Frank Mitarotonda   Compilers Exercise

Lab01

Crafting a Compiler

Exercise 1.11: The main difference MOSS uses in its approaches is its unique algorithm. Unlike other plagiarism detectors MOSS does a great job at being whitespace insensitive, meaning that when comparing two files containing code all white space matches are discarded when looking for plagiarism as whitespace is irrelevant. In addition MOSS does good job at only bringing forward any large matches located in the two source code files. Any short matches discovered such as, two alike variable names, are deemed uninteresting and not utilized. In conclusion MOSS also exhibits position independence and is able to capitalize on source code similarities despite the different locations of functions or objects. In my opinion that is the most important ability that MOSS supplies.

Exercise 3.1:

```
{main}{(}{)}{{}{const}{float}{payment}{=}{3}{8}{4}{.}{0}{0}{;}
{float}{bal}{;}
{int}{month}{=}{0}{;}
{bal}{=}{1}{5}{0}{0}{0}{;}
{while}{(}{bal}{>}{0}{ { }
{printf}{(}{“}{M}{o}{n}{t}{h}{:}{%}{2}{d}{B}{a}{l}{a}{n}{c}{e}
{:}{%}{1}{0}{.}{2}{f}{\}{n}{”}{,}{month}{,}{bal}{)}{;}
{bal}{=}{bal}{-}{payment}{+}{0}{.}{0}(1}{5}{*}{bal}{;}
{month}{=}{month}{+}{1}{;}
{}}
{}}
```

You would need more information about bal, month, and payment because they are not keyword identifiers

Dragon Book

Exercise 1.1.4:
The advantages of using C as a target language for a compiler is that the syntax is not as complex as other languages thereby making it easier for the lexer to tokenize. However, I think given any type of high-level language it would still be no easy task for a compiler to translate it into machine-readable code.

Exercise 1.6.1:
W =13
X =9
Y=13
Z=9