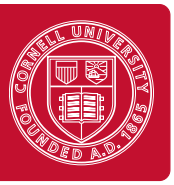
CS 4110 – Programming Languages and Logics
Lecture #10: Hoare Logic



1 Hoare Logic

How do we show that a partial correctness statement $\{P\}\ c\ \{Q\}$ holds? We know that $\{P\}\ c\ \{Q\}$ is valid if it holds for all stores and interpretations: $\forall \sigma, I.\ \sigma \vDash_I \{P\}\ c\ \{Q\}$. Furthermore, showing that $\sigma \vDash_I \{P\}\ c\ \{Q\}$ requires reasoning about the execution of command c (that is, $\mathcal{C}[\![c]\!]$), as indicated by the definition of validity.

It turns out that there is an elegant way of deriving valid partial correctness statements, without having to reason about stores, interpretations, and the execution of c. We can use a set of inference rules and axioms, called *Hoare* rules, to directly derive valid partial correctness statements. The set of rules forms a proof system known as Hoare logic.

$$\textsc{Skip}\ \frac{}{\{P\}\ \textbf{skip}\ \{P\}} \qquad \textsc{Assign}\ \frac{}{\{P[a/x]\}\ x := a\ \{P\}}$$

$$\textsc{Seq}\ \frac{\{P\}\ c_1\ \{R\} \qquad \{R\}\ c_2\ \{Q\}}{\{P\}\ c_1; c_2\ \{Q\}} \qquad \textsc{If}\ \frac{\{P \wedge b\}\ c_1\ \{Q\} \qquad \{P \wedge \neg b\}\ c_2\ \{Q\}}{\{P\}\ \textbf{if}\ b\ \textbf{then}\ c_1\ \textbf{else}\ c_2\ \{Q\}}$$

$$\textsc{While}\ \frac{\{P \wedge b\}\ c\ \{P\}}{\{P\}\ \textbf{while}\ b\ \textbf{do}\ c\ \{P \wedge \neg b\}}$$

The assertion P in the rule for while loops is essentially a loop invariant; it is an assertion that holds before and after each iteration, as shown in the premise of the rule. Therefore, it is both a pre-condition for the loop (because it holds before the first iteration); and also a post-condition for the loop (because it holds after the last iteration). The fact that P is both a pre- and post-condition for the while loop is reflected in the conclusion of the rule.

There is one more rule, the rule of consequence, which allows to strengthen pre-conditions and weaken post-conditions:

$$\textsc{Consequence}\ \frac{\vDash (P \Rightarrow P') \qquad \{P'\}\ c\ \{Q'\} \qquad \vDash (Q' \Rightarrow Q)}{\{P\}\ c\ \{Q\}}$$

These set of Hoare rules represent an inductive definition for a set of partial correctness statements $\{P\}\ c\ \{Q\}$. We will say that $\{P\}\ c\ \{Q\}$ is a theorem in Hoare logic, written $\vdash \{P\}\ c\ \{Q\}$, if we can build a finite proof tree for it.

2 Soundness and Completeness

At this point we have two kinds of partial correctness assertions:

- valid partial correctness statements $\vDash \{P\}\ c\ \{Q\}$, which hold for all stores and interpretations, according to the semantics of c, and
- Hoare logic theorems $\vdash \{P\}\ c\ \{Q\}$, that is, partial correctness statements that can be derived using the axioms and rules of Hoare logic.

The question is how do these sets relate to each other? More precisely, we have to answer two questions. First, is each Hoare logic theorem guaranteed to be valid partial correctness triple? In other words,

$$\text{does} \quad \vdash \{P\}\ c\ \{Q\} \quad \text{imply} \quad \vDash \{P\}\ c\ \{Q\}?$$

The answer is yes, and it shows that Hoare logic is sound. Soundness is important because it says that Hoare logic doesn't allow us to derive partial correctness assertions that actually don't hold. The proof of soundness requires induction on the derivations in $\vdash \{P\}\ c\ \{Q\}$ (we omit this proof).

The second question refers to the expressiveness and power of Hoare rules: can we always build a Hoare logic proof for each valid assertion? In other words,

$$\text{does} \quad \vDash \{P\}\ c\ \{Q\} \quad \text{imply} \quad \vdash \{P\}\ c\ \{Q\}?$$

The answer is a qualified yes: if $\vDash \{P\}\ c\ \{Q\}$ then there is a proof of $\{P\}\ c\ \{Q\}$ using the rules of Hoare logic, provided there are proofs for the validity of assertions that occur in the rule of consequence $\vDash (P \Rightarrow P')$ and $\vDash (Q' \Rightarrow Q)$. This result is known as the *relative completeness of Hoare logic* and is due to Cook (1974).

3 Example: Factorial

As an example illustrating how we can use Hoare logic to verify the correctness of a program, consider a program that computes the factorial of a number n:

$$\begin{array}{l}\{\mathsf{x} = n \wedge n > 0\} \\ \quad \mathsf{y} := 1; \\ \quad \textbf{while}\ \mathsf{x} > 0\ \textbf{do}\ \{ \\ \qquad \mathsf{y} := \mathsf{y} * \mathsf{x}; \\ \qquad \mathsf{x} := \mathsf{x} - 1 \\ \quad \} \\ \{\mathsf{y} = n!\}\end{array}$$

Because the derivation for this proof is somewhat large, we will go through the reasoning used to construct it step by step.

At the top level, the program is a sequence of an assignment and a loop. To use the SEQ rule, we need to find an assertion that holds after the assignment and before the loop. Examining the rule for while loops, we see that the assertion before the loop must be an invariant for the loop.

Inspecting the loop we see that it builds the factorial up in y starting with n, then multiplying it by $n-1$, then $n-2$, etc. At each iteration, x contains the next value multiplied into y, that is:

$$\mathsf{y} = n * (n-1) * ... * (\mathsf{x}+1)$$

If we multiply both sides of this equality by x! and re-write the equality we get $\mathsf{x}! * \mathsf{y} = n!$, which is an invariant for the loop. However, to make the proof go through, we need a slightly stronger invariant:

$$I = \mathsf{x}! * \mathsf{y} = n! \wedge \mathsf{x} \geq 0$$

Having identified a suitable loop invariant, let us take a step back and review where we are. We want to prove that our overall partial correctness specification is valid. To do this, we need to show two facts:

$$\{\mathsf{x} = n \wedge n > 0\}\ \mathsf{y} := 1\ \{I\} \tag{1}$$

$$\{I\}\ \textbf{while}\ \mathsf{x} > 0\ \textbf{do}\ \{\mathsf{y} := \mathsf{y} * \mathsf{x}; \mathsf{x} := \mathsf{x} - 1\}\ \{\mathsf{y} = n!\} \tag{2}$$

After showing that both (1) and (2) hold, we can use the rule SEQ to obtain the desired result.

To show (1), we use the ASSIGN axiom and obtain the following: $\{I[1/\mathsf{y}]\}\ \mathsf{y} := 1\ \{I\}$. Expanding this out, we obtain:

$$\{\mathsf{x}! * 1 = n! \wedge \mathsf{x} \geq 0\}\ y := 1\ \{\mathsf{x}! * \mathsf{y} = n! \wedge \mathsf{x} \geq 0\}$$

With the following implication,

$$\mathsf{x} = n \wedge n > 0 \implies \mathsf{x}! * 1 = n! \wedge \mathsf{x} \geq 0,$$

(which can be shown by an easy calculation) we obtain (1) using the rule CONSEQUENCE.

Now let us prove (2). To use the WHILE rule, we need to show that I is an invariant for the loop:

$$\{I \wedge \mathsf{x} > 0\}\ \mathsf{y} := \mathsf{y} * \mathsf{x}; \mathsf{x} := \mathsf{x} - 1\ \{I\} \tag{3}$$

We will show this by going backwards through the sequence of assignments:

$$\{(\mathsf{x}-1)! * \mathsf{y} = n! \wedge (\mathsf{x}-1) \geq 0\}\ \mathsf{x} := \mathsf{x} - 1\ \{I\} \tag{4}$$

$$\{(\mathsf{x}-1)! * \mathsf{y} * \mathsf{x} = n! \wedge (\mathsf{x}-1) \geq 0\}\ \mathsf{y} := \mathsf{y} * \mathsf{x}\ \{(\mathsf{x}-1)! * \mathsf{y} = n! \wedge (\mathsf{x}-1) \geq 0\} \tag{5}$$

Then, using the following implication:

$$I \wedge \mathsf{x} > 0 \implies (\mathsf{x}-1)! * \mathsf{y} * \mathsf{x} = n! \wedge (\mathsf{x}-1) \geq 0$$

we obtain (3) using CONSEQUENCE, (4), and (5). Thus, I is an invariant for the loop and so by WHILE we obtain,

$$\{I\}\ \textbf{while}\ \mathsf{x} > 0\ \textbf{do}\ \{\mathsf{y} := \mathsf{y} * \mathsf{x}; \mathsf{x} := \mathsf{x} - 1\}\ \{I \wedge \mathsf{x} \leq 0\}$$

To finish the proof, we just have to show

$$\begin{aligned} & I \wedge \mathsf{x} \leq 0 \implies \mathsf{y} = n! \\ i.e., \quad & \mathsf{x}! * \mathsf{y} = n! \wedge \mathsf{x} \geq 0 \wedge \mathsf{x} \leq 0 \implies \mathsf{y} = n! \end{aligned}$$

which holds as $\mathsf{x} \geq 0$ and $\mathsf{x} \leq 0$ implies $\mathsf{x} = 0$ and so $\mathsf{x}! = 1$. The result follows by CONSEQUENCE.