Lab One

Brandon Litwin

Brandon.Litwin1@Marist.edu

January 31, 2019

1 Crafting A Compiler

1.11 Measure of Software Similarity (MOSS) is a program that can detect the level of similarity in two programs, similar to a plagiarism detector like TurnItIn. It compares items like number of matching tokens and number of matching lines. One technique that makes MOSS unique is robust winnowing. It is a way of filtering noise (common words that any program would use) from the code and looking at larger strings of code that match. If the string matches at least a certain length, then the match is detected. In addition, a course instructor can input certain pieces of information that would be expected in every final answer, such as a certain form of output. This information is known as boilerplate. MOSS also is able to ignore whitespace by stripping out all the whitespace before analyzing the code. MOSS is more efficient than other programs in how it goes through the winnowing process.

3.1 Token.Identifier ("main") // requires a return type such as int
Token.LParen
Token.RParen
Token.LBrace
Token.Blank
Token.Keyword ("const")
Token.Blank
Token.Keyword ("float")
Token.Blank
Token.Identifier ("payment")
Token.Blank
Token.Assign
Token.Blank
Token.Constant ("384.00")
Token.Semi
Token.Blank
Token.Keyword ("float")
Token.Blank

Token.Identifier ("bal")
Token.Blank
Token.Semi
Token.Blank
Token.Keyword ("int")
Token.Blank
Token.Identifier ("month")
Token.Blank
Token.Assign
Token.Blank
Token.Constant ("0")
Token.Blank
Token.Identifier ("bal")
Token.Assign
Token.Constant ("15000")
Token.Semi
Token.Keyword ("while")
Token.Blank
Token.LParen
Token.Identifier ("bal")
Token.GreaterThan
Token.Constant ("0")
Token.Blank
Token.Keyword ("printf")
Token.LParen
Token.String ("Month: ")
Token.Mod
Token.Constant ("2")
// more info required here because of the context of these tokens Token.Identifier ("d")
Token.String ("Balance: ")
Token.Mod
Token.Decimal ("10.2")
// more info required here because of the context of these tokens Token.Identifier ("f")
Token.Newline
Token.String ('"')
Token.Comma
Token.Blank
Token.Identifier ("month")
Token.Comma
Token.Blank
Token.Identifier ("bal")
Token.RParen
Token.Semi
Token.Blank
Token.Identifier ("bal")
Token.Assign
Token.Identifier ("bal")
Token.Subtract
Token.Identifier ("payment")
Token.Add
Token.Decimal ("0.015")
Token.Multiply

Token.Identifier ("bal")
Token.Semi
Token.Blank
Token.Identifier ("month")
Token.Assign
Token.Identifier ("month")
Token.Add
Token.Constant ("1")
Token.Blank
Token.RBrace
Token.Blank
Token.RBrace
Token.EOF

2 Dragon

1.1.4 The advantage of using C as a target language for a compiler is that it can be run on almost any machine. For example, say a program written in Python needs to be compiled and the machine doesn't have Python installed. Python code can be run through a source-to-source translator to convert it to C. This can then be converted down to machine code if necessary.

1.6.1

```
int w, x, y, z;
int i = 4; int j = 5;
{
  int j = 7; // j is redefined and assigned 7
  i = 6; // i is assigned 6
  w = i + j; // w is 7 + 6 = 13
}
x = i + j; // x uses the first definition of j because it was redefined
    outside the scope of x, and i is a global variable, so when i was assigned
    6 x can use that. So x is 5+6 which is 11.
{
  int i = 8; // i is redefined and assigned 8
  y = i + j; // y can only use the global definition of j so it is 8 + 5 = 13
}
z = i + j; // i was redefined globally as 6 and j's global definition is 5 so
    z is 5+6 = 11.
```

w = 13
x = 11
y = 13
z = 11