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

MidTerm Test November, 2, 2017

Author: João Costa Seco
Notes: The test is closed book with the exception of a single handwritten (cheat) sheet. The test has a duration of 1h30.

Q-1 [6 val.] This question is about the definition of an abstract syntax and the operational semantics for a programming language. Consider the programming language with a three-valued logic, called `bool3`, presented in class with the concrete syntax given by the following grammar:

$$\begin{aligned} E ::=\ & \mathit{num} \mid E_1{+}E_2 \mid \mathsf{true} \mid \mathsf{false} \mid \mathsf{unknown} \\ \mid\ & E_1 \&\&\& E_2 \mid E_1 \| E_2 \mid \mathsf{not3}\ E_1 \mid \mathtt{die3} \mid E_1\langle\langle E_2/E_3/E_4\rangle\rangle \\ \mid\ & x \mid \mathsf{decl}\ x = E_1\ \mathsf{in}\ E_2 \end{aligned}$$

The language comprises the base constructs for: **integer literals** (*num*), and their corresponding operations, represented here by operation $E + E$; **three-valued boolean literals** (true, false, and unknown), and their corresponding operations, &&&, |||, and not3 with the following rules:

not3 true $\triangleq$ false	true &&& $a \triangleq a$	true ||| $a \triangleq$ true
not3 false $\triangleq$ true	false &&& $a \triangleq$ false	false ||| $a \triangleq a$
not3 unknown $\triangleq$ unknown	a &&& $b \triangleq b$ &&& a	a ||| $b \triangleq b$ ||| a

The language also includes a three-valued random expression `die3` that yields a random three-valued value, a conditional expression $E_1\langle\langle E_2/E_3/E_4\rangle\rangle$ that evaluates the condition E_1, and then its result is given by E_2 if the condition value is true, E_3 if the value is false, and E_4 if it is unknown.

Additionally, consider expressions for **identifier** use (x) and **declaration** decl $x = E_1$ in E_2. The semantics of the presented language follows the semantics presented in the course lectures. Consider the example written in the programming language `bool3`:

```
decl
  b1 = die3
  b2 = die3
  b3 = die3
in
  b1< 1 / b2< b3< 2 / 3 / 4> / 5 / 6> / 7>
```

Note: in the example the conditional expression is given by `E<E1/E2/E3>`.

a) **[1 val.]** **Define** the abstract syntax of the expressions `die3` and **conditional expression** in language `bool3` by means of abstract data type cases, using a set of (abbreviated) Java classes and interfaces.

b) **[1 val.]** **Define** the set of values of language `bool3` by means of an abstract data type, using a set of (abbreviated) Java classes and interfaces.

c) **[3 val.]** **Define** the operational semantics of language `bool3`, for the cases of **logic expressions** (it includes the conditional expression), by means of a method **eval**. Hint: The semantics of three-valued logic can be defined with integers, and operations **min** and **max**.

d) **[1 val.]** **State** the denotation (value) of the example above according to the semantics defined in the previous question. Use the expected semantics presented in the lectures for the remaining operators. Consider that `die3` yields the sequence false, unknown, and true.

Q-2 **[8 val.]** This question is about the definition of a type system for language `bool3`. 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.

a) **[1 val.]** **Define** the set of types used to type programs of language `bool3`.

b) **[3 val.]** **Define** the type system of language `bool3` for the case of **logic expressions** (it includes the conditional expression), by means of a **typecheck** method in the AST classes.

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.]** Consider the expression `(b<1/unknown/x>)<x/2/4>`. **Present**, if possible, a typing denotation and a typing environment that makes this expression well-typed.

e) **[1 val.]** **Enumerate** the execution errors that may occur during the execution of a program written in language `bool3`, according to the semantics defined in question **Q-1**.

f) **[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 compilation of programs using mutable environments. Consider the following program written in the `bool3` language and the compilation schema introduced in the course lectures.

```
decl
  x = 1
  y = 10
in
  decl
    x = x == y
    z = y * 10
    w = die3
  in
    x< w < z / y / 0 > / 0 / y >
```

a) **[2 val.]** **Indicate** what is the **compilation environment** for the subexpression `w < z / y / 0 >` above.

b) **[4 val.]** **List** the set of instructions that results from translating expression `w < z / y / 0 >` to the Jasmin assembly language.