*: check if program is correct when it terminates. This is characterized by wlp and the Hoare logic we will define shortly. The termination issue is handled separately.
- *Total correctness*: ensure both that the program terminates and that it is correct. This is characterized by wp.

Partial correctness is the more common approach nowadays, since it separates the two issues of correctness and termination. These two verification tasks use very different methods, and it is helpful to separate them. Often partial correctness is easier to establish, and once this is done, the correctness conditions can be used in conjunction with a well-founded relation to establish termination.

2 Syntax of Hoare Logic

To define Hoare logic, we need to define the well-formed formulas in the logic. Hoare logic is built on top of another conventional logic, such as first-order logic. For now, let us take first-order logic as our base logic. Let $\varphi, \psi, \ldots$ denote first-order formulas. The formulas of Hoare logic are *partial correctness assertions* (PCA's), also known as *Hoare triples*. They look like

$$\{\varphi\}\, c\, \{\psi\}.$$

Informally, this means, "if φ holds before execution of c, and if c terminates, then ψ will hold upon termination." This is equivalent to

$$\varphi \quad\Rightarrow\quad \mathsf{wlp}\ c\ \psi.$$

2.1 Proof Rules

We will discuss the semantics of Hoare logic later. For now, we just give the deduction rules for the language IMP with programs

$$c \quad ::= \quad \mathsf{skip} \ \mid\ x := a \ \mid\ c_0; c_1 \ \mid\ \mathsf{if}\ b\ \mathsf{then}\ c_1\ \mathsf{else}\ c_2 \ \mid\ \mathsf{while}\ b\ \mathsf{do}\ c.$$

The rules are

(skip) $\{\varphi\}\,\mathsf{skip}\,\{\varphi\}$

(assignment) $\{\varphi\{a/x\}\}\, x := a\, \{\varphi\}$

$$\text{(sequential composition)}\quad \frac{\{\varphi\}\,c_1\,\{\psi\} \quad \{\psi\}\,c_2\,\{\sigma\}}{\{\varphi\}\,c_1; c_2\,\{\sigma\}}$$

$$\text{(conditional)}\quad \frac{\{b \wedge \varphi\}\,c_1\,\{\psi\} \quad \{\neg b \wedge \varphi\}\,c_2\,\{\psi\}}{\{\varphi\}\,\mathsf{if}\ b\ \mathsf{then}\ c_1\ \mathsf{else}\ c_2\,\{\psi\}}$$

$$\text{(while)}\quad \frac{\{b \wedge \varphi\}\,c\,\{\varphi\}}{\{\varphi\}\,\mathsf{while}\ b\ \mathsf{do}\ c\,\{\varphi \wedge \neg b\}}$$

$$\text{(weakening)}\quad \frac{\varphi \Rightarrow \varphi' \quad \{\varphi'\}\,c\,\{\psi'\} \quad \psi' \Rightarrow \psi}{\{\varphi\}\,c\,\{\psi\}}.$$

In the assignment rule, $\varphi\{a/x\}$ denotes the safe substitution of the arithmetic expression a for the variable x in φ. As with the λ-calculus, there may be bound variables in φ bound by quantifiers $\forall$ and $\exists$, and these may have to be renamed to avoid capturing the free variables of a. In the weakening rule, the operator $\Rightarrow$ is implication in the underlying logic. Note the parallels between these rules and the definitions of wlp.

3 Soundness and Completeness

A deduction system defines what it means for a formula to be *provable*, whereas a semantics defines what it means for a formula to be *true*. Given a logic with a semantics and a deduction system, two desirable properties are

- *Soundness*: Every provable formula is true.
- *Completeness*: Every true formula is provable.

Soundness is a basic requirement of any logical system. A logic would not be good for much if its theorems were false! With respect to the small-step or big-step semantics of IMP, Hoare logic is sound.

Completeness, on the other hand, is a much more difficult issue. Hoare logic, as presented, is not complete in general. However, it is *relatively complete* given an oracle for truth in the underlying logic, provided that logic is expressive enough to express weakest preconditions. This is a famous result of Stephen Cook. Although first-order logic is not expressive enough to express weakest preconditions over arbitrary domains of computation, it is expressive enough over the natural numbers. Therefore Hoare logic is relatively complete for IMP programs over the integers.

4 Semantics of IMP Revisited

4.1 Syntax of Commands

$$c \quad ::= \quad \mathsf{skip} \;\mid\; x := a \;\mid\; c_0 \,;\, c_1 \;\mid\; \mathsf{if}\ b\ \mathsf{then}\ c_1\ \mathsf{else}\ c_2 \;\mid\; \mathsf{while}\ b\ \mathsf{do}\ c.$$

4.2 Big-Step Rules

States are $\sigma : \mathit{Var} \to \mathbb{Z}$.

$$\text{(skip)}\quad \langle \mathsf{skip},\, \sigma\rangle \Downarrow \sigma$$

$$\begin{array}{ll}
\text{(assignment)} & \dfrac{\langle a,\,\sigma\rangle \Downarrow n}{\langle x := a,\,\sigma\rangle \Downarrow \sigma[n/x]} \\[2ex]
\text{(sequential composition)} & \dfrac{\langle c_0,\,\sigma\rangle \Downarrow \tau \quad \langle c_1,\,\tau\rangle \Downarrow \rho}{\langle c_0\;;\,c_1,\,\sigma\rangle \Downarrow \rho} \\[2ex]
\text{(conditional)} & \dfrac{\langle b,\,\sigma\rangle \Downarrow \mathsf{true} \quad \langle c_1,\,\sigma\rangle \Downarrow \tau}{\langle \mathsf{if}\ b\ \mathsf{then}\ c_1\ \mathsf{else}\ c_2,\,\sigma\rangle \Downarrow \tau} \quad \dfrac{\langle b,\,\sigma\rangle \Downarrow \mathsf{false} \quad \langle c_2,\,\sigma\rangle \Downarrow \tau}{\langle \mathsf{if}\ b\ \mathsf{then}\ c_1\ \mathsf{else}\ c_2,\,\sigma\rangle \Downarrow \tau} \\[2ex]
\text{(while loop)} & \dfrac{\langle b,\,\sigma\rangle \Downarrow \mathsf{false}}{\langle \mathsf{while}\ b\ \mathsf{do}\ c,\,\sigma\rangle \Downarrow \sigma} \quad \dfrac{\langle b,\,\sigma\rangle \Downarrow \mathsf{true} \quad \langle c,\,\sigma\rangle \Downarrow \tau \quad \langle \mathsf{while}\ b\ \mathsf{do}\ c,\,\tau\rangle \Downarrow \rho}{\langle \mathsf{while}\ b\ \mathsf{do}\ c,\,\sigma\rangle \Downarrow \rho}
\end{array}$$

4.3 Binary Relation Semantics

In the semantics of IMP, states $\sigma, \tau, \ldots$ are functions $\mathit{Var} \to \mathbb{Z}$. Let St denote the set of all states. For each program c, the big-step rules determine a binary input/output relation on St, namely

$$[\![c]\!] \;\stackrel{\triangle}{=}\; \{(\sigma,\tau) \mid \langle c,\,\sigma\rangle \Downarrow \tau\} \;\subseteq\; \mathsf{St} \times \mathsf{St}.$$

With this notation, we can express the big-step rules in terms of some basic operations on binary relations, namely *relational composition* ($\circ$) and *reflexive transitive closure* (*):

$$\begin{aligned}
R \circ S \;&\stackrel{\triangle}{=}\; \{(\sigma,\rho) \mid \exists\tau\ (\sigma,\tau) \in R,\ (\tau,\rho) \in S\} \\
R^* \;&\stackrel{\triangle}{=}\; \bigcup_{n \geq 0} R^n \;=\; \{(\sigma,\tau) \mid \exists\sigma_0,\ldots,\sigma_n\ \sigma = \sigma_0,\ \tau = \sigma_n,\ \text{and}\ (\sigma_i,\sigma_{i+1}) \in R,\ 0 \leq i \leq n-1\},
\end{aligned}$$

where $R^0 \stackrel{\triangle}{=} \{(\sigma,\sigma) \mid \sigma \in \mathsf{St}\}$ and $R^{n+1} \stackrel{\triangle}{=} R \circ R^n$. The big-step rules are equivalent to the following:

$$\begin{array}{lrcl}
\text{(skip)} & [\![\mathsf{skip}]\!] & = & \{(\sigma,\sigma) \mid \sigma \in \mathsf{St}\} \\
\text{(assignment)} & [\![x := a]\!] & = & \{(\sigma,\sigma[n/x]) \mid \langle a,\,\sigma\rangle \Downarrow n\} \\
\text{(sequential composition)} & [\![c_0\;;\,c_1]\!] & = & [\![c_0]\!] \circ [\![c_1]\!] \\
\text{(conditional)} & [\![\mathsf{if}\ b\ \mathsf{then}\ c_1\ \mathsf{else}\ c_2]\!] & = & [\![b]\!] \circ [\![c_1]\!] \;\cup\; [\![\neg b]\!] \circ [\![c_2]\!] \\
\text{(while loop)} & [\![\mathsf{while}\ b\ \mathsf{do}\ c]\!] & = & ([\![b]\!] \circ [\![c]\!])^* \circ [\![\neg b]\!],
\end{array}$$

where in the conditional and while loop,

$$\begin{aligned}
[\![b]\!] \;&\stackrel{\triangle}{=}\; \{(\sigma,\sigma) \mid \langle b,\,\sigma\rangle \Downarrow \mathsf{true}\} \\
[\![\neg b]\!] \;&\stackrel{\triangle}{=}\; \{(\sigma,\sigma) \mid \langle b,\,\sigma\rangle \Downarrow \mathsf{false}\} \;=\; [\![\mathsf{skip}]\!] - [\![b]\!].
\end{aligned}$$

In fact, this would have been a much more compact way to define them originally.

4.4 Semantics of Weakest Liberal Preconditions and Partial Correctness Assertions

We can now give a formal semantics for weakest liberal preconditions and Hoare partial correctness assertions. Let L denote the underlying logic (typically first-order logic). Write $\sigma \vDash \varphi$ if the formula φ of L is true in state σ, and write $\vDash \varphi$ if φ is true in all states. We wish to define what it means for a weakest liberal precondition assertion $\mathsf{wlp}\ c\ \psi$ to be true in a state σ, written $\sigma \vDash \mathsf{wlp}\ c\ \psi$, and for a partial correctness assertion $\{\varphi\}c\{\psi\}$ to be true, written $\vDash \{\varphi\}c\{\psi\}$.

$$\begin{aligned}
\sigma \vDash \mathsf{wlp}\ c\ \psi \quad &\stackrel{\triangle}{\iff} \quad \forall\tau\ \ (\sigma,\tau) \in [\![c]\!] \;\Rightarrow\; \tau \vDash \psi \\[1ex]
\vDash \{\varphi\}c\{\psi\} \quad &\stackrel{\triangle}{\iff} \quad \forall\sigma\ \ \sigma \vDash \varphi \;\Rightarrow\; \sigma \vDash \mathsf{wlp}\ c\ \psi \\
&\iff \quad \forall\sigma,\tau\ \ \sigma \vDash \varphi \wedge (\sigma,\tau) \in [\![c]\!] \;\Rightarrow\; \tau \vDash \psi.
\end{aligned}$$

4.5 Soundness and Relative Completeness of Hoare Logic

Let us write $\vdash \{\varphi\}c\{\psi\}$ to assert that $\{\varphi\}c\{\psi\}$ is provable in Hoare logic. Then soundness and relative completeness of Hoare logic are captured in the following theorems. The relative completeness result is due to Cook.

Theorem 1 (Soundness)**.** $\vdash \{\varphi\}c\{\psi\} \;\Rightarrow\; \vDash \{\varphi\}c\{\psi\}$.

Theorem 2 (Relative Completeness)**.** *Assume that the underlying logic L is* expressive *in the sense that all weakest liberal preconditions are expressible in L; that is, for each program c and formula ψ of L, there is a formula ψ' of L such that for all σ, $\sigma \vDash \psi'$ iff $\sigma \vDash \mathsf{wlp}\ c\ \psi$. Then $\vDash \{\varphi\}c\{\psi\} \;\Rightarrow\; \vdash \{\varphi\}c\{\psi\}$, provided we are allowed to assume all true formulas of L as axioms.*

Proof sketch. The proof is by structural induction on c. We will just sketch the proof for two cases, assignments and the while loop.

For an assignment $x := a$, suppose $\vDash \{\varphi\}x := a\{\psi\}$. Then $\forall\sigma \;\; \sigma \vDash \varphi \;\Rightarrow\; \sigma \vDash \mathsf{wlp}\ (x := a)\ \psi$. But $\mathsf{wlp}\ (x := a)\ \psi = \psi\{a/x\}$, so $\forall\sigma \;\; \sigma \vDash \varphi \;\Rightarrow\; \sigma \vDash \psi\{a/x\}$, therefore $\vDash \varphi \rightarrow \psi\{a/x\}$. We can thus assume $\vdash \varphi \rightarrow \psi\{a/x\}$, since we are allowed to take true formulas of L as axioms. Then $\vdash \{\psi\{a/x\}\}x := a\{\psi\}$ by the assignment rule of Hoare logic, thus $\vdash \{\varphi\}x := a\{\psi\}$ by the weakening rule of Hoare logic.

Now for the while loop. Suppose $\vDash \{\varphi\}\mathsf{while}\ b\ \mathsf{do}\ c\{\psi\}$. Then $\forall\sigma \;\; \sigma \vDash \varphi \;\Rightarrow\; \sigma \vDash \mathsf{wlp}\ (\mathsf{while}\ b\ \mathsf{do}\ c)\ \psi$. Since L is expressive, $\mathsf{wlp}\ (\mathsf{while}\ b\ \mathsf{do}\ c)\ \psi$ is equivalent to a formula ρ of L, and $\vDash \varphi \rightarrow \rho$. Since the programs

$$\mathsf{while}\ b\ \mathsf{do}\ c \qquad\qquad \mathsf{if}\ b\ \mathsf{then}\ (c; \mathsf{while}\ b\ \mathsf{do}\ c)\ \mathsf{else\ skip}$$

are semantically equivalent, we have

$$\begin{aligned}
\rho \;&\Leftrightarrow\; \mathsf{wlp}\ (\mathsf{while}\ b\ \mathsf{do}\ c)\ \psi \\
&\Leftrightarrow\; \mathsf{wlp}\ (\mathsf{if}\ b\ \mathsf{then}\ (c; \mathsf{while}\ b\ \mathsf{do}\ c)\ \mathsf{else\ skip})\ \psi \\
&\Leftrightarrow\; (b \;\Rightarrow\; \mathsf{wlp}\ c\ (\mathsf{wlp}\ (\mathsf{while}\ b\ \mathsf{do}\ c)\ \psi)) \wedge (\neg b \;\Rightarrow\; \mathsf{wlp\ skip}\ \psi) \\
&\Leftrightarrow\; (b \;\Rightarrow\; \mathsf{wlp}\ c\ \rho) \wedge (\neg b \;\Rightarrow\; \psi),
\end{aligned}$$

thus $\vDash \rho \wedge \neg b \rightarrow \psi$ and $\vDash \rho \wedge b \rightarrow \mathsf{wlp}\ c\ \rho$. The latter says exactly that $\vDash \{\rho \wedge b\}c\{\rho\}$. By the induction hypothesis, $\vdash \{\rho \wedge b\}c\{\rho\}$, and by the fact that we may assume all true formulas of L as axioms, $\vdash \varphi \rightarrow \rho$ and $\vdash \rho \wedge \neg b \rightarrow \psi$. Then

$$\begin{aligned}
\vdash \{\rho \wedge b\}c\{\rho\} \;&\Rightarrow\; \vdash \{\rho\}\mathsf{while}\ b\ \mathsf{do}\ c\{\rho \wedge \neg b\} && \text{by the Hoare while rule} \\
&\Rightarrow\; \vdash \{\varphi\}\mathsf{while}\ b\ \mathsf{do}\ c\{\psi\} && \text{by weakening.}
\end{aligned}$$

□