Interpretação e Compilação de Linguagens– 2016-2017
Interpretation and Compilation of Programming Languages

MidTerm Test October, 21, 2016

Author: João Costa Seco
Notes: The test is open book. Students can use any (individual) printed material that each student brings along. The test has a duration of 1h30.

Q-1 **[7 val.]** This question is about the definition of an abstract syntax and the operational semantics for a programming language. Consider the programming language, called `microIt`, presented in class with the concrete syntax given by the following grammar:

$$
\begin{array}{rl}
E ::= & num \mid bool \mid E_1{+}E_2 \\
\mid & x \mid \mathsf{decl}\ x = E_1\ \mathsf{in}\ E_2 \\
\mid & \mathsf{var}(E) \mid\ !E \mid E{:=}E \\
\mid & E_1?E_2{:}E_3 \\
\mid & \mathsf{for}\ x\ \mathsf{in}\ E_1..E_2\ \mathsf{do}\ E_3
\end{array}
$$

The language comprises the base constructs for: **integer literals** (*num*), **boolean literals** (*bool*), and their usual operations, represented here by operation $E{+}E$; **identifier** use (x) and **declaration** $\mathsf{decl}\ x = E_1\ \mathsf{in}\ E_2$. The language includes the expression for creation of a variable $\mathsf{var}(E)$, the dereferencing of a variable $!E$, and the assignment expression $E{:=}E$. The semantics of the presented language follows the semantics presented in the course lectures. The language also includes a conditional expression that yields the result of E_2 if E_1 denotes true, and of E_3 if E_1 denotes false; and an iterator operation ($\mathsf{for}\ x\ \mathsf{in}\ E_1..E_2\ \mathsf{do}\ E_3$) over a range given by two integer values (denoted by E_2 and E_3), evaluating expression E_3 for all possible values of identifier x, that range from the integer value denoted by expression E_1 to the integer value of expression E_2 (inclusive).
Consider the example written in the programming language `microIt`:

```
decl s = var(0) in
decl x = for x in 1..10 do (x%2==0)?(s:=!s+1):0 in !s
```

Notice that operators `!=`, `==`, and `%` are generally represented in the language by the operator `+`.

a) **[1 val.]** **Define** the abstract syntax of the **iteration operation** in language `microIt` by means of an abstract data type, defined by set of (abbreviated) Java classes and interfaces.

b) **[2 val.]** **Define** the set of values of language `microIt` by means of an abstract data type, defined by a set of (abbreviated) Java classes and interfaces.

c) **[2 val.]** **Define** the operational semantics of language `microIt` by means of a method **eval** for the **iteration expression**.

d) **[1 val.]** **State** the denotation (value) of the example above according to the semantics defined in the previous question, and the expected semantics for the remaining operators.

e) **[1 val.]** **Explain** why the iteration expression above only makes sense in the context of an imperative language. (max 50 words.)

Q-2 **[7 val.]** This question is about the definition of the type system for language `microIt`. To answer the following questions you may use abstract data types, defined by a set of Java classes and interfaces, and the corresponding methods using Java Code. You may also use the notation used in the lecture slides.

a) **[2 val.]** **Define** the set of types used to type programs of language `microIt`.

b) **[2 val.]** **Define** the type system of language `microIt` by means of a **typecheck** function, for the **iteration expression**.

c) **[1 val.]** **State** the type denotation of the example expression in question **Q-1**, according to the type semantics defined in question **Q-2a**.

d) **[1 val.]** **Enumerate** the execution errors that may occur during the execution of a program written in language `microIt`, according to the semantics defined in question **Q-1**.

e) **[1 val.]** **Indicate and justify** which execution errors may be prevented by the type system, and those that cannot.

Q-3 **[6 val.]** This question is about the evaluation and compilation of programs using mutable environments. Consider the following program written in the `microIt` language.

```
decl
  x = 1
  y = 10
in
  decl
    x = x + y
    z = y * 10
    w = 20
  in
    decl
      v = w + y        (*)
    in
      x + y + v + z    (*)
```

a) **[1 val.]** **Indicate** what are the **evaluation environments** for the subexpressions `w + y`, and `x + y + v + z`, highlighted in the listing above with `(*)`.

b) **[1 val.]** **Indicate** what are the **compilation environments** for the subexpressions `w + y`, and `x + y + v + z`, highlighted in the listing above with `(*)`.

c) **[4 val.]** **List** the set of instructions that results from translating expression `x + y + v + z` to the Jasmin assembly language.

Clearly State the compilation addresses (result of function `find`) of all 4 identifiers used.