iJRASET
International Journal For Research in Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 1 **Issue:** I **Month of publication:** August 2013

DOI:

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

INTERNATIONAL JOURNAL FOR RESEARCH IN APPLIED SCIENCE AND ENGINEERING TECHNOLOGY (IJRASET)

LEXICAL ANALYSIS FOR THE MEASUREMENT OF CONCEPTUAL DUPLICITY BETWEEN C PROGRAMS

[1] **Akhil Gupta,** [2] **Dr. Sukhvir Singh**

[1] *NCCE, Israna, Pin no.132107,* [1] *akhil2108@gmail.com*

[2] *NCCE, Israna, Pin no.132107,* [2] *boora.s@yahoo.com*

Abstract: *Plagiarism in programming assignment is major problem which motivate the need of faster detection approach then manual checking. Detecting source code plagiarism in programming assignments through manual inspection process is very error-prone. Detect plagiarism in source code to maintain the concept or design of the original programmer or creator. Program similarity checking is an important application in the field of education. Recent article unveil that many student find guilty in source code plagiarism.*

Keywords: *Plagiarism in programming assignments, Detecting source code plagiarism, Text plagiarism, source code plagiarism, Conceptual duplicity.*

1. INTRODUCTION

In the present IT driven world, internet has a profound influence on the human society in every aspect, at the same time it is also leading to an increased amount of plagiarism in many formats. Apart from the internet, plagiarism is spreading its wings in the area of "programming assignments" also.

Program similarity checking is an important application in the field of education. Recent article unveil that many student find guilty in source code plagiarism.

In that case automation has to provide which can address such type of problem. So I propose a tool which measure the source code plagiarism between programming assignment in other words I would like to say that it checks the conceptual duplicity between two programs. Checking the conceptual duplicity refer to the concept or logic between programs.

Plagiarism can be classified in two categories. It can be either text plagiarism or source code plagiarism. In text plagiarism, text document has to be check with other text document and lot of research has been already done in this field. Source code plagiarism is nothing but checking the figment between source codes. Source code plagiarism assure about conceptual duplicity between programming assignments.

To development of source code plagiarism detection tool we used XML text file which plays an important role in the measurement of conceptual duplicity between two c program.

In starting c source code in converted in formatted code after that we will identify the key element or structure of source code and then generate XML file which represent the source code. Identification of key elements can be done via lexical analysis. There are some tools available which directly identify the tokens in c source code without writing any code. After applying the lexical analysis, c source code would convert in XML text file corresponding to the above formatted code generated.

Then generate the different similarity comparison algorithm which measure the similarity of source code and on the basis of those algorithms we calculate the procedural similarities under defined source code plagiarism categories.

2. LITERATURE SURVEY

- MOSS (measurement of software similarities) is the existing plagiarism detection system which is used for detecting plagiarism for programs written in procedural languages. MOSS is able to detect following plagiarism: completing copy, changing comments, adding spaces, changing the order of independent statement.
- Sim is a tool which is used to measure structural similarity between two c program. Sim uses a string alignment in a given program. Sim was robust to common modification such as name changes, reordering of statement and adding/removing comment or white spaces.
- BUAA_AntiPlagiarism is system whose output is a group of clusters of all suspicious plagiarized programs after calculating the pair wise similarities.
- JPlag is a system that finds similarities among multiple sets of source code files. JPlag considers the syntax and structures of program.

- Pk2 tool detect the plagiarism in programming assignment. Pk2 tool process each given project source file, transforming it into an internal representation.

3. PROBLEM FORMULATION

In today's scenario source code cannot be judge only by plagiarism but now special attention is also required towards conceptual duplicity among source code so we require a tool which detects the plagiarism between source code in other words I would say that a tool which is responsible to check the conceptual duplicity between programs. Checking the conceptual duplicity refer to the concept, logic or figment between program.

CATEGORIES OF CONCEPTUAL DUPLICITY

- Changing Data types
- Changing the order of statement
- Changing the order of block of statement
- Rename Identifier
- Changing the operator sequence
- Changing the operand sequence
- Adding redundant statement
- Completing Copy
- Changing comments
- Replacing control structure with equivalent control structure

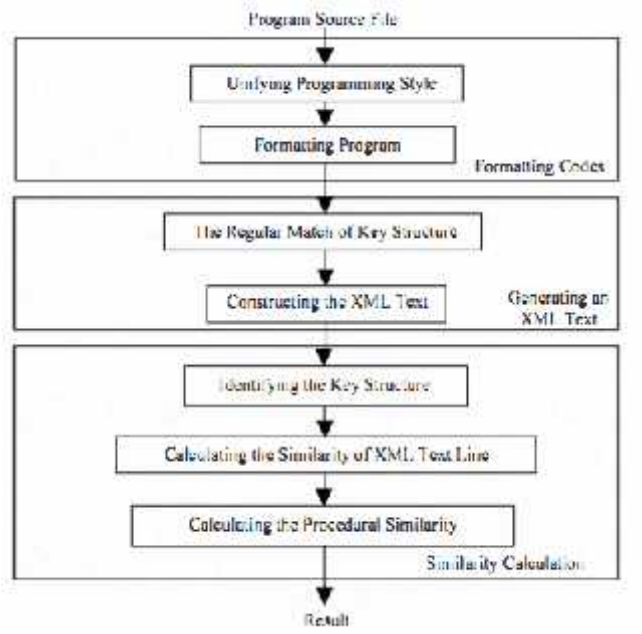
4. PROPOSED SOLUTION

Measurement of conceptual similarities between two c source codes will be done on the basis of XML text file.

Steps that are required to perform in order to achieve target.

- Formatting Code
- Generate the XML text which represents the source code.
- Identify the key structure.
- According to different key structure we design a different similarities comparison algorithm to calculate the procedural similarities.

Architecture of source plagiarism detection tool has been divided into three phases. Phase first, second and third known as formatted code, generating an XML text and similarity calculation respectively.



First phase has two part unifying programming style and formatting program. Second phase of architecture also has two parts regular match of key elements and generate XML text whereas last phase of architecture has three parts indentifying key elements, calculating the similarities of XML text line and finally calculate the procedural similarity.

5. DISCUSSION & FUTURE WORK

In the course of our thesis we have faced many new challenges. The main challenge was to develop the architecture that can measure the conceptual duplicity between two c source codes. We have seen that there are some other tools available which is used for the same purpose but they all have some limitation.

Our plagiarism detection tools will measure conceptual duplicity on the basis of 10 categories. There categories are Changing Data types, Changing the order of statement, Changing the order of block of statement, Rename Identifier, Changing the operator sequence, Changing the operand sequence, Adding redundant statement, Completing Copy, Changing comments, Replacing control structure with equivalent control structure.

6. FUTURE WORK

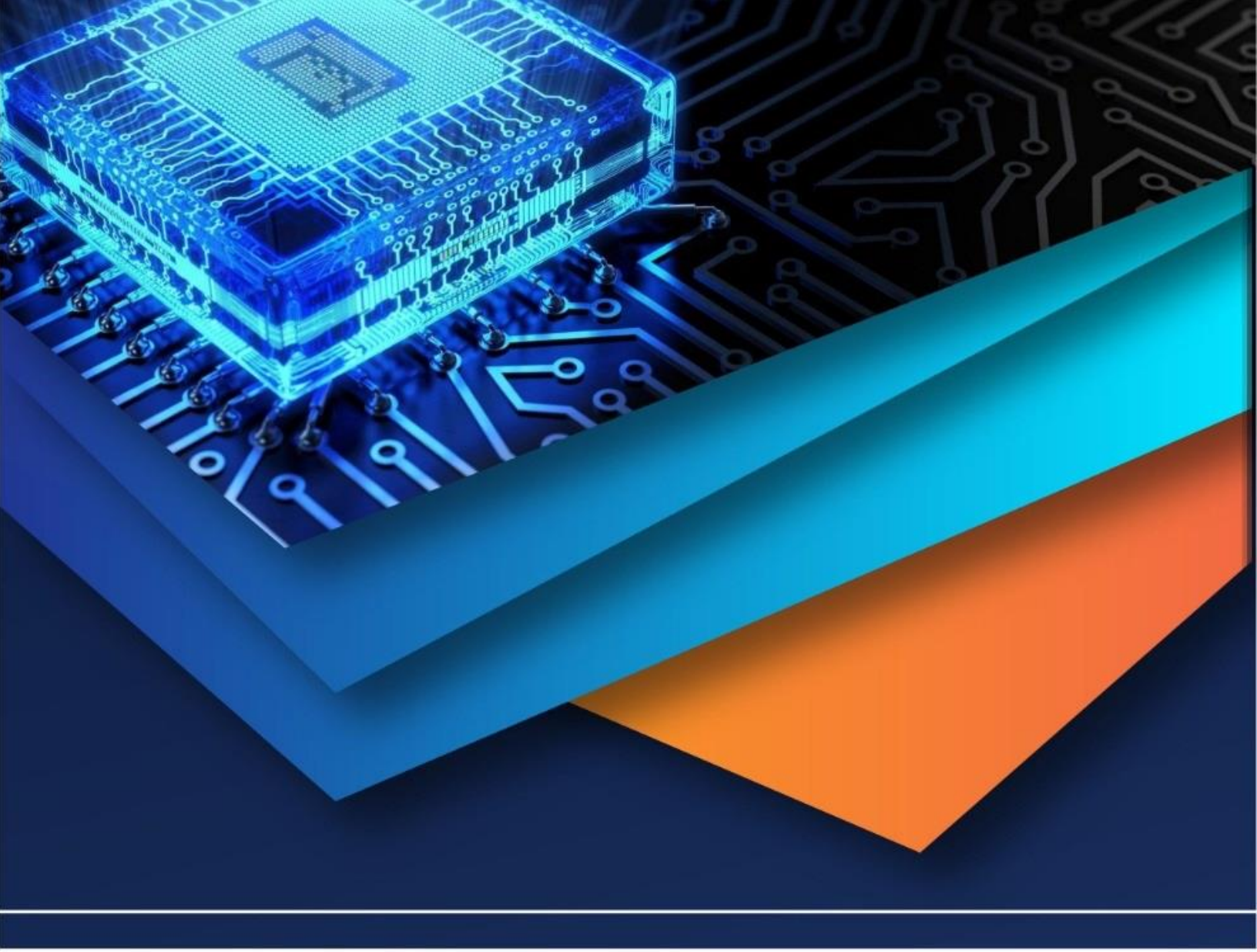
This entire thesis work is divided into three phases. First one deals with the formatted code which has been completed. We are left with two phases which are stated below:

We will identify the key element or structure of source code and then generate XML file which represent the source code. Identification of key elements can be done via lexical analysis.

According to the different token string of XML text line, the plagiarism detection model based on XML calls different similarity calculation function to calculate the similarity of XML text line.

REFERENCES

1. S.Schleimer, D.Wilkerson, A.Aiken: Winnowing: Local Algorithms for Document Fingerprinting. ACM SIGMOD 2003[c]. San Diego: ACM press,2003 pp 204-212
2. D.Gitchell, N.Tran, Sim: A Utility For Detecting Similarity in Computer Programs. ACM SIGCSE 1999[c]. New orleans :ACM press, 1999 pp 266-270
3. H.Xiong, H.Yan, Z.Li, BUAA AntiPlagiarism: A System to Detect Plagiarism for C Source Code. 2009 IEEE.
4. R.Brixtel,M.Fontaine: Language-Independent Clone Detection Applied to Plagiarism Detection. 2010 Working Conference on Source Code Analysis and Manipulation.
5. F. Rosales, A. Garcia: Detection of Plagiarism in Programming Assignments.
6. A. Aiken, Moss: A system for detecting software plagiarism.
7. K. Emeric, K. Moritz, JPlag: A system that finds similarities among multiple sets of source code files.
8. M. Szuskiewicz: Automatic Plagiarism Detection in Software Code. Information and Communications Technology.



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY