Lab One

Grace Kelly

Grace.Kelly1@Marist.edu

January 29, 2019

1 Crafting a Compiler

Exercise 1.11 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?

Answer:

While most plagiarism detecting systems take in a document and strictly search for identical lines of code or information. MOSS is not a system for completely and automatically detecting plagiarism within programming. For MOSS to measure similarities between documents of code, the system takes the standard versions of each document. MOSS is able to detect plagiarism that would easily be overlooked by a professor just scanning over code one by one. Even if a student adds extra comments and white space as well as attempting to change variable names MOSS will be able to detect the similarities of each document of code. MOSS takes the standard versions of each document and compares not just the appearance of each document but also the overall structure of the code. Even with the changes stated above, the structure of code does not change drastically. MOSS is able to calculate the number of lines and tokens produced within each document. The MOSS system will produce a output that displays the similarities between the tokens and lines of each document.

Once MOSS produces the plagiarism output information, it is up to the professor or user of the system to decide what percentage of similarities is plagiarism and what percentage of similarities is normal. Therefore, MOSS produces output that the user has to intemperate to their own discretion.

Exercise 3.1 Assume the following text is presented to a C scanner: (code in book)

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

Answer:

IDENTIFIER main

Open Paren

Close Paren

Open Bracket

const

float

IDENTIFIER payment

Assign

Literal 384.00

semicolon

float

IDENTIFIER bal

semicolon

int

IDENTIFIER month

assign

literal 0

semicolon

IDENTIFIER bal

assign

literal 15000

semicolon

While

Open Paren

IDENTIFIER bal

Greater Than

literal 0

Close Paren

Open Bracket

Printf

Open Paren

literal "Month: %2d Balance: %10.2f
n"

IDENTIFIER month

IDENTIFIER bal

Close Paren

semicolon

IDENTIFIER bal

assign

IDENTIFIER bal

subtraction

IDENTIFIER payment

addition

literal 0.015

multiplication

IDENTIFIER bal

semicolon

IDENTIFIER month

assign

IDENTIFIER month

addition

literal 1

Close Bracket

Close Bracket

Identifier token must have extra information returned with them in the token code.

---

2 Dragon

Exercise 1.1.4 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?

Answer:

The C language has many compilers available for use for various different hardware systems. Therefore, by using C as the target language for a source-to-source translator/compiler it allows for code to be run on different hardware systems. This minimizes the chance of running into a system that can not run the code.

Exercise 1.6.1 For the block-structured C code of Fig. 1.13(a), indicate the values assigned to w x y, and z

(code in book)

Answer:

w = 13

x = 11

y = 13

z = 11