ETH
Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

D. Basin and P. Müller

Formal Methods and Functional Programming

Exercise Sheet 12: Axiomatic Semantics 2

Submission deadline: May 30th, 2010

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.

Extra Note

For the headache (and maybe assignment 1), you may want to work on an inner loop and then "plug in" your proof to a larger context. If you choose to do this then (as discussed in the exercises), you should make use of the frame rule (not required for the exam):

$$\frac{\{\ P\ \}\ s\ \{\ Q\ \}}{\{\ P \wedge R\ \}\ s\ \{\ Q \wedge R\ \}}\ (\text{if } \mathit{FV}(s) \cap \mathit{FV}(R) = \emptyset)$$

Furthermore, you may need to eliminate extra logical variables from your inner proof. e.g., If we have proven $\{\ P \wedge x = X_0\ \}\ s\ \{\ Q\ \}$ and want to use it in a larger proof in which the logical variable X_0 does not occur, you can employ the following recipe:

(a) Find an expression e for the value of x in the precondition of x. The expression e comes from analysis of the code which comes before s - it should not mention X_0 (and, for simplicity, we assume e will also not have variables in common with s).

(b) You can then assume that $\vdash \{\ P[x/X_0] \wedge x = e\ \}\ s\ \{\ Q[e/X_0]\ \}$.

(c) Use the frame rule further to conjoin any extra information which needs to be preserved across execution of s.

(d) Use the resulting triple in the larger proof, when dealing with s.

None of the note above is necessary for the exam - it just might help you construct large examples. If you'd like to read about this in more detail, there are some extra notes in a separate document online.

Assignment 1

Let s be the following statement:

```
y := 1;
z := 0;
while z<x do
  y := y * 2;
  z := z + 1
end
```

(a) What might be a suitable loop invariant?

Hint: Mention all of the variables mentioned in the loop.

(b) Find a suitable loop variant.

(c) Prove that $\vdash \{ \, \mathtt{x} = 10 \, \} \; s \; \{ \Downarrow \mathtt{y} = 1024 \, \}$.

Assignment 2

Show that (for all statements s_1,s_2, and for all predicates P and Q):

$$\vdash \{ \, P \, \} \; s_1; s_2 \; \{ \Downarrow Q \, \} \quad \Leftrightarrow \quad \textit{there exist } P',Q',R' \textit{ such that: } \begin{cases} P \Rightarrow P' \\ Q' \Rightarrow Q \\ \{ \, P' \, \} \; s_1 \; \{ \Downarrow R' \, \} \\ \{ \, R' \, \} \; s_2 \; \{ \Downarrow Q' \, \} \end{cases}$$

Assignment 3

This question concerns termination and the Zune bug, as discussed in the lectures.

(a) Suppose that, for some statement s, the triple $\{ \, \textit{true} \, \} \; s \; \{ \Downarrow \textit{true} \, \}$ can be derived. What does this tell us about s?

(b) Let s' be the following IMP statement:

```
while (365 < days) do
  if (L(year)) then
    if (366 < days) then
      days = days - 366; year = year + 1
    else
      skip
    end
```

```
  else
    days = days - 365; year = year + 1
  end
end
```

We assume that L(year) may be used as a boolean expression. Using days as a loop variant, attempt to derive that $\vdash \{\ \mathit{true}\ \}\ s'\ \{\ \Downarrow \mathit{true}\ \}$. Where does your proof fail?

(c) Let s'' be the following (corrected) IMP statement:

```
while (L(year) and 366 < days or not L(year) and 365 < days) do
  if (L(year)) then
      days = days - 366
  else
      days = days - 365;
  end;
  year += 1
end
```

Show that $\vdash \{\ \mathit{true}\ \}\ s''\ \{\ \Downarrow \mathit{true}\ \}$.

Assignment 4

Show that, for all statements s_1,s_2 and s_3, and for all predicates P and Q:

$$\vdash \{\ P\ \}\ (s_1; s_2); s_3\ \{\ \Downarrow Q\ \} \quad \Rightarrow \quad \vdash \{\ P\ \}\ s_1; (s_2; s_3)\ \{\ \Downarrow Q\ \}$$

Assignment 5 - Headache of the week

Recall the first exercise of sheet 7, in which you wrote a program to compute the floor of the Mth root of N (when $M > 0$ and $N \geq 0$ are both integers). This question requires an IMP statement to perform the same task - given that the values M and N are stored in variables x and y respectively, define an IMP statement s which is guaranteed to terminate, and which, on termination, will have stored in a variable z the value $\lfloor \sqrt[M]{N} \rfloor$. You may choose s to be any IMP statement which you believe achieves this goal - you might like to use your solution to sheet 7, or the sample solution.

Prove, that

$$\vdash \{\ x = M \wedge y = N \wedge M > 0 \wedge N \geq 0\ \}\ s\ \{\ \Downarrow z^M \leq N \wedge N < (z{+}1)^M\ \}$$

Assignment 6

In the lecture, you have seen parts of the modelling language *Promela*. In this and later assignments you will use Promela and the model checker *Spin* for modelling and analysing concurrent programs. You can find detailed information about how to install the model checker Spin on your computer at the webpage `http://spinroot.com/spin/Man/README.html`.

Some useful information for running Spin:

- If Spin is invoked without any options, it performs a *random simulation*. For example,

  ```
  $ spin foo.pr
  ```

  performs a random simulation for the Promela model specified in the file `foo.pr`.

- The command line option `-a` generates a protocol *specific analyzer*. The output is written into a C file pan.c, which you can, e.g., compile with the GNU C compiler. Running the compiled file will explore the state space of the Promela model and check whether the model might deadlock. The assertions in the model are also checked.

  ```
  $ spin -a foo.pr
  $ gcc -o pan pan.c
  $ ./pan
  ```

Note that there is also a graphical interface for Spin, called iSpin, which you can also download from the Spin webpage `http://spinroot.com`. We recommend that you also install iSpin on your computer. See `http://spinroot.com/spin/Man/Manual.html#S` for more information.

The purpose of this assignment is to get familiar with Promela and the model checker Spin.

(a) Consider again the following statement (which you already saw on previous exercise sheets).

```
y := 0;
while x > 0 do
  y := y + x;
  x := x - 2
end
```

Write a model in Promela to check if the statement starts in a state σ with $\sigma(\mathrm{x}) = 3$, it will reach a state σ' with $\sigma'(\mathrm{y}) = 4$.

(b) Write a model in Promela to verify that the following program will result in a state in which x has either the value 1 or 4.

$$x := 1 \,[]\, x := 2; x := x + 2$$

(c) Write a model in Promela to verify that the following program will result in a state in which x has one of the values 1 or 3 or 4.

$$x := 1 \texttt{ par } x := 2; x := x + 2$$