**Assignment 3**

Problem set questions:

**21. Using the virtual machine instructions given in Section 3.5.1.1, give an operational semantic definition of the following:**

**a)** Java **do**-**while**

Syntax:

do

{

Statement;

}

While (expression);

Operational Semantic:

Loop:

Statement;

**If** expression == **false** **goto** out

**goto** Loop

Out…

**c)** C ++ **If then else**

Syntax:

If(Boolean\_expression)

{

Statements\_1;

}

Else

{

Statements\_2;

}

Operational Semantic definition:

**If**(Boolean\_expression == True) **goto** L1

**goto** L2

L1: Statements\_1;

L2: Statements\_2;

**22. Write a denotational semantics mapping function for the following statements:**

b) Java **do-while**

Syntax:

do

{

Statement;

}

While (expression);

Denotational Semantic:

Mdw (**execute** Ln **till** bool, s) Δ=

If Mst (Ln, s) == **error**

then **error**

else if Mbool (Bool, s) == **undef**

then **error**

else if Mbool (Bool, s) == true

then Mdw (**execute** Ln **till** Bool, Mst (Ln, s))

else

S

In here “Mdw”represents the java do-while loop, “Bool” is the condition to be satisfied, and “Ln” is the statement list to be executed. Program state is “s”, “Δ=” defines that the mathematical function is being used. Mapping function “Mst (Ln, s)” maps statement list and program states to new states.

If error in the statement list, then error is generated, if the condition specified by Boolean expression is not defined, then error is generated, if it is true then the loop is executed and the state of the program changed to new state defined by “Mst (Ln, s)”

c) Java Boolean expressions

Syntax:

<bool\_exp> 🡪 <L\_exp> <op> <R\_exp>

<L\_exp> 🡪 <int> | <var>

<R\_exp> 🡪 <int> | <var>

<op> 🡪 > | =

Denotational semantics:

MBE (<bool\_exp>, s) Δ=

<L\_exp><op><R\_exp> =>

If MBE (<bool\_exp>.<L\_exp>, s) == **undef** OR

If MBE (<bool\_exp>.<R\_exp>, s) == **undef**

Then **error**

else if (<bool\_exp>.<op> == ’>’)

then MBE (<bool\_exp>.<L\_exp>, s) **>** MBE (<bool\_exp>.<R\_exp>, s)

else MBE (<bool\_exp>.<L\_exp>, s) **==** MBE (<bool\_exp>.<R\_exp>, s)

“MBE” represents mapping function for java Boolean expression, “<bool\_exp>” is the expression to be executed. Current program state is denoted by “s”, “Δ=” defines that the mathematical function is being used. The <L\_exp> and <R\_exp> specify left and right operands respectively.

If the left or right operand of a Boolean expression is undefined, then it generates error.

Although there may be any Boolean expression, here I am only considering “>” and “==” for the example.