CS301P Compiler Design Laboratory Lab#11

Date: November 07, 2023

Objective: To design and implement semantic analyzer that deals with variable declarations in a block structured language.

Problem Statement

As learned in class, semantic analysis/translation can be peformed while parsing itself by implementing the rules/actions with the parser. The task for this lab is the following. Consider type declarations in a block structured language like C, in which a program may be written in terms blocks where each block may have declaration section followed by the statements section. Consider variable declarations consisting of premitive data types **int**, **float**, **char** and composite data type, **array** (including multidimensional arrays). Assume sizes of char, float, and int data types are 1, 4, and 4 bytes respectively.

Design a grammar to generate the all possible type declarations consisting of data types mentioned above, and implement the necessary functionality

- to add the type information for each identifier defined in the program,
- to find the size requirements for each declaration section,
- to maintain the relative memory address (i.e., offset) for each variable declared. Assume 16-bit memory addresses.

Input : Blocks of C variable declarations Output: Display symbol table entries Execusion: $./parser prog.c

1. Testcase:

   Input:

   ```
   {
     int a, b, c;
     char e;
     float pi = 3.24;
   }
   ```

   Output:

   ```
   0x0000 a int
   0x0004 b int
   0x0008 c int
   0x0009 e char
   0x000A pi float
   ```

2. Testcase:

Input:

```
{
        int a;
        int x, y;
        float c;
        {
                int a;
                int b;
        }
        {
                 char m;
                 int n[10];
        }
}
```

Output:

```
0x0000 a int
0x0004 x int
0x0008 y int
0x000C c float

0x0000 a int
0x0004 b int

0x0000 m char
0x0001 n intarray 40
```

3. Testcase

Input:

```
{
         int a;
         int b;
         int a;
}
```

Output:

```
error: redeclaration of 'a'
```

4. Testcase

Input:

```
{
         int a;
         char a;
         {
                 int c;
                 int c;
                 z = a + c;
         }
}
```

Output:

```
error: conflicting types for 'a'
error: redeclaration of 'c'
error: var 'z' is not declared in the scope
```

5. Testcase

Input:

```
{
        int a;
        int b;
        int z;
        z = a + b++;
}
```

Output:

```
t0 = a + b
t1 = b + 1
b = t1
z = t0
```

Submission Guidelines

- The name of the parser executable should be *parser*
- The respective lex and yacc programs can have the same name but with the extension *.l* and *.y*, respectively.
- The names for the given program should be *prob1* of course with appropriate extensions.
- Submit also the 4 test cases that you have tried. The files should be named *test1.c*, ... *test4.c*
- Other submission requirements remain same as week#1.

Evaluation Guidelines

Same as week#1

Academic Honesty

Same as week#1