Lab One

Devin White
Devin.White@Marist.edu

February 15, 2022

1 Crafting a Compiler

1.1 Exercise 1.11 (MOSS)

The Measure Of Software Similarity (MOSS) [SWA03] tool can detect similarity of programs written in a variety of modern programming languages. Its main application has been in detecting similarity of programs submitted in computer science classes, where such similarity may indicate plagiarism (students, beware!). In theory, detecting equivalence of two programs is undecidable, but MOSS does a very good job of finding similarity in spite of that limitation. Investigate the techniques MOSS uses to find similarity. How does MOSS differ from other approaches for detecting possible plagiarism?

Upon investigating MOSS it is a very effective application to detect similarities. MOSS uses a very specific set of rules to find similarities. First off it ignores white space while trying to find similarities so changing the way a program looks will still flag a similarity from MOSS. MOSS will also not look at variable names as they can be easily changed. On top of this, it will only be flagging portions that are long and significant enough. A simple while loop will likely not be flagged as it is very commonly used in programs. MOSS will also show the source of where the suspected similar code originated.

1.2 Exercise 3.1 (token sequence)

Assume the following text is presented to a C scanner:

$$\begin{aligned}
&main() \\
&\qquad constfloatpayment = 384.00; \\
&\qquad floatbal; \\
&\qquad intmonth = 0; \\
&\qquad bal = 15000; \\
&\qquad while(bal > 0) \\
&\qquad printf("Month : 2dBalance : 10.2f", month, bal); \\
&\qquad bal = bal - payment + 0.015 * bal; \\
&\qquad month = month + 1;
\end{aligned} \tag{1.1}$$

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

Lexer - PROGRAM$_{S}TART[main]foundat(1:1)$

lexer – LEFT$_{P}ARENTHESES[\ (\ ]foundonline$ $(1:5)$

lexer – RIGHT$_{P}ARENTHESES[\ )\ ]foundonline$ $(1:6)$

Lexer – LEFT$_{B}RACE[\quad]foundonline$ $(1:7)$

Lexer — CONST [const] found on line (2:1)

Lexer — FLOAT$_{T}YPE[float]foundonline(2,7)$

Lexer — ID [payment] found on line (2,13)

Lexer – ASSIGNMENT$_{O}PERATOR[\ =\ ]foundonline$ $(2:21)$

Lexer – FLOAT$_{D}IGIT[$ 384.00 $]foundonline$ $(2:23)$

Lexer — LINE$_{E}ND[;]foundonline(2,29)$

Lexer — FLOAT$_{T}YPE[float]foundonline(2,7)$

Lexer – ID [bal] found on line (3:7)

Lexer — LINE$_{E}ND[;]foundonline(3,10)$

Lexer— INT [int] found on line (4:1)

Lexer — ID [month] found on line (4:5)

Lexer – ASSIGNMENT$_{O}PERATOR[\ =\ ]foundonline$ $(4:11)$

Lexer – DIGIT [0] found on line (4:13)

Lexer – LINE$_{E}ND[;]foundonline$ $(4:14)$

Lexer— ID [bal] found on line (5:1)

Lexer – ASSIGNMENT$_{O}PERATOR[\ =\ ]foundonline$ $(5:4)$

Lexer – DIGIT [15000] found on line (5:5)

Lexer – LINE$_E ND[;] found on line$ (5 : 10)

Lexer – WHILE [while] found on line (6:1)

Lexer – LEFT$_P ARENTHESES[\ (\ ] found on line$ (6 : 7)

Lexer – ID [bal] found on line (6:8)

Lexer – GREATER$_T HAN[>] found on line$ (6 : 11)

Lexer – DIGIT [0] found on line (6:12)

Lexer – RIGHT$_P ARENTHESES[\ )\ ] found on line$ (6 : 13)

Lexer – LEFT$_B RACE[\ \ ] found on line$ (6 : 14)

Lexer – PRINT [printf] found on line (7:1)

Lexer – LEFT$_P ARENTHESES[\ (\ ] found on line$ (7 : 7)

Lexer – QUOTE ["] found on line (7:8)

Lexer – STRING [Month:

Lexer – QUOTE ["] found on line (7:37)

Lexer – COMMA [,] found on line (7:38)

Lexer – ID [month] found on line (7:40)

Lexer – COMMA [,] found on line (7:45)

Lexer – ID [bal] found on line (7:47)

Lexer – RIGHT$_P ARENTHESES[\ )\ ] found on line$ (7 : 50)

Lexer – LINE$_E ND[;] found on line$ (7 : 51)

Lexer – ID [bal] found on line (8:1)

Lexer – ASSIGNMENT$_O PERATOR[=] found on line$ (8 : 4)

Lexer – ID [bal] found on line (8:5)

Lexer — MINUS$_O PERATOR found on line$ (8 : 8)

Lexer – ID [payment] found on line (8:9)

Lexer – ADDITION$_O PERATOR[+] found on line$ (8 : 16)

Lexer – DIGIT [0.015] found on line (8:17)

Lexer – MULTIPLY$_O PERATOR[+] found on line$ (8 : 26)

Lexer – ID [bal] found on line (8:23)

Lexer – LINE$_E ND[;] found on line$ (8 : 26)

Lexer— ID [month] found on line (9:1)

Lexer – ASSIGNMENT$_O PERATOR[=] found on line$ (9 : 6)

Lexer— ID [month] found on line (9:7)

Lexer – ADDITION$_O PERATOR[+] found on line$ (9 : 12)

Lexer – DIGIT [1] found on line (9:13)

Lexer – RIGHT$_{B}RACE[\ \ ]foundonline\ (10:1)$

Lexer – RIGHT$_{B}RACE[\ \ ]foundonline\ (11:1)$

Lexer – EOP [ ] found on line (11:2) (1.2)

2 Dragon

2.1 Exercise 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?

(2.1)

One of the biggest advantages to using C as a target language is that it compiles down to something readable by most all devices.

2.2 Exercise 1.6.1 (variables in blockstructured code)

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

(2.2)

The values for each w, x, y, and z are 13, 11, 13, and 11 respectively.