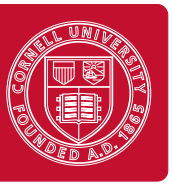
CS 4110 – Programming Languages and Logics
Lecture #8: Hoare Logic



Based on material by Stephen Chong, Greg Morrisett, Andrew Myers, George Necula, and Radu Rugina

1 Assertions

What can we say in pre-conditions and post-conditions? In the examples we saw in the last lecture, we used program variables, equality, logical variables (e.g., i), and conjunction ($\wedge$). What we allow in pre-conditions and post-conditions directly influences the sorts of program properties we can describe using partial correctness statements.

In this course, the language we will use for writing assertions is the set of logical formulas including comparisons between arithmetic expressions, standard logical operators (and, or, implication, negation), as well as quantifiers (universal and existential). Assertions may introduce logical variables that are different than the variables appearing in the program.

$$
\begin{aligned}
i, j &\in \mathbf{LVar} \\
a \in \mathbf{Aexp} &::= x \mid i \mid n \mid a_1 + a_2 \mid a_1 \times a_2 \\
P, Q \in \mathbf{Assn} &::= \textbf{true} \mid \textbf{false} \mid a_1 < a_2 \mid P_1 \wedge P_2 \mid P_1 \vee P_2 \mid P_1 \Rightarrow P_2 \mid \neg P \mid \forall i.\ P \mid \exists i.\ P
\end{aligned}
$$

Observe that the domain of boolean expressions $\mathbf{Bexp}$ is a subset of the domain of assertions. Notable additions over the syntax of boolean expression are quantifiers ($\forall$ and $\exists$). For instance, one can express the fact that variable x divides variable y using existential quantification: $\exists i.\ x \times i = y$.

2 Validity of assertions

Now we would like to describe what we mean by "assertion P holds in store σ". But to determine whether P holds or not, we need more than just the store σ (which maps program variables to their values); we also need to know the values of the logical variables. We describe those values using an interpretation I,

$$I : \mathbf{LVar} \to \mathbf{Int},$$

and define the function $\mathcal{A}_i[\![a]\!]$, which is like the denotation of expressions extended to logical variables in the obvious way:

$$
\begin{aligned}
\mathcal{A}_i[\![n]\!](\sigma, I) &= n \\
\mathcal{A}_i[\![x]\!](\sigma, I) &= \sigma(x) \\
\mathcal{A}_i[\![i]\!](\sigma, I) &= I(i) \\
\mathcal{A}_i[\![a_1 + a_2]\!](\sigma, I) &= \mathcal{A}_i[\![a_1]\!](\sigma, I) + \mathcal{A}_i[\![a_2]\!](\sigma, I)
\end{aligned}
$$

Now we can express the validity (or satisfiability) of assertion as a relation $\sigma \vDash_I P$ read as "P is valid in store σ under interpretation I," or "store σ satisfies assertion P under interpretation I." We will write $\sigma \nvDash_I P$ whenever $\sigma \vDash_I P$ doesn't hold.

We define the validity relation as follows:

$\sigma \vDash_I \textbf{true}$	(always)
$\sigma \vDash_I a_1 < a_2$	if $\mathcal{A}_i[\![a_1]\!](\sigma, I) < \mathcal{A}_i[\![a_2]\!](\sigma, I)$
$\sigma \vDash_I P_1 \wedge P_2$	if $\sigma \vDash_I P_1$ and $\sigma \vDash_I P_2$
$\sigma \vDash_I P_1 \vee P_2$	if $\sigma \vDash_I P_1$ or $\sigma \vDash_I P_2$
$\sigma \vDash_I P_1 \Rightarrow P_2$	if $s \nvDash_I P_1$ or $\sigma \vDash_I P_2$
$\sigma \vDash_I \neg P$	if $s \nvDash_I P$
$\sigma \vDash_I \forall i.\ P$	if $\forall k \in Int.\ \sigma \vDash_{I[i \mapsto k]} P$
$\sigma \vDash_I \exists i.\ P$	if $\exists k \in Int.\ \sigma \vDash_{I[i \mapsto k]} P$

We can now say that an assertion P is valid (written $\vDash P$) if it is valid in any store, under any interpretation: $\forall \sigma, I.\ \sigma \vDash_I P$.

Having defined validity for individual assertions, we now define the validity of partial correctness statements. We say that a partial correctness statement $\{P\}\ c\ \{Q\}$ is valid in store σ and interpretation I, written $\sigma \vDash_I \{P\}\ c\ \{Q\}$, if:

$$\forall \sigma'.\ \text{if } \sigma \vDash_I P \text{ and } \mathcal{C}[\![c]\!]\sigma = \sigma' \text{ then } \sigma' \vDash_I Q$$

Note that this definition depends on the execution of c in the initial store σ.

Finally, we can say that a partial correctness triple is valid (written $\vDash \{P\}\ c\ \{Q\}$), if it is valid in any store and interpretation:

$$\forall \sigma, I.\ \sigma \vDash_I \{P\}\ c\ \{Q\}.$$

Now we know what we mean when we say "assertion P holds" or "partial correctness statement $\{P\}\ c\ \{Q\}$ is valid."

3 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.

4 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).