Problem Sheet 1: Axiomatic Semantics

Chris Poskitt
ETH Zürich

Starred exercises (∗) are more challenging than the others.

1 Partial and Total Correctness

Recall the *Hoare triple* from lectures,

$$\{pre\}\ P\ \{post\}$$

where P denotes some program instructions, and $pre, post$ are respectively pre- and postconditions (*assertions* about the program state). The triple denotes a *specification* of the program, with respect to the given pre- and postconditions. Such a specification, in this course, can be interpreted in one of two different ways. If the specification holds in the sense of *partial correctness*, we will write:

$$\models \{pre\}\ P\ \{post\}$$

and if it holds in the sense of *total correctness*, we will write:

$$\models_{tot} \{pre\}\ P\ \{post\}$$

1.1 Exercises

Define, in English, the meaning of:

i. $\models \{pre\}\ P\ \{post\}$

ii. $\models_{tot} \{pre\}\ P\ \{post\}$

Which senses of correctness ($\models$, $\models_{tot}$, or neither) do the following triples hold under? Explain your judgements, giving counterexamples when they do not hold. (Formal proofs are *not* yet required. Assume a standard semantics for the program constructs. Assume that integer overflows cannot occur.)

iii. $\{x = 21 \wedge y = 5\}$ `skip` $\{y = 5\}$

iv. $\{y = 5\}$ `skip` $\{x = 21 \wedge y = 5\}$

v. $\{x > 10\}$ $\texttt{x := 2 * x}$ $\{x > 21\}$

vi. $\{x \geq 0 \wedge y > 1\}$ $\texttt{while x < y do x := x * x}$ $\{x \geq y\}$

vii. $\{x = 5\}$ $\texttt{while x > 0 do x := x + 1}$ $\{x < 0\}$

2 A Hoare Logic for Partial Correctness

In the previous section we reasoned about the correctness of triples in an *ad hoc* way. In this section we will be more rigorous, instead *proving* such triples using a formal system of proof rules called a *Hoare logic*.

In particular, we will consider a Hoare logic for partial correctness. If a Hoare triple $\{pre\}\ P\ \{post\}$ can be *proven* using the rules of this Hoare logic, we will write:

$$\vdash \{pre\}\ P\ \{post\}.$$

Provability and Validity

Note carefully the distinction between $\vdash$, *provability*, and $\models$, *validity*. An important property, called *soundness*, is that every triple provable in a Hoare logic indeed holds in the desired sense of correctness. For our partial correctness Hoare logic, you can safely assume that it is *sound*, i.e.:

$$\vdash \{pre\}\ P\ \{post\} \quad \text{implies} \quad \models \{pre\}\ P\ \{post\}.$$

Figure 1 contains the proof rules of a Hoare logic for partial correctness that should be used in the exercises overleaf. Let x denote a program variable, and e denote a side-effect free expression. Let the lower-case symbols p, p', q, q', r denote assertions, b a Boolean expression, and the upper-case symbols P, Q arbitrary programs.

A triple can be proven in this Hoare logic if it can be instantiated from an axiom ([ass] or [skip]), or if it can be derived as the conclusion of an inference rule ([comp], [if], [while], or [cons]). (A reminder of how to apply axioms and inference rules is given in the appendix of this problem sheet.)

$$\text{[ass]} \quad \vdash \{p[e/x]\}\ x := e\ \{p\}$$

$$\text{[skip]} \quad \vdash \{p\}\ \texttt{skip}\ \{p\}$$

$$\text{[comp]}\ \frac{\vdash \{p\}\ P\ \{r\} \quad \vdash \{r\}\ Q\ \{q\}}{\vdash \{p\}\ P;\ Q\ \{q\}}$$

$$\text{[if]}\ \frac{\vdash \{b \land p\}\ P\ \{q\} \quad \vdash \{\neg b \land p\}\ Q\ \{q\}}{\vdash \{p\}\ \texttt{if}\ b\ \texttt{then}\ P\ \texttt{else}\ Q\ \{q\}}$$

$$\text{[while]}\ \frac{\vdash \{b \land p\}\ P\ \{p\}}{\vdash \{p\}\ \texttt{while}\ b\ \texttt{do}\ P\ \{\neg b \land p\}}$$

$$\text{[cons]}\ \frac{p \Rightarrow p' \quad \vdash \{p'\}\ P\ \{q'\} \quad q' \Rightarrow q}{\vdash \{p\}\ P\ \{q\}}$$

Figure 1: A Hoare logic for partial correctness

2.1 Exercises

In the exercises that require a proof (i.e. $\vdash$), you should use the proof rules of the Hoare logic in Figure 1.

Hint: the appendix contains a recap on the different types of proof rules—*axioms* and *inference rules*—and how to apply them in order to prove a Hoare triple.

i. What does the rule of consequence, [cons], allow us to do in proofs? Why must we show that $p \Rightarrow p'$ and $q' \Rightarrow q$ are valid implications?

ii. Explain the intuition behind the axiom of assignment, [ass].

iii. Show that $\vdash \{x > 0\}$ $\texttt{x} := \texttt{x} + \texttt{1};\ \texttt{skip}$ $\{x > 1\}$.

iv. Show that $\vdash \{x = a \wedge y = b\}$ $\texttt{t} := \texttt{x};\ \texttt{x} := \texttt{x} + \texttt{y};\ \texttt{y} := \texttt{t}$ $\{x = a + b \wedge y = a\}$.

v. Explain the intuition behind the proof rule [while].

vi. Let INV denote $in + m = 250$. Show that:

$$\vdash \{INV\}\ \texttt{while}\ (\texttt{i} > \texttt{0})\ \texttt{do}\ \texttt{m} := \texttt{m} + \texttt{n};\ \texttt{i} := \texttt{i} - \texttt{1}\ \{INV\}.$$

vii. Define a proof rule for the Pascal construct:

$$\texttt{repeat}\ P\ \texttt{until}\ b$$

without reference to the [while] rule.

Hint: the loop body is always executed *at least* once.

viii. A new command has been added to the language: `surprise`. It has a very unusual semantics. Upon execution, `surprise` randomly chooses a variable in the program state, and randomly changes its assignment. Propose and justify an axiom for this command.

ix. ($*$) Propose an alternative, "forward" axiom of assignment in which a substitution is not applied to the precondition, i.e. replace the question marks in:

$$\vdash \{p\}\ x := e\ \{??\}.$$

Hint: the postcondition contains an existential quantifier.

Appendix: Applying Proof Rules

We review in this appendix how to prove triples in Hoare logics through the application of axioms and inference rules.

The proof rules [skip] and [ass] of Figure 1 are examples of *axioms*[1]. Axioms describe "self-evident" facts about programs, in the sense that they can be applied without additional reasoning. A triple $\{pre\}\ P\ \{post\}$ is *proven* in a Hoare logic, denoted $\vdash \{pre\}\ P\ \{post\}$, if it can be instantiated from such an axiom. For example, from the [skip] axiom, we can immediately prove any triple for `skip` that has the same pre- and postcondition, for example:

$$\vdash \{x = 5\}\ \texttt{skip}\ \{x = 5\}$$

$$\vdash \{\text{true}\}\ \texttt{skip}\ \{\text{true}\}$$

Most triples, of course, cannot simply be instantiated from axioms. For these, we must apply *inference rules*, such as [comp], [if], [while], and [cons] from Figure 1. Inference rules consist of at least one premise (given above the horizontal line), and exactly one conclusion (given underneath). If we can prove the premises, then we can deduce the triple in the conclusion. For example, to prove the triple:

$$\{x > 0\}\ \texttt{x := x + 1;}\ \texttt{skip}\ \{x > 1\}$$

simply instantiate [comp] with this triple as the conclusion, and then proceed to prove the premises:

$$\vdash \{x > 0\}\ \texttt{x := x + 1}\ \{r\}$$

$$\text{and} \quad \vdash \{r\}\ \texttt{skip}\ \{x > 1\}.$$

Here r is some assertion, to be determined, that holds at the midpoint of the two subprograms.

Visualising proofs is a matter of personal taste. For smaller proofs, I usually find it helpful to visualise them as *proof trees*, with axiom instantiations as the leaves, applications of inference rules as the body, and the triple we want to prove as the root. For example, the above could be visualised as follows (note that we have taken r to be $x < 1$):

$$\text{[comp]}\ \frac{\text{[??]}\ \dfrac{??}{\vdash \{x > 0\}\ \texttt{x := x + 1}\ \{x > 1\}} \qquad \text{[skip]}\ \dfrac{}{\vdash \{x > 1\}\ \texttt{skip}\ \{x > 1\}}}{\vdash \{x > 0\}\ \texttt{x := x + 1;}\ \texttt{skip}\ \{x > 1\}}$$

(The left-hand side of this proof tree is left incomplete because it forms part of an exercise!)

An inference rule we should give some special attention to is [cons], the rule of consequence. Two of its premises require showing—outside of the Hoare logic—that some logical implications are valid. We will discuss this further in the exercises.

[1] More precisely, *axiom schemata*, but most authors drop the latter part in their terminology.