Parametrized String Matching Implementation for Software Plagiarism Check

Amit Tomar (MT2013008)
Siddhesh Dosi (MT2013150)
Srinivas R. Vaidya (MT2013152)

International Institute of Information Technology, Bangalore

April 15, 2014

Objective

- This project aims at developing a Parametrized String Matching Implementation for Software Plagiarism Check, that given a collection of files which contain code in some programming language, will show a set of possible duplications of parts of the code among these.
- Comparing pieces of software will require discounting comments (optional and language dependent), extra/blank lines and spaces, variable renaming etc.
- The theory of parametrized string matching will be used to implementation this project.
- System will have an easy-to-use UI for selecting files/folders and shall report the plagiarism related information (matches found) in the UI in a nice manner.

Functionality

- Plagiarism check among two given files.
- Pairwise plagiarism check among all the files in a given folder.
- Ignoring a code snippet for plagiarism check.

Technology

- Exploring C++ / Python as option.

Coding standard to be followed

1. **Class**
   All class names should begin with capital letter, with all subsequent words beginning with a capital letter too.
   eg.
   class ThisIsAClassName
   class Administration
   class FooClass

2. **Function**
   All function names should begin with a small letter, with all subsequent words beginning with a capital letter.
   eg.
   void thisIsAFunction(void);
   int fooFunction (string) ;

   Names representing methods or functions must be verbs. eg.
   int getSystemVolume(void) ;
   void setSytemContrast(int) ;
   char findFirstCharacter(void) ;

Coding standard to be followed

3. **Variables**
   All primitive data type variables must begin with the first character of data type in small followed by the name of variable starting with a capital letter. Subsequent words will have their first character capital.
   eg.
   dataType dThisIsAVariableName
   int iSystemVolume;
   char cAnAlphabet;
   float fCurrentMonthSalary;
   long lVolumeOfWater;
   double dTotalTtax;
   string sMyName;
   bool bIsSet;

   Use of global variables should be avoided as far as possible.

Coding standard to be followed

4. **Objects of class** All object names must begin with "obj" followed by the exact class name.
   eg.
   FooClass objFooClass;
   In case multiple instances of a class are to be used, above described name is to be followed by some information about the object. All these subsequent words must begin with a capital letter.
   eg.
   Employee objEmployeeAmit;
   Employee objEmployeeHemant;
   Employee objEmployeeKaustubh;

5. **Pointer types**
   If a variable is of pointer type, then its name should be preceded by "ptr_"
   eg.
   int * ptr_iSalary;
   char * ptr_cNewCharacter;
   FooClass * ptr_objFooClassMemory;

Coding standard to be followed

6. **Array type** All array names should be preceded by "arr_"
   eg.
   int arr_iSalary [10];
   char arr_cNewCharacter [200] ;
   FooClass arr_objFooClassMemory [5] ;

7. **List type**
   All list names should be preceded by "lst_"
   eg.
   list[int] lst_iRollNumber;

Brief summary of work done

- **Requirement Specification** : Date of submission : 7 - Feb - 2014
  (Already submitted to Prof. Srinivasraghvan)

- **Literature Survey** : Expected date of completion : 20 - April - 2014
  Expected time : 40 Hours

Plan for remaining work

- **Building suffix tree data structure** : Expected date of completion : 15-May-2014, Expected time : 30 Hours
- **Identifying duplicate code using suffix tree**: Expected date of completion : 1-June-2014, Expected time : 25 Hours
- **Implementation of UI**: Expected date of completion : 10-June-2014, Expected time : 10 Hours
- **Parameterized implementation for software plagiarism check**: Expected date of completion : 1-July-2014, Expected time : 40 Hours
- **Integration of UI with parameterized string matching code**: Expected date of completion : 15-July-2014, Expected time : 20 Hours
- **Testing**: Expected date of completion : 31-July-2014, Expected time : 20 Hours

Tools used

S.No.	Software	Version	Purpose
1	Ubuntu Linux	13.04	Operating system.
2	GitHub	1.8.3.2	Version Control
3	Spyder	2.2.1	IDE for Python
4	GIMP	2.6	Image editing for documentation
5	Gummi	0.6.5	LaTeX editing for documentation

References

1 Peter Vamplew, Julian Dermoudy, *An Anti-Plagiarism Editor for Software Development Courses*, Proceeding ACE-05 Proceedings of the 7th Australasian conference on Computing education - Volume 42 Pages 83- 90 , 2005.