Practice Test 3/4

Type Theory and Operational Semantics

Question 1

For this question you need the typing rules of Appendix A.

a) Write down the steps taken by a typechecker on the following program fragment:

```
int x;
{
    x = 2;
    int x = 3;
}
y = 2*x;
```

b) Under what condition will the typechecker confirm that the code fragment above is valid?

c) What will be the values of x and y after executing the code fragment?

Question 2

For this question you will need the operational semantics rules in Appendix B.

a) Write down, in the form of a proof tree, the steps taken by an interpreter on the code

```
x+++++x*x
```

in an environment for which `x:=1.`

b) Assume that the value of x is 2. Write down the value of x after executing

```
x=x++;
```

and after executing, instead,

```
x=++x;
```

Appendix A: Typing Rules

Below I put pictures from the book for most of the rules and add two rules that are not in the book.

The function lookupVar refers to the function of the typechecker that looks up a variable in a context and returns the type of the variable.

Notice that the book uses ==> where I write the more commonly used |- instead.

Variables :

```
— — — — — — — — — — — —    if t == lookupVar x Gamma
     Gamma |- x : t
```

Assignment:

```
        Gamma |- e : t
— — — —-— — — — — — — — — —    if t == lookupVar x Gamma
       Gamma |- x = e : t
```

Integers (doubles are similar):

```
— — — — — — — — — — — — — — — — — —-
          Gamma |- 3 : int
```

Binary operations (*, -, etc are similar):

$$\frac{\Gamma \Longrightarrow a : t \quad \Gamma \Longrightarrow b : t}{\Gamma \Longrightarrow a + b : t} \quad \text{if } t \text{ is } \texttt{int} \text{ or } \texttt{double}$$

Blocks:

$$\frac{\Gamma . \Longrightarrow r_1 \ldots r_m \text{ valid} \quad \Gamma \Longrightarrow s_2 \ldots s_n \text{ valid}}{\Gamma \Longrightarrow \{r_1 \ldots r_m\} s_2 \ldots s_n \text{ valid}}$$

Sequences of statements:

Gamma |- s1 valid   Gamma |- s2 … sn valid
——————————————————————— if s1 is not a declaration
Gamma |- s1 … sn

Declarations:

$$\frac{\Gamma(x : T) \Longrightarrow s_2 \ldots s_n \text{ valid}}{\Gamma \Longrightarrow Tx; s_2 \ldots s_n \text{ valid}}$$

Expressions:

$$\frac{\Gamma \Longrightarrow e : t}{\Gamma \Longrightarrow e;\ \text{valid}}$$

Appendix B: Operational Semantics

Assignment:

$$\frac{\gamma \vdash e \Downarrow \langle v, \gamma' \rangle}{\gamma \vdash x \texttt{=} e \Downarrow \langle v, \gamma'(x := v) \rangle}$$

Binary operations (replace - by +,*. etc):

$$\frac{\gamma \vdash a \Downarrow \langle u, \gamma' \rangle \quad \gamma' \vdash b \Downarrow \langle v, \gamma'' \rangle}{\gamma \vdash a \texttt{-} b \Downarrow \langle u - v, \gamma'' \rangle}$$

Postincrement (postdecrement works similarly):

$$\frac{}{\gamma \vdash x\texttt{++} \Downarrow \langle v, \gamma(x := v + 1) \rangle} \quad \text{if } x := v \text{ in } \gamma$$

Preincrement (predecrement works similarly):

$$\frac{}{\gamma \vdash \texttt{++}x \Downarrow \langle v + 1, \gamma(x := v + 1) \rangle} \quad \text{if } x := v \text{ in } \gamma$$