CMPU 331 · Compilers · Fall 2019

Assignment 3

Due Tuesday, November 12, 2019

This assignment covers semantic analysis, including symbol tables and type checking. You may discuss this assignment with other students and work on the problems together. However, your write-up should be your own individual work, and you should indicate in your submission who you worked with, if applicable. Assignments should be submitted through Moodle as a PDF by 11:59pm. (You can export a PDF from Google Docs, Word, LibreOffice, LaTeX, or any other text editor you prefer.)

Problem 1

Consider the following code example in DL, from page 248 of the textbook "A Practical Approach to Compiler Construction" by Des Watson:

```
factorial(n);
{
    if (n == 0) { return 1 }
    else { return n * factorial(n - 1) }
}

int x;
{
    x = 1;
    while (x <= 10) {
        print(factorial(x));
        x = x + 1
    }
}
```

(a) What are the symbols in this code example that need entries in the symbol table?

(b) What are the types of the symbols in the symbol table?

(c) Which symbols are local and which are global?

Problem 2

Consider the following type rules for DL, where:

- $x : T$ means x has type T
- $y : T[i]$ means y has an array type, with array size i, containing elements of type T
- $S[T/x]$ sets the type T for the variable x in the type environment
- $S(x) = T$ looks up the type T of the variable x in the type environment
- $S \vdash$ means "it is provable, given the type environment S, that..."
- and rules have the form:

$$\frac{\vdash Hypothesis \quad \vdash Hypothesis}{\vdash Conclusion}$$

Constant:

$$\frac{\vdash i \text{ is an integer constant}}{\vdash i : int}$$

Variable Declaration:

$$\frac{S[int/x] \quad S \vdash E_1 : T_1}{S \vdash x : int \text{ in } E_1 : T_1}$$

Variable Reference:

$$\frac{S(x) = int}{S \vdash x : int}$$

Array Declaration:

$$\frac{S[int[i]/y] \quad \vdash i : int \quad S \vdash E_1 : T_1}{S \vdash y : int[i] \text{ in } E_1 : T_1}$$

Array Reference:

$$\frac{S(y) = int[i]}{S \vdash y : int[i]}$$

Array Element Reference:

$$\frac{S(y) = int[i] \quad S \vdash E_1 : int}{S \vdash y[E_1] : int}$$

Addition:

$$\frac{S \vdash E_1 : int \quad S \vdash E_2 : int}{S \vdash E_1 + E_2 : int}$$

Subtraction:

$$\frac{S \vdash E_1 : int \quad S \vdash E_2 : int}{S \vdash E_1 - E_2 : int}$$

Multiplication:

$$\frac{S \vdash E_1 : int \quad S \vdash E_2 : int}{S \vdash E_1 * E_2 : int}$$

Division:

$$\frac{S \vdash E_1 : int \;\; S \vdash E_2 : int}{S \vdash E_1 / E_2 : int}$$

Assign to a variable:

$$\frac{S(x) = int \;\; S \vdash E_1 : int}{S \vdash x = E_1 : int}$$

Assign to an array element:

$$\frac{S(y) = int[i] \;\; S \vdash E_1 : int \;\; S \vdash E_2 : int}{S \vdash y[E_1] = E_2 : int}$$

Note: this is not a complete set of type rules for DL.

For each of the following small DL programs, would the statement on the line marked with the comment `typecheck?` pass a type check using the rules above? If it would pass a type check, explain which type rules were used in the check. If it would not pass a type check, explain which type rules failed.

(a)

```
int x;
{
    x = 2 + 3 \* typecheck? *\
}
```

(b)

```
{
    x = 7 * 4 \* typecheck? *\
}
```

(c)

```
int y[10];
{
    y[5] = 9 / 3 \* typecheck? *\
}
```

(d)

```
int a[5];
{
    a = 15 - 3 \* typecheck? *\
}
```

(e)

```
int a, b, c[10];
{
    a = 3
    b = 5
    c[a] = b + 6 \* typecheck? *\
}
```