Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

D. Basin and P. Müller

Formal Methods and Functional Programming

Exercise Sheet 9: Big Step Semantics

Submission deadline: May 10th, 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.

Assignment 1

Let s be the following statement:

```
y := 1;
while x>0 do
  y := y * 2;
  x := x - 1
end
```

(a) What function is implemented by the **IMP** program s when the variable x initially stores a positive integer?

(b) Let σ be a state with $\sigma(\mathrm{x}) = 2$. Prove that there is a state σ' with $\sigma'(\mathrm{y}) = 4$ such that $\langle s, \sigma\rangle \to \sigma'$ using the deduction rules of the natural semantics of **IMP**.

Assignment 2

Consider the extension of **IMP** with the construct

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

where s is a statement and b a Boolean expression.

Give deduction rules for the natural semantics that capture the semantics of this loop construct.

Assignment 3

In the lecture, you have seen the proof of the direction from left to right of the following claim:

$\langle \texttt{while } b \texttt{ do } s \texttt{ end}, \sigma\rangle \rightarrow \sigma' \Leftrightarrow \langle \texttt{if } b \texttt{ then } s; \texttt{while } b \texttt{ do } s \texttt{ end else skip end}, \sigma\rangle \rightarrow \sigma'$

Prove the direction from right to left of the claim.

Assignment 4

In this assignment you will write a simple interpreter for **IMP** programs. You will use the programming language Haskell. A skeleton of the **IMP** interpreter as a literate Haskell file is available at the course web page. The skeleton file contains the data types for arithmetic expressions, Boolean expressions, and statements for representing **IMP** programs in Haskell. Moreover, the skeleton file contains some auxiliary functions. We strongly encourage you to use this skeleton file instead of what you have developed in assignment 5 of exercise sheet 8, since the skeleton contains already a parser and several auxiliary functions (e.g., evaluation of arithmetic and Boolean expressions).

Download the skeleton file and complete the definition of the function

```
transNS :: Config -> Config
```

The place where you should insert your code in the skeleton file is marked by the word `TODO`. The function `transNS` should encode the rules of the transition relation from the lecture for the natural semantics. Feel free to extend **IMP**, e.g., with local variables.

Please mail your solution of this assignment to your tutor. The email addresses of the tutors are:

Alex Summers	`alexander.summers@inf.ethz.ch`
Yannis Kassios	`ioannis.kassios@inf.ethz.ch`
Malte Schwerhoff	`scmalte@student.ethz.ch`

Assignment 5 - Headache of the week

Extend the natural semantics of **IMP** to support pointers and pointer arithmetic.

We want to introduce two new expressions:

- `*e` returns the value contained by the reference pointed by the expression e
- `&x` returns the reference pointed by variable `x`

In this context, the semantics of the original assignment statement `x:=e` is to set the reference pointed by `x` to the value of expression e. We introduce a new statement $*\mathtt{e}_1 := \mathtt{e}_2$ whose semantics is to assign the value of expression $\mathtt{e}_2$ to the location pointed by $\mathtt{e}_1$.

Hint: you have to modify the structure of the state on which the natural semantics is defined.