# Plagiarism Detection Techniques

*Nimisha .T*

*13MCA11030*

# Contents:

- Introduction
- Definition of Plagiarism
- Avoiding plagiarism
- Text based plagiarism detection techniques
- Tools used for text based plagiarism
- Source code based plagiarism detection techniques
- Tools used for code based plagiarism
- Disadvantages of the plagiarism detection technology
- Conclusion

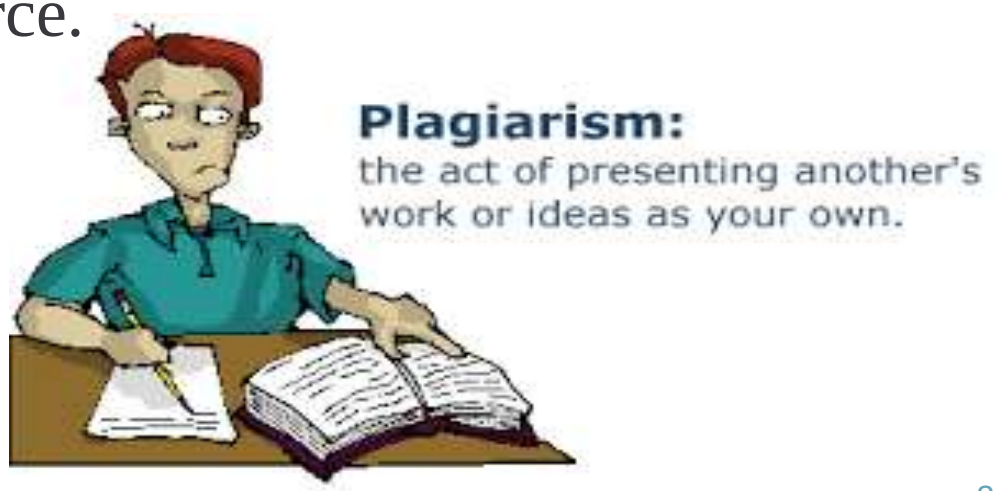
# Introduction

- Plagiarism is a significant problem on almost every college and university campus.
- The problems of plagiarism go beyond the campus, and have become an issue in industry, journalism, and government activities.

# Definition of Plagiarism

Plagiarize according to the Merriam-Webster Online dictionary is:

- To steal and pass off the idea or words of another as one's own.
- To use another's production without crediting the source
- To commit literary theft
- To present as new and original idea or product derived from an existing source.



# The following are considered as Plagiarism:

- Turning in someone else's work as your own.
- Copying words or ideas from someone else without giving credit.
- Failing to put a quotation in quotation marks
- Giving incorrect information about the source of a quotation.
- Changing words but copying sentence structure.
- Copying so many words or ideas from a source that it makes up the majority of your work, even though by credit.

# Deliberate and Accidental Plagiarism

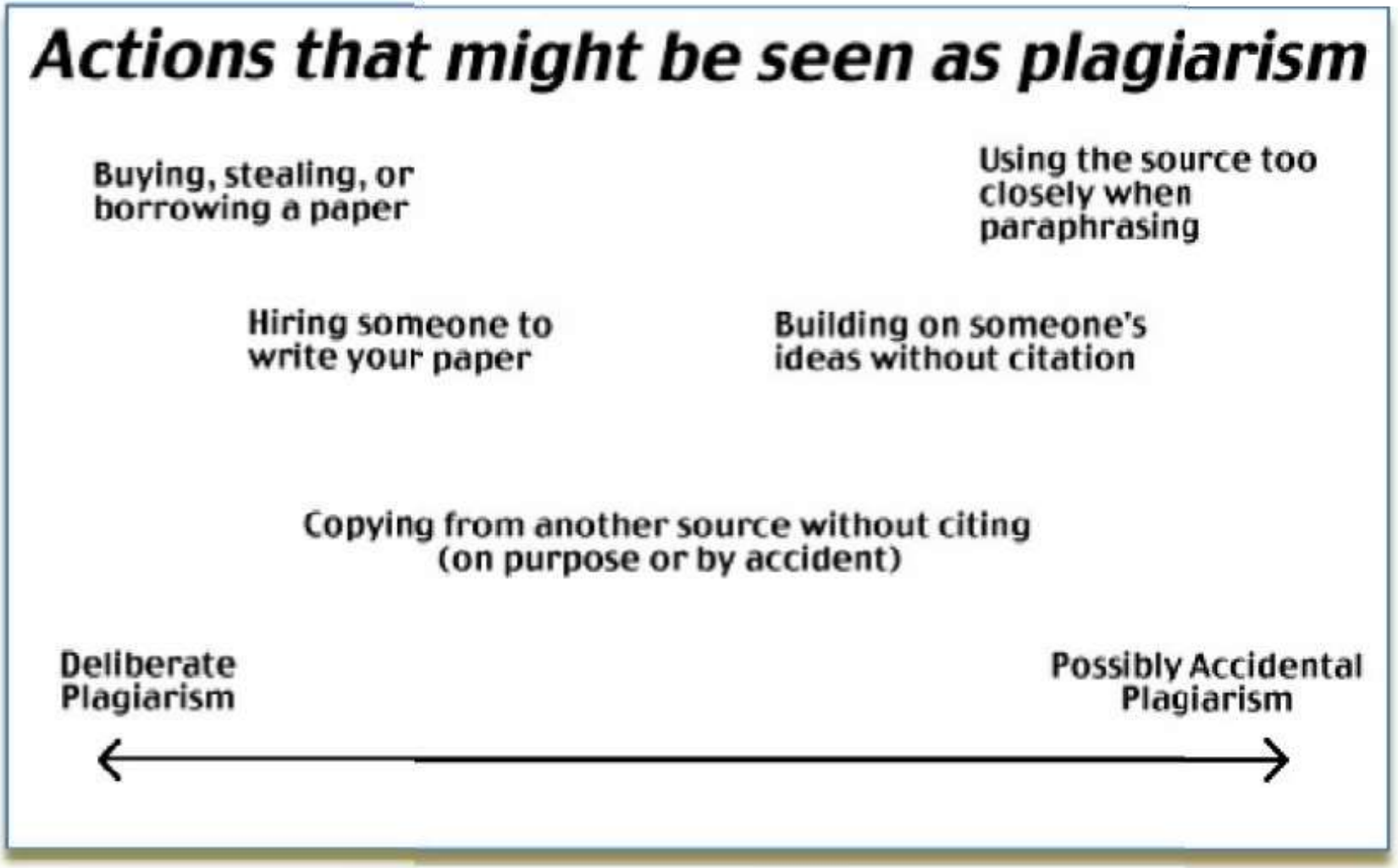
- Deliberate (intentional) Plagiarism :

  Steals the property of somebody else and claims it to be his own.

- Accidental (unintentional) Plagiarism :

  Somebody unknowingly cites a phrase or copies words without acknowledging the author of the material.

# Deliberate and Accidental Plagiarism



# Avoiding plagiarism

Two methods :

- Plagiarism prevention
- Plagiarism detection

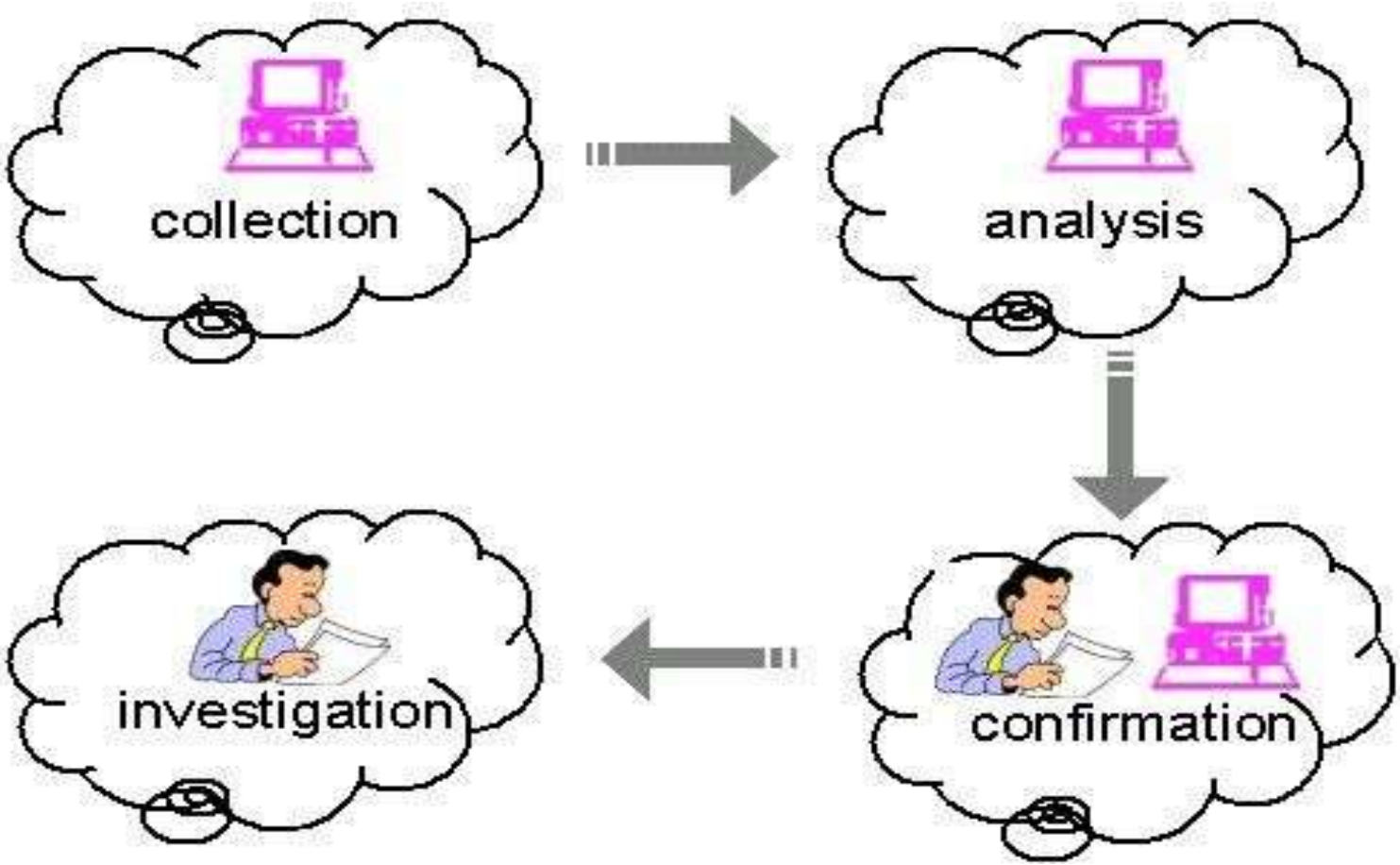
# Plagiarism Prevention


- Collaborative effort for recognize and counter plagiarism at every level.
- Educate students about the appropriate use of intellectual material.
- Minimize the possibility of submission of plagiarized content.

- Plagiarism prevention is difficult to achieve & also take a long time.

# Plagiarism Detection


Culwin and Lancaster's four stages of detecting plagiarism:



# Plagiarism Detection technique

- **Text based plagiarism detection techniques**
- **Source code based plagiarism detection techniques**

# Text based plagiarism detection techniques

- Substring matching
- Keyword similarity
- Exact fingerprint match
- Text parsing

# Substring Matching

- Try to identify maximal matches in pairs
- which then are used as plagiarism indicators.
- Typically, the substrings are represented in suffix trees.
- Graph-based measures are employed to capture the fraction of the plagiarized sections.

# Keyword Similarity

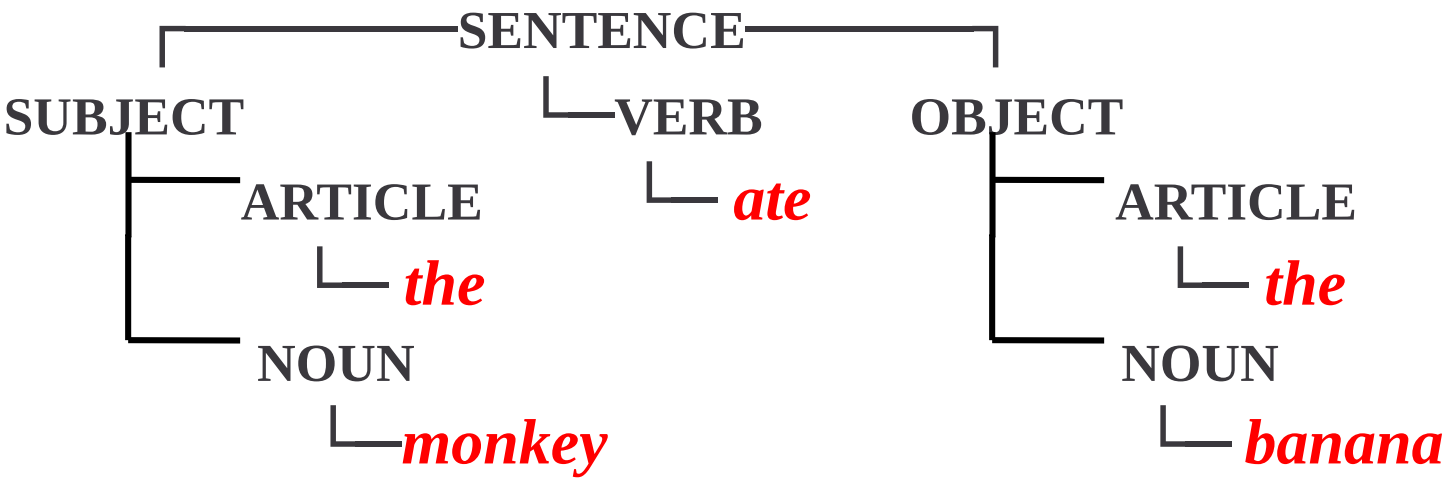
- Extract topic identifying keywords from a document.
- Compare with keywords of other document.
- If the similarity exceeds a threshold, the candidate documents are divided into smaller pieces.
- which are then compared recursively.
- This approach assumes that plagiarism usually happens in topically similar documents.

# Exact Fingerprint Match

- The documents are partitioned into term sequences called chunks.
- which then are used as plagiarism indicators.
- from which digital digests are computed that form the document's fingerprint.
- digests are inserted into a hash table then collisions indicate matching sequences.
- For the fingerprint computation some standard hashing suffers from two severe problems:
  - Computationally expensive,
  - A small chunk size (3-10 words) must be chosen to identify matching passages

# Text parsing

- Any sentence of the text can be automatically represented in the form of the tree.
- which reflects the structure of the sentence
- Example: The phrase *the monkey ate the banana* will be parsed by such software as,



# Text parsing (Continue…)

- Once a parse tree is created, we can invoke a tree matching procedure
- Initially the algorithm builds a flowchart-styled parse tree for each file to be analyzed
- Then for each pair of files, the algorithm performs a rough “abstract comparison”, when only types of the parse tree elements ( like Assignment, Loop, Branching) are taken into account.
- This is done recursively for the each level of tree nodes If the similarity percentage becomes lower, the trees are immediately treated as not similar.

# Text parsing (Continue…)

- If the abstract comparison indicates enough similarity, a special low-level "micro comparison" procedure is invoked.
- Each node represents an individual statement
- Each tree node turns into a separate sub tree that has to be compared with the corresponding sub tree taken from another file.
- E.g. the phrases *the monkey ate the banana* and *the banana was eaten by the monkey* will be very close after the tokenization.

# Tools used for text based plagiarism

**Some tools are:**

- PlagAware
- PlagScan
- CheckForPlagiarism.net
- iThenticate
- PlagiarismDetection.org

# PlagAware

- Is an online-service used for plagiarism detection
- It can search, find, analyze and trace plagiarism in the specified topic similar to the topics
- PlagAware is a search Engine
- provide different types of report that help the user to decide that is his document has been plagiarized or not
- Mainly used in academic filed
- Multiple Document Comparison
- Does not support synonym and sentence structure checking.

# PlagScan

- It is online software used for textual plagiarism checker
- Complex algorithms for checking and analyzing uploaded document
- Unique signature extracted from the document's structure that is then compared with PlagScan database and millions of online documents.
- Detect most of plagiarism types either directs copy and paste or words switching
- PlagScan supports all the language that use the international UTF-8 encoding and all language with Latin or Arabic characters

# CheckForPlagiarism.net

- One of the best online plagiarism checkers that used to stop or prevention of online plagiarism.
- The fingerprint-based approach used to analyze and summarize collection of document and create a kind of fingerprint for it.
- Uses its own database that include millions documents and articles over World Wide Web
- Support synonym and sentence structure checking
- Can compare set of different documents simultaneously with other documents

# iThenticate

- One of the application or services designed especially for the researchers and authors' publisher
- It have own database that contain millions of documents
- Users who have account can do either online and offline comparison of submitted documents against it and to identify plagiarized content.
- Considered as the first online plagiarism checker
- Document to document and multiple documents checking
- Supports more than 30 languages
- Does not support synonym and sentence structure checking

# PlagiarismDetection.org

- It is an online service provides high level of accuracy result in plagiarism detection
- Use its own database that contains millions of documents
- Supports English languages and all languages that using Latin characters
- Does not support multiple document comparison
- Does not support synonym and sentence structure checking

# Source code based plagiarism detection techniques

- Lexical Similarities
- Parse Tree Similarities
- Program Dependence Graphs
- Metrics

# Lexical Similarities

- Converts source code into a stream of lexical tokens from which compiler extract meaning from the source.
- During the lexical analysis phase, the source code undergoes a series of transformation
- Some of these transformations, such as the identification of reserved words, identifiers are beneficial for plagiarism detection.

# Lexical Similarities (Continue…)

- Consider the following two snippets of Java Code:

```
int[] A = {1,2,3,4};
for(int i = 0; i < A.length; i++)
{
A[i] = A[i] + 1;
}
```

```
int[] B = {1, 2, 3, 4};
for(int j = 0; j < B.length; j++)
{
B[j] = B[j] + 1;
}
```

# Lexical Similarities (Continue…)

The lexical stream of the 2 snippets of code is :

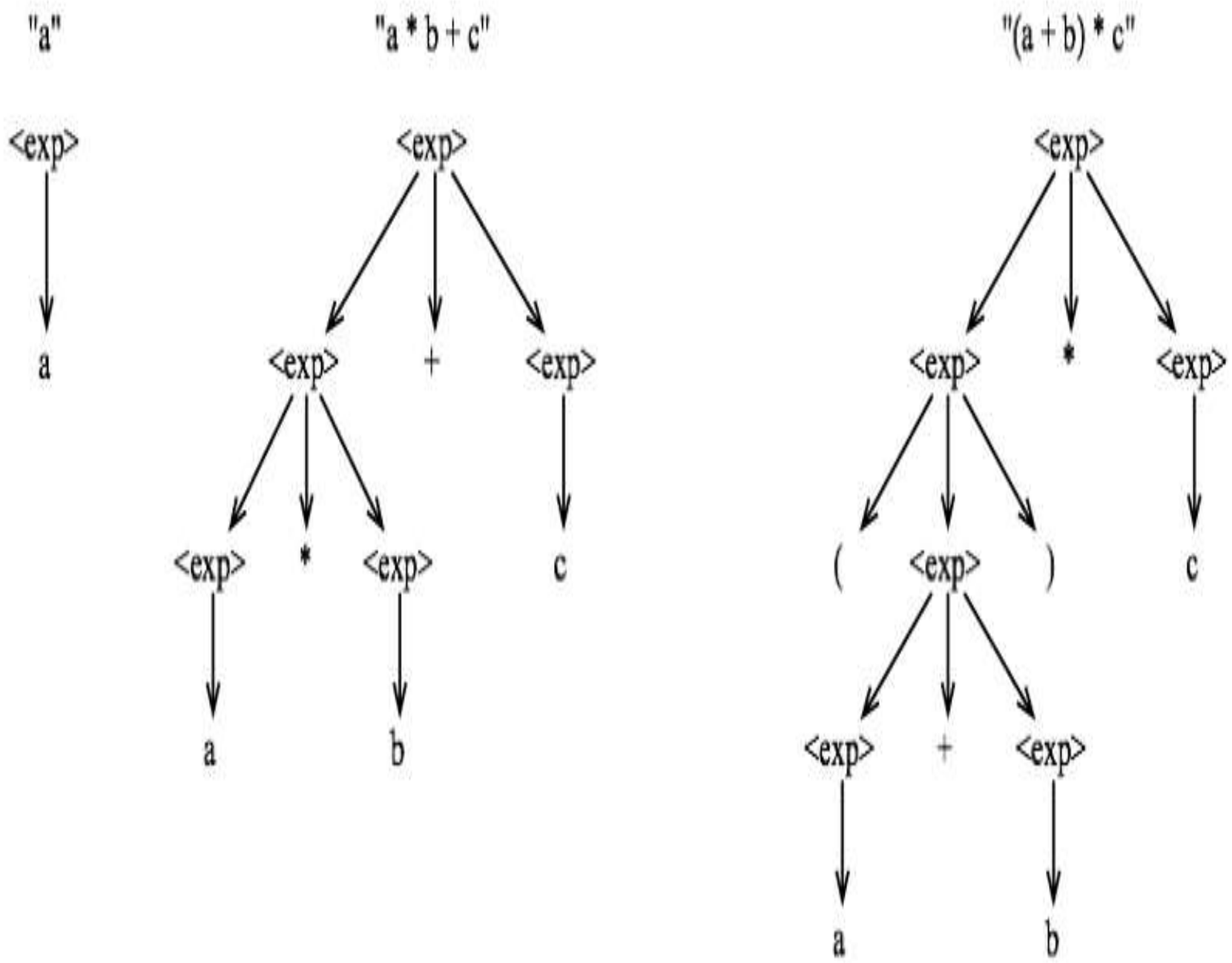
LITERAL_int LBRACK RBRACK IDENT ASSIGN LCURLY NUM_INT COMMA NUM_INT COMMA NUM_INT COMMA NUM_INT RCURLY SEMI

LITERAL_for LPAREN LITERAL_int IDENT ASSIGN NUM_INT SEMI IDENT LT IDENT DOT IDENT SEMI IDENT INC RPAREN LCURLY NUM_INT SEMI

Both the java snippets will have the exact lexical stream

# Parse Tree Similarities

- The parse tree or derivation tree built from the lexical for a program also exhibits structure for a given program
- A compiler, during the compilation process builds a parse tree which represents the program.
- The parse tree will have the same structure for both the snippet of code as the lexical streams are same.
- An algorithm for detecting plagiarism using this method would first, parse each program.
- Next, for each pair of parse trees, it attempts to find as many common sub trees as possible.
- Use this number as a measure of similarity between the two programs

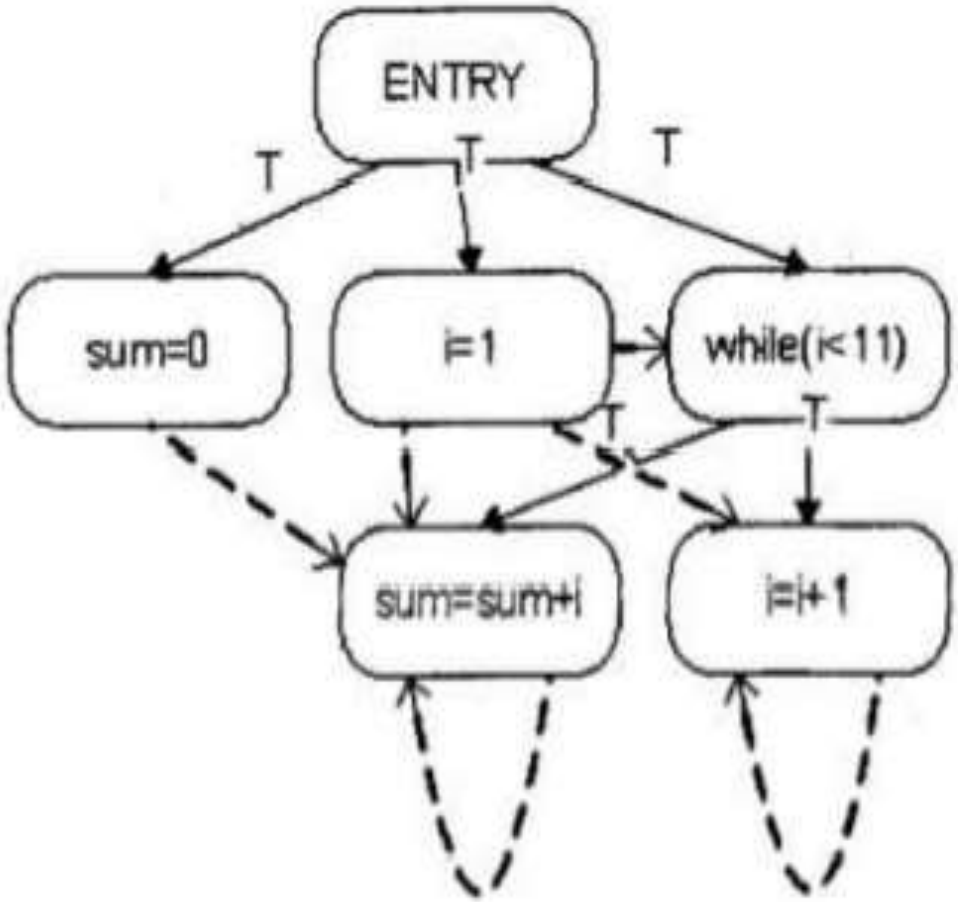

"a"
"a * b + c"
"(a + b) * c"
<exp>
a
<exp>
<exp>
+
<exp>
<exp>
*
<exp>
a
b
c
<exp>
<exp>
*
<exp>
(
<exp>
)
c
<exp>
+
<exp>
a
b

# Program Dependence Graphs(PDG)

- PDG is a graph representation of the source Code
- It is a directed, labeled graph which represents the data and the control dependencies within one procedure.
- Basic statements like variable declarations, assignments, and procedure calls are represented by vertices in PDGs.
- It depicts how the data flows between statements and how statements are controlled by other statements.
- The data and control dependencies between statements are represented by edges between vertices in PDGs
- Data and control dependencies are plotted in solid and dashed lines respectively.

# Program Dependence Graphs (Continue…)

Example:




```
void sum{
int i, sum;
sum=0;
i=1;
while(i<11){
sum=sum+i;
i=i+1;
}}
```

# Metrics

- Plagiarism detection by similarity analysis using software metrics.
- Software metrics are:
  - Number of function calls
  - Number of used or defined local variables
  - Number of used or defined non-local variables
  - Number of parameters
  - Number of statements
  - Number of branches
  - Number of loops

# Metrics (Continue…)

- Each fragment characterized by a set of features measured by metrics
- Metrics computation requires the parsing of source code to identifying interesting fragments
- Metrics are simple to calculate and can be compared quickly
- False positives: two fragments with the same scores on a set of metrics may do entirely different things.

# Tools used for code based plagiarism

- **MOSS**
- **JPlag**
- **CodeMatch**

# MOSS (Measure of Software Similarity)

- Can be applied to a range of programming languages
- Registered instructors can submit batches of programs to the moss server.
- Result is placed on a web page on the moss web server.
- A link to that web page is returned when checking the document is finished.
- The MOSS database stores an internal representation of programs, and then looks for similarities between them.

# JPlag

- JPlag compares submitted programs in pairs
- It assumes that plagiarists may vary the names of variables or classes, but they are least likely to change the control structure of a program.
- It presents its results as a set of HTML pages.
- The pages are sent back to the client and stored locally.
- JPlag was easier to use but supported fewer languages than MOSS

# CodeMatch

- Compares thousands of source code files in multiple directories and subdirectories
- Determine those files which are closely correlated.
- Useful for finding open source code within proprietary code.
- Discovering common standard algorithms within different programs.

# Disadvantages of the plagiarism detection technology

- Plagiarism detection systems are built based on a few languages. To check for plagiarism with the same software can be difficult.
- Most of the detection software checking is done with some repository situated in an organization. Other people are unable to access it and verify for plagiarism.
- As the number of digital copies are going up the repository size should be large and the plagiarism detection software should be able to handle it.
- Some software ask us to load a file to their link .The file is copied to their database . This cause our data being leaked or hacked for other purposes.

# Conclusion

- Plagiarism is rampant now. With most of the data available to us in digital format the venues for plagiarism is opening up.
- To avoid this kind of cheating and to acknowledge the originality of the author new detection techniques are to be created.
- To protect the intellectual property source code new techniques are to be developed and implemented.

# References


- http://www.ijirae.com/volumes/vol1/issue7/AUCS10085.06.pdf
- http://dspace.cusat.ac.in/jspui/bitstream/123456789/3618/1/PDT.pdf
- http://elearningindustry.com/top-10-free-plagiarism-detection-tools-for-teachers
- http://www.plagiarism.org/plagiarism-101/what-is-plagiarism/
- http://www.cs.uu.nl/research/techreps/repo/CS2010/2010-015.Pdf
- https://en.wikipedia.org/wiki/Category:Plagiarism_detectors


Q&A
time
?

# “Thank You”