Assignment 4 (Written): Semantics and Hoare Logic

17-654/17-754: Analysis of Software Artifacts
Jonathan Aldrich (`jonathan.aldrich@cs.cmu.edu`)

Due: Thursday, February 14, 2008 (5:30 pm)

100 points total

Paper users turn in a file named <username>-17654-A4.{txt,pdf,doc}, where username is your Andrew id, or hand in a paper response in class. SASyLF tool users should turn in a .zip file named as above, containing all of their solution files (654-asst4-pt1.slf, 654-asst4-pt2.slf, 654-asst4-pt3.slf, and answers.{txt,pdf,doc} for question 7). At the top of the document (or for tool users, the first file), state your name, Andrew id, and how long you spent on the assignment.

Assignment Objectives:

- Simulate WHILE programs using formal semantics, and prove simple properties of their behavior
- Write with specification constructs including pre- and post-conditions, loop invariants and variant functions
- Prove small programs correct using Hoare logic techniques

1 Formal Semantics (50 points)

Question 1 (10 points).

Produce a proof showing how the following expression reduces to a value in the context of the variable environment given, using the big-step semantics for WHILE expressions given in class.

Your proof should be detailed, i.e. show each inference rule applied until an axiom is used at the top of each branch.

$\{n \mapsto 5, y \mapsto 2\} \vdash n + 5 == y * (10 - n) \Rightarrow true$

Question 2 (10 points).

Produce a detailed proof showing how the WHILE statement below executes in the empty environment. Again, your proof should be detailed, showing each inference rule that is applied until axioms are reached. Include the proof for any relevant expression evaluations.

$\{\} \vdash n := 1; y := n + 2 \Rightarrow \{n \mapsto 1, y \mapsto 3\}$

Question 3 (30 points).

Consider the following WHILE program:

$i := n;$
$r := 0;$
$while\ (0 < i)\ do$
$\quad i := i - 1;$
$\quad r := r + m$

Using the semantics of WHILE, prove that for all natural numbers (numbers greater than or equal to zero) nat_1, nat_2 in the environment $\{n \mapsto nat_1, m \mapsto nat_2\}$, the program above executes to produce the environment $\{n \mapsto nat_1, m \mapsto nat_2, i \mapsto 0, r \mapsto nat_1 * nat_2\}$. Your proof should include all statement execution rules and state explicitly any premises of these rules. However, you do not have to prove facts about expression evaluation or mathematical truths–just stating what would need to be proved is enough for this assignment.

Hint: as with the proof for factorial done in lecture, you will need to use induction to prove the while loop correct. You may find the following lemma useful:

Lemma loopMultiplies: For all nat_1, nat_2, nat_3, we have:

$\{n \mapsto nat_1,\ m \mapsto nat_2,\ i \mapsto nat_3,\ r \mapsto nat_2 * (nat_1 - nat_3), \}$
$\vdash while\ 0 > i\ do\ i := i - 1;\ r := r + m$
$\Downarrow \{n \mapsto nat_1,\ m \mapsto nat_2,\ i \mapsto 0,\ r \mapsto nat_2 * nat_1\}$

The sample solution took about 26 judgments. If your proof is considerably longer than this, you are either giving the details of expression evaluation (which is not necessary for question 3, although it is necessary for question 2) or you may be on the wrong track.

2 Hoare Logic (50 points)

Question 4 (10 points).

Prove the following WHILE program correct:

$\{$*true*$\}$
$x_1 := nat_1;$
if $x_1 > 10$ *then* $x_1 := 10$ *else skip*
$\{x_1 \leq 10\}$

Show your work, i.e. the intermediate information computed by weakest preconditions between every two statements. SASyLF users, prove the code correct using Hoare Logic (but using "by solve" for each of the proof obligations).

Question 5 (10 points).

Consider the following WHILE program:

$x_1 := 0;$
$x_2 := 0;$
while $(x_1 < nat_1)$ *do*
  $x_1 := x_1 + 1;$
  $x_2 := x_2 + nat_2$

For the while program given above, state a (a) precondition, (b) postcondition, and (c) the loop invariant. The postcondition should precisely define the value of x_2 in terms of nat_1 and nat_2.

Question 6 (20 points).

Using weakest preconditions, state the proof obligations that assure for the program above that (a) the invariant is initially true on entry to the loop, (b) the loop invariant holds after execution of each loop, and (c) the postcondition holds. Show your work, i.e. the intermediate information computed by weakest preconditions between every two statements. SASyLF users, prove the code correct using Hoare Logic (but using "by solve" for each of the proof obligations).

Question 7 (10 points).

Which of the proof techniques above—semantics-based and Hoare Logic-based—do you prefer, and why? Are there any advantages of your second choice, whatever it is?