ETH
Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

D. Basin and P. Müller

Formal Methods and Functional Programming

Exercise Sheet 11: Axiomatic Semantics

Submission deadline: May 23th, 2011

Please submit your solution before **9:15am** on the submission date specified above. Solutions can be submitted via e-mail or by using the boxes to the left of **RZ F1**. Make sure that the first page always contains your name, the exercise sheet number as well as your tutor's name and the weekday (Tuesday or Wednesday) of your exercise group. Don't forget to staple your pages if you submit more than one page.

Assignment 1 - loop invariants

Consider the following program `s`:

```
i := 0;
r := 1;
while i < k do
  i := i + 1;
  r := r * n
end
```

In this exercise we want to find an invariant for the while loop in `s`. Recall that a loop invariant is a formula that holds before the loop, and that is preserved by the loop body.

(a) Which of the following formulas are invariants of this loop? You may assume that `k` stores a positive integer at the beginning of the program. For those which are not, show why this is not the case, that is where the proof fails.

$$\mathtt{i} \geq 0 \tag{1}$$

$$\mathtt{i} > 0 \tag{2}$$

$$\mathtt{i} \geq 0 \wedge \mathtt{r} = \mathtt{n}^{\mathtt{i}} \tag{3}$$

$$\mathtt{i} \geq 0 \wedge \mathtt{i} \leq \mathtt{k} \wedge \mathtt{r} = \mathtt{n}^{\mathtt{i}} \tag{4}$$

$$\mathtt{i} \geq 0 \wedge \mathtt{i} < \mathtt{k} \wedge \mathtt{r} = \mathtt{n}^{\mathtt{i}} \tag{5}$$

Hint: You might want to consider the axiomatic semantics rule for while loops.

(b) Prove in axiomatic semantics, that the above program s computes $\mathtt{n}^{\mathtt{k}}$. More formally, show that

$$\vdash \{\, \mathtt{k} \geq 1 \ \wedge K = \mathtt{k} \,\}\ s\ \{\, \mathtt{r} = \mathtt{n}^K \,\}$$

Hint: You might want to use one of the formulas from above (which are loop invariants) as you loop invariant in you proof.

Assignment 2 - rule of consequence

Recall the rule of consequence as presented in the lecture

$$\frac{\{\, \mathbf{P}' \,\}\ s\ \{\, \mathbf{Q}' \,\}}{\{\, \mathbf{P} \,\}\ s\ \{\, \mathbf{Q} \,\}} \text{ if } \mathbf{P} \Rightarrow \mathbf{P}' \text{ and } \mathbf{Q}' \Rightarrow \mathbf{Q}$$

and compare it to the following *unsound* variation

$$\frac{\{\, \mathbf{P}' \,\}\ s\ \{\, \mathbf{Q}' \,\}}{\{\, \mathbf{P} \,\}\ s\ \{\, \mathbf{Q} \,\}} \text{ if } \mathbf{P}' \Rightarrow \mathbf{P} \text{ and } \mathbf{Q} \Rightarrow \mathbf{Q}'$$

Give textual arguments why the first rule is sound and why the second one is not and support your argumentation in the second case with two counter-examples.

Assignment 3 - program correctness

Consider the following program s computing the quotient and the remainder of $\mathtt{x}/\mathtt{y}$.

```
z := 0;
while y <= x do
  z := z + 1;
  x := x - y
end
```

Prove that $\vdash \{\, \mathtt{x} = X \wedge \mathtt{y} = Y \,\}\ s\ \{\, X = \mathtt{x} + Y \cdot \mathtt{z} \wedge Y > \mathtt{x} \,\}$.

Hint: You might need to find a suitable loop invariant.

Assignment 4 - trivial postcondition

Show, by structural induction on the statement s, that $\vdash \{\, \mathbf{P} \,\}\ s\ \{\, \text{true} \,\}$ for all statements s and all properties $\mathbf{P}$.

Assignment 5 - Headache of the week

Consider the following program s computing the greatest common divisor (gcd) of two given positive integers:

```
b := x;
c := y;
while b # c do
  if b < c then
    c := c - b
  else
    b := b - c
  end
end;
z := b
```

Convince yourself that the program terminates when x and y store positive integers.

Tasks:

(a) Formalise the claim that the above program computes the gcd of x and y as pre- and postcondition $\mathbf{P}$ and $\mathbf{Q}$, respectively.

(b) Find an invariant for the loop.

(c) Show that $\vdash \{ \mathbf{P} \}\ s\ \{ \mathbf{Q} \}$.

Recall the definition of the gcd:
Let x, y be positive integers. The number z is the greatest common divisor of x and y iff $z|x$ and $z|y$ and there is no z', with $z' > z$, such that $z'|x$ and $z'|y$. Here, $z|x$ means that z divides x, i.e., $z \cdot k = x$, for some $k \in \mathbb{N}$.

Hint: Consider using a relationship between the input variables x, y and the 'loop' variables b, c as part of your loop invariant.