CS 4100/5100

Project 1

Due: Wednesday, March 19th at 5:00 pm

In this assignment, we will see how using FLEX for lexical analysis of code can be used to detect potential plagiarism in student assignments. In this assignment we will follow a simplified process similar to the process used by MOSS (Measure Of Software Similarity) which uses the Winnowing Algorithm. That process is as follows.

1. Student programs are tokenized, removing all irrelevant features that are often changed to hide plagiarism, such as variable names, comments, whitespace, etc).
    a. Use LEX code to tokenize the file and remove whitespace and comments
    b. All tokens you assign should be numeric and the same length.
    c. Consider the purpose of your program when deciding what elements should be grouped together as a token, and what elements should not be grouped together.
2. Create digital "fingerprints" for each program based on the tokenized code.
    a. Take in the string of tokens for one student submission
    b. Remove the white space between tokens, leaving one string of digits
    c. Break that string into over-lapping k-mers
        i. k-mers, a string of length k
        ii. Decision of k is up to you.
        iii. The token string "001004003002010" would produce the following overlapping 4-mers: 0010, 0100, 1004, 0040, 0400, 4003 …
        iv. Hash each k-gram using your preferred simple hash function
    d. Select a window size w
        i. For each window a w hashed k-mers, select the minimum value in that window. This is the fingerprint of that window
        ii. Store the fingerprints for each window of the total set of hashed k-mers.
3. Once all submissions have been tokenized, broken into k-mers, hashed, and fingerprinted, you are willing to compare pairs of submissions
    a. for each pair of submissions a and b, you score the similarity of the two submissions as follows
        i. Find the number of SHARED fingerprints between the two submissions a and b
        ii. Divide the number of shared fingerprints by the total number of fingerprints between a and b. This is the similarity score between a and b
    b. Once you have found the similarity score for ALL pairs of submissions, rank those pairs from high to low similarity and create a report file containing this information
4. Then MANUALLY look at the top pairs identified by the program. In your report, discuss whether or not these submissions appear to indicate plagiarism. In your report discuss if you see a clear threshold for similarity that indicates that submissions should be further investigated for plagiarism

Your program will be CMOS, a MOSS like program that works on C code. The program will take in one command line argument, which is the name of the directory containing student examples.

In order to complete this project you will need to complete:

1. cmos.l : A LEX program that can parse and tokenize one student submission
2. cmos.cpp : A C++ program that can read in one file containing all tokenized submissions (tokens.txt) and performs the fingerprint analysis as described above and in the Winnowing Algorithm paper.
3. A makefile to compile your project
4. A Bash script (provided) that will read through the directory of examples (directory name taken in as an argument), tokenize each one, prepare the token.txt file for CMOS to analyze, and calls CMOS to perform the analysis

Additionally, you should include a brief project report outlining the following aspects of your project. Your report should be converted to a PDF before submission.

1. What regular expressions/tokens did you identify, and how does your program react to each one.
2. A brief description in your own words of how you implemented the Winnowing algorithm
3. The results of your analysis, including identifying any submissions found by the algorithm that you believe may be plagiarism

You will submit all required code on Canvas. You do not need to include the example files. FLEX and g++ often do not play well together, and versioning is really important. I recommend working on one of the school prime machines, or at least testing your code on pu3 (the prime machine I will use to test) to ensure that everything is working correctly.

Group Work Policy

You may work with one partner on this assignment, but you are not required to do so. We will not need this project for a later stage, so you are not tied to that partner for the rest of the semester. Only one group member should submit, and they should include both partners names in all code file, the project report, and as a comment in the Canvas submission.

Late Policy

Late projects are subject to a 20% penalty for up to 24 hours after the due date. After 24 hours, the assignment will no longer be accepted. You are responsible for submitting the correct file for your project submission and for submitting on time.