Table of Contents

1 Introduction

2 Problem Analysis

3 Other work

4 Presentation
Code-Fragment

- any number of lines of a program
- begin-end-block (like a function)
- sequence of simple statements
As a Code Clone we will consider two code fragments, what are similar in a before defined way.
Clone-Types

Type-1 differ only in layout, whitespace and comment
Type-2 differ in identifiers and literals
Type-3 added and/or removed statements
Type-4 use different syntax to do the same computation
What do we want?

A program that does
- find clones
What do we want?

A program that does

- find clones
- without too many false positives
What do we want?

A program that does

- find clones
- without too many false positives
- in a fast way
Why do students copy code?

- too lazy to program on their own
Why do students copy code?

- too lazy to program on their own
- they just cannot do the homework
Why do students copy code?

- too lazy to program on their own
- they just cannot do the homework

Resulting in mostly Type-1 and Type-2 clones.
Why do students copy code?

- too lazy to program on their own
- they just cannot do the homework

Resulting in mostly Type-1 and Type-2 clones.

No Type-4 clones wanted

All programs are Type-4 clones!
A basic clone detection based on the Diff algorithm
- load the files
A basic clone detection based on the Diff algorithm

- load the files
- remove unaltered code
File-Diff

A basic clone detection based on the Diff algorithm

- load the files
- remove unaltered code
- remove too short files
File-Diff

A basic clone detection based on the Diff algorithm
- load the files
- remove unaltered code
- remove too short files
- compare all files and check for clones
A basic clone detection based on the Diff algorithm

- load the files
- remove unaltered code
- remove too short files
- compare all files and check for clones
- output found clones
Disadvantages:
- easy to counter with layout, ...
- only finds whole copied files, not partly copied ones

Advantages:
- very easy to implement
- does always work (program could not compile)
A clone detection algorithm working on the tokens of the program

- create Abstract Syntax Tree (AST)
Function-Diff

A clone detection algorithm working on the tokens of the program

- create Abstract Syntax Tree (AST)
- find all functions
Function-Diff

A clone detection algorithm working on the tokens of the program

- create Abstract Syntax Tree (AST)
- find all functions
- get the tokens for the functions
Function-Diff

A clone detection algorithm working on the tokens of the program

- create Abstract Syntax Tree (AST)
- find all functions
- get the tokens for the functions
- remove too short functions
Function-Diff

A clone detection algorithm working on the tokens of the program

- create Abstract Syntax Tree (AST)
- find all functions
- get the tokens for the functions
- remove too short functions
- find unchanged functions
Function-Diff

A clone detection algorithm working on the tokens of the program

- create Abstract Syntax Tree (AST)
- find all functions
- get the tokens for the functions
- remove too short functions
- find unchanged functions
- compare all functions with each other (Diff algorithm)
Function-Diff

A clone detection algorithm working on the tokens of the program

- create Abstract Syntax Tree (AST)
- find all functions
- get the tokens for the functions
- remove too short functions
- find unchanged functions
- compare all functions with each other (Diff algorithm)
- output all clones
Disadvantages:
- The code must be compileable

Advantages:
- Layout does not matter anymore
We did not use existing projects, because ..

- not locally run
Other work

We did not use existing projects, because ..

- not locally run
- not free of charge
We did not use existing projects, because ..

- not locally run
- not free of charge
- not open source
Other work

We did not use existing projects, because ..

- not locally run
- not free of charge
- not open source
- the overall performance was not to good
Program presentation.
Thank you for listening.