Lab One

Emily Doran

Emily.Doran1@Marist.edu

February 23, 2021

Crafting A Compiler

1.11 (MOSS)

Investigate the techniques MOSS uses to find similarity. How does MOSS differ from other approaches for detecting possible plagiarism?

The Measure of Software Similarity, better known as MOSS, is a system that is used to help detect plagiarism by detecting the similarities between code. Aside from being able to find similarities between code written in different languages, what makes MOSS unique is that it also compares the tokens of the program. MOSS returns the number of tokens matched between programs, as well as the number of lines matched.

3.1 (Token Sequence)

Assume the following text is presented to a C scanner:

```
main(){
    const float payment = 384.00;
    float bal;
    int month = 0;
    bal=15000;
    while (bal>0){
        printf("Month: %2d Balance: %10.2f\n", month, bal);
        bal=bal-payment+0.015*bal;
        month=month+1;
    }
}
```

What token sequence is produced? For which tokens must extra information be returned in addition to the token code?

```
FUNCTION [ main ] found at (1:1)
L_PAREN [ ( ] found at (1:5)
R_PAREN [ ( ] found at (1:6)
L_BRACE [ { ] found at (1:7)
```

```
CONST [ const ] found at (2:5)
TYPE [ float ] found at (2:11)
ID [ payment ] found at (2:17)
ASSIGN_OP [ = ] found at (2:25)
DIGIT [ 384.00 ] found at (2:27)
PUNCT [ ; ] found at (2:33)
TYPE [ float ] found at (3:5)
ID [ bal ] found at (3:11)
PUNCT [ ; ] found at (3:14)
TYPE [ int ] found at (4:5)
ID [ month ] found at (4:9)
ASSIGN_OP [ = ] found at (4:15)
DIGIT [ 0 ] found at (4:17)
PUNCT [ ; ] found at (4:18)
ID [ bal ] found at (5:5)
ASSIGN_OP [ = ] found at (5:8)
DIGIT [ 15000 ] found at (5:9)
PUNCT [ ; ] found at (5:14)
WHILE [ while ] found at (6:5)
L_PAREN [ ( ] found at (6:11)
ID [ bal ] found at (6:12)
GREATER_THAN [ > ] found at (6:15)
DIGIT [ 0 ] found at (6:16)
R_PAREN [ ( ] found at (6:17)
L_BRACE [ { ] found at (6:18)
PRINTF [ printf ] found at (7:9)
L_PAREN [ ( ] found at (7:10)
STRING [ "Month: %2d Balance: %10.2f\n" ] found at (7:10)
PUNCT [ , ] found at (7:40)
ID [ month ] found at (7:42)
PUNCT [ , ] found at (7:47)
ID [ bal ] found at (7:49)
R_PAREN [ ) ] found at (7:52)
PUNCT [ ; ] found at (7:53)
ID [ bal ] found at (8:9)
ASSIGN_OP [ = ] found at (8:12)
ID [ bal ] found at (8:13)
SUBTRACT_OP [ - ] found at (8:16)
ID [ payment ] found at (8:17)
ADDITION_OP [ + ] found at (8:24)
DIGIT [ 0.015 ] found at (8:25)
MULTIPLY_OP [ * ] found at (8:30)
ID [ bal ] found at (8:31)
PUNCT [ ; ] found at (8:34)
ID [ month ] found at (9:9)
ASSIGN_OP [ = ] found at (9:14)
ID [ month ] found at (9:15)
ADDITION_OP [ + ] found at (9:20)
DIGIT [ 1 ] found at (9:21)
PUNCT [ ; ] found at (9:22)
R_BRACE [ } ] found at (10:5)
R_BRACE [ } ] found at (11:1)
```

For identifier tokens, we need to store some additional information because the scanner doesn't know when an identifier should be entered into the symbol table or if it should return a pointer.

Dragon

1.1.4 (Advantages of C as a Target Language)

A compiler that translates a high-level language into another high-level language is called a *source-to-source* translator. What advantages are there to using C as a target language for a compiler?

C is a good target language for a compiler because it is low level, so it isn't necessary for the compiler to translate to machine code. Also, since C is such a widely used language, the compilers can be used on many different platforms.

1.6.1 (Variables in Block-Structured Code)

For the block-structured C code of Fig. 1.13(a) below, indicate the values assigned to *w, x, y,* and *z.*

```
1  int w, x, y, z;
2  int i = 4; int j = 5;
3  {   int j = 7;
4      i = 6;
5      w = i + j;
6  }
7  x = i + j;
8  {   int i = 8;
9      y = i + j;
10 }
11 z = i + j;
```

w = 13

x = 11

y = 13

z = 11