Lab One

Shannon Cordoni

Shannon.Cordoni@Marist.edu

February 13, 2022

1 PROBLEM ONE: CRAFTING A COMPILER 1.11

MOSS is an interesting new technology used to find similarities in computer programs. It compares the similarities between to files and returns a percentage representing the amount of similarity between the two files.

2 PROBLEM TWO: CRAFTING A COMPILER 3.1

The token sequence that would be produced is:

```
DEBUG Lexer - MAIN [ main ]
DEBUG Lexer - CONST [ const ]
DEBUG Lexer - FLOAT [ float ]
DEBUG Lexer - ID [ payment ]
DEBUG Lexer - ASSIGNMENT [ = ]
DEBUG Lexer - DIGIT [ 3 ]
DEBUG Lexer - DIGIT [ 8 ]
DEBUG Lexer - DIGIT [ 4 ]
DEBUG Lexer - DECIMAL [ . ]
DEBUG Lexer - DIGIT [ 0 ]
DEBUG Lexer - DIGIT [ 0 ]
DEBUG Lexer - SEMICOLON [ ; ]
DEBUG Lexer - FLOAT [ float ]
DEBUG Lexer - ID [ bal ]
DEBUG Lexer - SEMICOLON [ ; ]
DEBUG Lexer - INT [ int ]
DEBUG Lexer - ID [ month ]
DEBUG Lexer - ASSIGNMENT [ = ]
DEBUG Lexer - DIGIT [ 0 ]
DEBUG Lexer - SEMICOLON [ ; ]
DEBUG Lexer - ID [ bal ]
DEBUG Lexer - ASSIGNMENT [ = ]
DEBUG Lexer - DIGIT [ 1 ]
DEBUG Lexer - DIGIT [ 5 ]
DEBUG Lexer - DIGIT [ 0 ]
DEBUG Lexer - DIGIT [ 0 ]
```

```
28  DEBUG Lexer - DIGIT [ 0 ]
29  DEBUG Lexer - SEMICOLON [ ; ]
30  DEBUG Lexer - WHILE [ while ]
31  DEBUG Lexer - LPAREN [ ( ]
32  DEBUG Lexer - ID [ bal ]
33  DEBUG Lexer - GREATERTHAN [ > ]
34  DEBUG Lexer - DIGIT [ 0 ]
35  DEBUG Lexer - RPAREN [ ) ]
36  DEBUG Lexer - LBRACE [ { ]
37  DEBUG Lexer - PRINT [ print ]
38  DEBUG Lexer - LPAREN [ ( ]
39  DEBUG Lexer - OPENQUOTE [ " ]
40  DEBUG Lexer - CHAR [ M ]
41  DEBUG Lexer - CHAR [ o ]
42  DEBUG Lexer - CHAR [ n ]
43  DEBUG Lexer - CHAR [ t ]
44  DEBUG Lexer - CHAR [ h ]
45  DEBUG Lexer - SEMICOLON [ : ]
46  DEBUG Lexer - SPACE [   ]
47  DEBUG Lexer - PERCENT [ % ]
48  DEBUG Lexer - DIGIT [ 2 ]
49  DEBUG Lexer - CHAR [ d ]
50  DEBUG Lexer - SPACE [   ]
51  DEBUG Lexer - CHAR [ B ]
52  DEBUG Lexer - CHAR [ a ]
53  DEBUG Lexer - CHAR [ l ]
54  DEBUG Lexer - CHAR [ a ]
55  DEBUG Lexer - CHAR [ n ]
56  DEBUG Lexer - CHAR [ c ]
57  DEBUG Lexer - CHAR [ e ]
58  DEBUG Lexer - SEMICOLON [ : ]
59  DEBUG Lexer - SPACE [   ]
60  DEBUG Lexer - PERCENT [ % ]
61  DEBUG Lexer - DIGIT [ 1 ]
62  DEBUG Lexer - DIGIT [ 0 ]
63  DEBUG Lexer - DECIMAL [ . ]
64  DEBUG Lexer - DIGIT [ 2 ]
65  DEBUG Lexer - CHAR [ f ]
66  DEBUG Lexer - CHAR [ \ ]
67  DEBUG Lexer - CHAR [ n ]
68  DEBUG Lexer - CLOSEQUOTE [ " ]
69  DEBUG Lexer - COMMA [ , ]
70  DEBUG Lexer - ID [ month ]
71  DEBUG Lexer - COMMA [ , ]
72  DEBUG Lexer - ID [ bal ]
73  DEBUG Lexer - RPAREN [ ) ]
74  DEBUG Lexer - SEMICOLON [ ; ]
75  DEBUG Lexer - ID [ bal ]
76  DEBUG Lexer - ASSIGNMENT [ = ]
77  DEBUG Lexer - ID [ bal ]
78  DEBUG Lexer - MINUS [ - ]
79  DEBUG Lexer - ID [ payment ]
80  DEBUG Lexer - PLUS [ + ]
81  DEBUG Lexer - DIGIT [ 0 ]
82  DEBUG Lexer - DECIMAL [ . ]
83  DEBUG Lexer - DIGIT [ 0 ]
84  DEBUG Lexer - DIGIT [ 1 ]
85  DEBUG Lexer - DIGIT [ 5 ]
86  DEBUG Lexer - MULTIPLY [ * ]
87  DEBUG Lexer - ID [ bal ]
88  DEBUG Lexer - SEMICOLON [ ; ]
89  DEBUG Lexer - ID [ month ]
90  DEBUG Lexer - ASSIGNMENT [ = ]
91  DEBUG Lexer - ID [ month ]
92  DEBUG Lexer - PLUS [ + ]
```

```
93  DEBUG Lexer - DIGIT [ 1 ]
94  DEBUG Lexer - SEMICOLON [ ; ]
95  DEBUG Lexer - RBRACE [ } ]
96  DEBUG Lexer - RBRACE [ } ]
```

3 PROBLEM THREE: DRAGON 1.1.4

The advantages to using C as a target language for a compiler is that is it available on many platforms allowing for wide spread use. Along with having a simple set of keywords it makes it easy for fast computation of machine code.

4 PROBLEM THREE: DRAGON 1.6.1

The value assigned to w is $i + j$, or $6 + 7$ which is 13. The value assigned to x is also $i + j$, but this time it equivocates to $6 + 5$ which is 11. The value assigned to y is also $i + j$, or $8 + 5$, which is 13. Lastly, the value assigned to z which is also $i + j$, or $6 + 5$ which is 11.