Problem Sheet 9: Program Proofs

Chris Poskitt
ETH Zürich

1 Axiomatic Semantics

This section provides some additional questions on Hoare logic. Some proof rules are provided in Figure 1.

i. Devise an axiom for the command $\texttt{havoc}(\texttt{x}_0, \ldots, \texttt{x}_n)$, which assigns arbitrary values to the variables $\texttt{x}_0, \ldots, \texttt{x}_n$.

ii. Write a program that computes the factorial of a natural number stored in variable `x` and assigns the result to variable `y`. Prove that the program is correct using our Hoare logic.

iii. Define a proof rule for the `from-until-loop` construct.

iv. Consider the following annotated program, where A is an array indexed from 1 with elements of type G, n is an integer variable storing A's size, k is another integer variable, v is a variable of type G initialised to some fixed value, and *found* is a Boolean variable.

$\{n \geq 0\}$

```
from
    k := n
    found := False
until found or k < 1 loop
    if A[k] = v then
        found := True
    else
        k := k − 1
    end
end
```

$\{(found \Longrightarrow 1 \leq k \leq n \wedge A[k] = v) \wedge (\neg found \Longrightarrow k < 1)\}$

(a) What does the program do? In particular, what does the value of k represent on exit?

(b) Prove the triple using the axioms and inference rules of Hoare logic.

v. Sarah Proofgood has successfully shown that given an arbitrary program P and postcondition *post*, the triple:

$$\{\mathrm{WP}[P, post]\}\ P\ \{post\}$$

can be proven in our Hoare logic, i.e. $\vdash \{\mathrm{WP}[P, post]\}\ P\ \{post\}$. Here, $\mathrm{WP}[P, post]$ is an assertion expressing the *weakest (liberal) precondition* relative to P and *post*; that is, the weakest condition that must be satisfied for P to establish *post* (without guaranteeing termination).

Using Sarah's result, show that *any* valid triple $\models \{p\}\ P\ \{q\}$ is provable in our Hoare logic, i.e. $\vdash \{p\}\ P\ \{q\}$.

2 Separation Logic

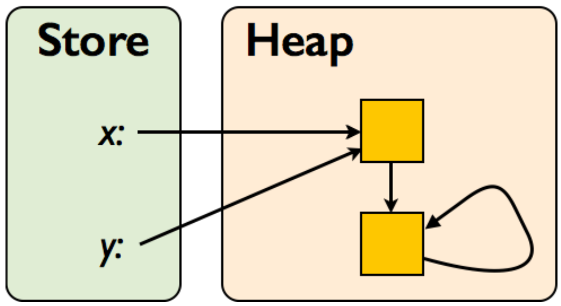
This section provides some additional practice on using separation logic. The small axioms and frame rule of separation logic are given in Figure 2.

i. Are the following assertions satisfiable? Justify your answers.

$$p * \neg p$$

$$x = y * \neg(x = y)$$

ii. Consider the following program state:



Which of the following assertions does this state satisfy? For the assertions it does not satisfy: why not?

(a) $\exists v.\ x \mapsto v * v \mapsto v$

(b) $\exists v.\ x \mapsto v * v \mapsto v * y \mapsto v$

(c) $y \mapsto \_$

(d) $(x = y) \wedge (y \mapsto \_ * \text{true})$

(e) $(x = y) * \text{true}$

iii. Starting from precondition {emp}, apply the axioms and inference rules of separation logic to derive a postcondition expressing exactly the contents of the store and heap at termination (assume that `x` and `y` are the only variables). Then, depict the post-state using the store and heap diagrams presented in the lectures.

```
x := cons(5,9);
y := cons(6,7);
x := [x];
[y+1] := 9;
dispose(y);
```

iv. A well-formed binary tree t is defined by the grammar:

$$t \triangleq n \mid (t_1, t_2)$$

i.e. t can be either a leaf, which is a single number n, or an internal node with a left subtree t_1 and a right subtree t_2. Consider the following definition of the inductive predicate $tree(t, i)$ which asserts that i is a pointer to a well-formed binary tree t:

$$\begin{aligned} tree(n, i) &\triangleq i \mapsto n \\ tree((t_1, t_2), i) &\triangleq \exists l, r.\ i \mapsto l, r * tree(t_1, l) * tree(t_2, r) \end{aligned}$$

Using these definitions, give a proof outline of the following triple. There must be at least one assertion between every two commands.

$\{tree((1, t), i)\}$

```
x := [i];
[i] := 2;
y := [i+1];
dispose(i);
dispose(x);
dispose(i+1);
```

$\{tree(t, y)\}$

Appendix: Proof Rules

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

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

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

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

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

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

Figure 1: A Hoare logic for partial correctness

$$\vdash \{e \mapsto \_\}\ [e] := f\ \{e \mapsto f\}$$

$$\vdash \{e \mapsto \_\}\ \texttt{dispose}(e)\ \{\text{emp}\}$$

$$\vdash \{X = x \wedge e \mapsto Y\}\ x := [e]\ \{e[X/x] \mapsto Y \wedge Y = x\}$$

$$\vdash \{\text{emp}\}\ x := \texttt{cons}(e_0, \ldots, e_n)\ \{x \mapsto e_0, \ldots, e_n\}$$

$$\frac{\vdash \{p\}\ P\ \{q\}}{\vdash \{p * r\}\ P\ \{q * r\}}$$

side condition: no variable modified by P appears free in r

Figure 2: The small axioms and frame rule of separation logic