Programming Languages

Spring 2013

January 14, 2013

This exam is open books and open notes.

Problem 1 [40 Pts]

Consider the following syntax description of a simple programming language with

```
<program>   ::= main <block>
<block>     ::= { <decl> <statement> }
<decl>      ::= epsilon
             |   var <identifier> ; <declaration>
             |   proc <identifier> ( <identifier> default <number>)
                                     <block> ; <decl>
<statement>  ::= <statement> ; <statement>
             |   <identifier> = <expression>
             |   call <identifier> ( <expression> )
             |   call <identifier>
             |  while <expression> do <statement> endwhile
             |  <block>
```

Let us ignore the expressions (those are the standard expressions dealing with natural numbers studied in class). The novelty of the language is the ability to declare procedures, and each procedure has one formal parameter. Note that the actual parameter is optional and a default value is indicated if the actual parameter is missing.

Provide the denotational semantics - start by defining the necessary semantic algebras and then provide the valuation functions. The language needs to make use of **dynamic scoping** in handling non-local references to identifiers.

Problem 2 [20 Pts]

Answer the following questions:

- Describe precisely the data structures necessary to support the implementation of the programming language discussed in Problem 1—focusing in particular on the support of procedures and non-local references.

  Provide a **pseudo-code** describing how to resolve a non-local reference.

- Discuss whether your solution in Problem 1 does or does not allow the procedures being defined to be recursive. Motivate your answer.

Problem 3 [40 Pts]

Consider the following program:

```
{ p }
re = 0;
q = 1;
{ s }
while (q != x)
     q = q*2;
     re = re+1;
 endwhile
 { r }
```

Answer the following questions:

1. Provide the precondition p and postcondition r with the following restrictions:

   - p must have the form $x > 1 \wedge x = \ldots$ (you will need to determine "x=..." part, making it as less restrictive as possible)
   - r must have the form $re = \ldots$

2. Provide the loop invariant s according to the chosen precondition and postcondition

3. Develop the first 3 levels of Hoare's partial correctness proof starting from

$$\frac{\dots}{\{s\} \quad while\ (q \neq x)\ q = q * 2;\ re = re + 1\ \ endwhile \quad \{r\}}$$

4. Provide a Floydian expression and present the well-founded and termination conditions for this program (argue why they are correct).