Scribe: Benjamin Ragheb and Alexa Sharp Lecturer: Dexter Kozen

1 Review of IMP Syntax

Recall that the IMP syntax is

$$\begin{array}{rcl} a & ::= & n \mid X \mid a_0 \ \mathbf{op} \ a_1 \\ b & ::= & \mathbf{true} \mid \mathbf{false} \mid \neg b \mid b_0 \wedge b_1 \mid b_0 \vee b_1 \\ c & ::= & \mathbf{skip} \mid X := a \mid \mathbf{if}\ b\ \mathbf{then}\ c_0\ \mathbf{else}\ c_1 \mid \mathbf{while}\ b\ \mathbf{do}\ c \mid c_0; c_1 \end{array}$$

2 Assertions

Assertions are extensions of the boolean language, and are often undecidable. We define the assertion sets:

Intvar integer variables i, j

AExpv extended arithmetic expressions $a, a_0, \ldots$

Assn extended boolean expressions A, B, C

And the syntax is

$$\begin{array}{rcl} a & ::= & n \mid X \mid i \mid a_0 + a_1 \mid a_0 - a_1 \mid a_0 \times a_1 \\ A & ::= & \mathbf{false} \mid a_0 \leq a_1 \mid A_0 \to A_1 \mid \forall i \,.\, A \end{array}$$

Intvar may seem redundant, but it is here because we want to allow quantification over Intvars but not over locations.

From this syntax we can construct everything we need. Some examples are

$$\begin{array}{rcl} \neg A & =^{def} & A \to \mathbf{false} \\ \exists i \,.\, A & =^{def} & \neg\forall i \,.\, \neg A \\ A_0 \vee A_1 & =^{def} & \neg A_0 \to A_1 \\ A_0 \wedge A_1 & =^{def} & \neg(\neg A_0 \vee \neg A_1) \\ a_0 = a_1 & =^{def} & a_0 \leq a_1 \wedge a_1 \leq a_0 \\ \mathbf{true} & =^{def} & \mathbf{false} \to \mathbf{false} \end{array}$$

2.1 Semantics of Assertions

The semantics of the assertion syntax defined above are as follows:

I: Intvar $\to Z$ interpretations
$\mathcal{A}v$: Aexpv $\to$ {interpretations} $\to \Sigma \to Z$

Instead of using Cv, we change the notation to $\sigma \models^I A$ (introduced by Tarski in the 40's), which means that given the state σ and interpretation I the assertion A is true. We can define $\models^I$ inductively, for example

$$\sigma \models^I \forall i \,.\, A \iff^{def} \sigma \models^{I\{n/i\}} A \quad \text{for all } n \in \mathbb{N}$$

3 Hoare Logic (CAR. Hoare '69)

The basic assertion in Hoare logic is the partial correcness assertion, which have the form $\{A\}c\{B\}$, where

$$\begin{aligned} c &= \text{program} \\ A &= \text{precondition} \\ B &= \text{postcondition} \end{aligned}$$

Intuitively, this means if A holds of the starting state, and if you execute c and if it halts then B is true of the halting state. Note that c does not necessarily have to halt.

We would like to extend $\sigma \models^I$ such that

$$\sigma \models^I \{A\}c\{B\} \Longleftrightarrow^{def} (\sigma \models^I A \Rightarrow \mathcal{C}[\![c]\!]\sigma \models^I B)$$

So for a fixed I, when the input state σ satisfies precondition A, the output state will satisfy postcondition B.
But what if c doesn't halt? Then this doesn't make sense. So we add $\perp$, the non-halting computation. We extend our states to $\Sigma_\perp =^{def} \Sigma \cup \perp$, and we define $\mathcal{C}[\![c]\!] = \perp$ if c does not halt. Also, $\perp \models^I A$ for all A. Then with these extensions, our assertion will hold.

3.1 Proof Rules (for Hoare Logic)

We have the following proof system for deriving valid partial correction assertions:

$$\{A\}\mathbf{skip}\{A\}$$

$$\{B[a/X]\}X := a\{B\}$$

$$\frac{\{A\}c_0\{B\} \quad \{B\}c_1\{C\}}{\{A\}c_0; c_1\{C\}}$$

$$\frac{\{A \wedge b\}c_0\{B\} \quad \{A \wedge \neg b\}c_1\{B\}}{\{A\}\mathbf{if}\ b\ \mathbf{then}\ c_0\ \mathbf{else}\ c_1\{B\}}$$

$$\frac{\{A \wedge b\}c\{A\}}{\{A\}\mathbf{while}\ b\ \mathbf{do}\ c\{A \wedge \neg b\}}$$

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

We call this last rule the "weakening rule", or the "consequence rule". We say $\vdash \{A\}c\{B\}$ ($\{A\}c\{B\}$ is *forced*) if $\{A\}c\{B\}$ follows from these rules.
Note that for assignment, the substitution occurs in the precondition.
The proof system defined above is both sound and complete, in the sense that they can prove anything modulo number theory.

4 Weakest Preconditions

We define the weakest precondition set as follows:

$$wp[\![c, B]\!] =^{def} \{\sigma \in \Sigma_\perp \mid \mathcal{C}[\![c]\!]\sigma \models^I B\}$$

We can produce a formula (assertion) $w[\![c, B]\!]$ in number theory, such that for any I

$$w[\![c, B]\!]^I =^{def} wp[\![c, B]\!]$$

Some examples are

$$\begin{aligned} w[\![X := a, B]\!] &= B[a/X] \\ w[\![c; c' B]\!] &= w[\![c, w[\![c' B]\!]]\!] \end{aligned}$$

Lemma
$\models \{w[\![c, B]\!]\} \subset B$
if $\models \{A\}c\{B\}$ then $\models A \to w[\![c, B]\!]$

Theorem (Relative Completeness) (Cook 1974)
Hoare logic is relatively complete: if $\models \{A\}c\{B\}$ then $\vdash \{A\}c\{B\}$.

"Relative" means relative to number theory - you get to assume true statements of number theory for free – use them in the weakening rule:

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

Proof Show $\vdash \{w[\![c, B]\!]\}c\{B\}$ by induction. Then show $\models \{A\}c\{B\} \Rightarrow \vdash \{w[\![c, B]\!]\}c\{B\}$ and $\models A \to w[\![c, B]\!] \Rightarrow \{A\}c\{B\}$ by weakening.

5 Proving Expressiveness

We define the weakest preconditions inductively:

$$\begin{aligned} w[\![\textbf{skip}, B]\!] &= B \\ w[\![X := a, B]\!] &= B[a/X] \\ w[\![c; c', B]\!] &= w[\![c, w[\![c', B]\!]]\!] \\ w[\![\textbf{if } b \textbf{ then } c \textbf{ else} c', B]\!] &= (b \wedge w[\![c, B]\!]) \vee (\neg b \wedge w[\![c', B]\!]) \\ w[\![\textbf{while } b \textbf{ do } c, B]\!] &= ?? \end{aligned}$$

The definition of while is the hard one – we need to use the coding power of number theory to encode arbitrary length sequences of integers as single integers (use the Chinese Remainder Theorem)