Melissa Iori
CMPT 432 - Compilers
Lab 1

Crafting a Compiler

1.11

MOSS works to find code plagiarism by using a “winnowing” algorithm that hashes the document and compares hashed “fingerprints” to find syntactic and semantic similarities. This thwarts attempts to change just variable names and whitespace in a piece of code, since the fingerprints will still be similar regardless of what variable names are used. Other methods such as Plague, YAP and JPlag transform the source code into a stream of tokens and analyze it for similarity to other source code, but removing comments may not be to their advantage, and they are also vulnerable to switching lines of code around in order.

Sources:
Winnowing in MOSS: http://theory.stanford.edu/~aiken/publications/papers/sigmod03.pdf
Other methods:
http://www.drdobbs.com/architecture-and-design/detecting-source-code-plagiarism/184405734

3.1

T_id - main
T_openParentheses - (
T_closeParentheses -)
T_openBlock - {
T_const - const
T_type - float
T_id - payment
T_assign - =
T_num - 384.00
T_statementEnd - ;
T_type - float
T_id - bal
T_statementEnd - ;
T_type - int
T_id - month
T_assign - =
T_num - 0
T_statementEnd - ;
T_id - bal
T_assign - =

T_num - 15000
T_statementEnd - ;
T_keyword - while
T_openParentheses - (
T_id - bal
T_greaterThan - >
T_num - 0
T_closeParentheses -)
T_openBlock - {
T_id - printf
T_openParentheses - (
T_literal - “Month: %2d Balance: %10.2f\n”
T_comma - ,
T_id - month
T_comma - ,
T_id - bal
T_closeParentheses -)
T_statementEnd - ;
T_id - bal
T_assign - =
T_id - bal
T_subtract - -
T_id - payment
T_add - +
T_num - 0.015
T_multiply - *
T_id - bal
T_statementEnd - ;
T_id - month
T_assign - =
T_id - month
T_add - +
T_num - 1
T_statementEnd - ;
T_closeBlock - }
T_closeBlock - }

All T_id must have their names stored and a pointer into the symbol table.
All T_num must have their values stored.

All T_literal must have their values stored.

Dragon

1.1.4

Using C as a target language for a compiler can be useful if one seeks the advantages of C code, but does not know how to write in C. This person could write a program in a higher-level language, then put it through a source-to-source compiler and receive equivalent C code. It is also useful in that C code is easier to read and verify than the assembly code, bytecodes or hex codes that are produced by many compilers. C is also much more portable than the assembly languages.

1.6.1

w == 13

x == 9

y == 13

z == 9