Lab Uno

Jae Kyoung Lee (LJ)

Jae.Lee2@Marist.edu

February 6, 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:

MOSS can detect whether a student cheated or not by uploading the student's file into the system, which MOSS then will compare the student's work to the all documents that were submitted. Students can change variable names, enter random white spaces and code to avoid plagiarism, but the structure of the code is identical, which MOSS will detect as plagiarism (resulting a big negative infinite score). However, nothing is perfect as professors will need to make judgment on some cases since even though the logic is identical, the structure may vary. For instance, if a student use a for/while loop in their code, which MOSS detects as plagiarism, it is up to the professor whether the student plagiarized or not.

Exercise 3.1 : 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?

Answer:

(id_main) (openParen) (closeParen) (openBrace) (const) (float) (id_payment) (assign) (literal_"384.00") (semiColon) (float) (id_bal) (semiColon) (int) (id_month) (assign) (literal_"0") (semiColon) (id_bal) (assign) (literal_"15000") (semiColon) (while) (openParen) (id_bal) (greaterThan) (literal_"0") (closeParen) (openBrace) (printf) (openParen) (literal_"Month: %2d Balance: %10.2f\n") (id_month) (id_bal) (semiColon) (id_ bal) (assign) (id_bal) (minus) (id_payment) (plus) (literal_"0.015") (multiply) (id_bal) (semiColon) (id_month) (assign) (id_month) (plus) (literal_"1") (semiColon) (closeParen) (closeParen)

2 Compilers Principles, Techniques, and Tools (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: Setting C as a target language for a compiler is advantageous since C is a universal language, which means it can be ran on many machines, almost all hardware. Moreover, there are a lot of compilers that translates from multiple languages to C.

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

```
int w, x, y, z;
int i = 4; int j = 5;
{
  int j = 7;
  i = 6;
  w = i + j;
}
x = i + j;
{
  int i = 8;
  y = i + j;
}
z = i + j;
```

Answer:

w, y = 13

x, z = 11