LLVM: built-in scalable code clone detection based on semantic analysis

Institute for System Programming of the Russian Academy of Sciences

Sevak Sargsyan : sevaksargsyan@ispras.ru
Shamil Kurmangaleev : kursh@ispras.ru
Andrey Belevantsev : abel@ispras.ru
Considered Clone Types

1. Identical code fragments except whitespaces, layout and comments.

2. Identical code fragments except identifiers, literals, types, layout and comments.

3. Copied fragments of code with further modifications. Statements can be changed, added or removed.
Considered Clone Types: Examples

**Original source**

```c
4: void sumProd(int n) {
5:     float sum = 0.0;
6:     float prod = 1.0;
7:     for (int i = 1; i <= n; i++) {
8:         sum = sum + i;
9:         prod = prod * i;
10:    } foo(sum, prod);
11: }
12: }
```

**Clone Type 1**

```c
void sumProd(int n) {
     float sum = 0.0; //C1
     float prod = 1.0; // C2
     for (int i = 1; i <= n; i++) {
         sum = sum + i;
         prod = prod * i;
         foo(sum, prod);
     }
}
```

**Clone Type 2**

```c
void sumProd(int n) {
     int s = 0; //C1
     int p = 1; // C2
     for (int i = 1; i <= n; i++) {
         s = s + i;
         p = p * i;
         foo(s, p);
     }
}
```

**Clone Type 3**

```c
void sumProd(int n) {
     int s = 0; //C1
     int p = 1; // C2
     for (int i = 1; i <= n; i++) {
         s = s + i * i;
         foo(s, p);
     }
}
```

- Tabs and comments are added
- Variables names and types are changed
- Instructions are deleted, modified
Code Clone Detection Applications

1. Detection of semantically identical fragments of code.
3. Detection of semantic mistakes arising during incorrect copy-paste.
Code clone detection approaches and restrictions

**Textual (detects type 1 clones)**

**Lexical (detects type 1,2 clones)**

**Syntactic (detects type 1,2 clones and type 3 with low accuracy)**

**Metrics based (detects type 1,2,3 clones with low accuracy)**

**Semantic (detects type 1,2,3 clones, but has big computational complexity)**
Formulation Of The Problem

Design code clone detection tool for C/C++ languages capable for large projects analysis.

Requirements:

- Semantic based (based on Program Dependence Graph)
- High accuracy
- Scalable (analyze up to million lines of source code)
- Detect clones within number of projects
Architecture

Generate PDGs during compilation time of the project based on LLVM compiler.

Analyze PDGs to detects code clones
Architecture: PDGs’ generation

1. Construction of PDG
2. Optimizations of PDG
3. Serialization of PDG
Example of Program Dependence Graph

C/C++ Code

void foo() {
    int b = 5;
    int a = b*b;
}

LLVM bitcode

define void @foo() #0 {
    %b = alloca i32
    %a = alloca i32
    store i32 5, i32* %b
    %1 = load i32* %b
    %2 = load i32* %b
    %3 = mul nsw i32 %1, %2
    store i32 %3, i32* %a
}

PDG

Edges with blue color are control dependences
Edges with black color are data dependences
Architecture: PDGs’ analyzes

1. Load dumped PDGs
2. Split PDGs to sub graphs
3. Fast checks (check if two graphs are not clones)
4. Maximal isomorphic sub graphs detection (approximate)
5. Filtration
6. Printing
Automatic clones generation for testing: LLVM optimizations

- **C/C++ source code**
- **Unoptimized bitcode**
- **Optimized bitcode**
- **PDG**
- **Compare PDGs to detect clone**

- Generated by clang
- Standard optimization passes of LLVM are applied
Automatic clones generation for testing: PDGs’ merge

List of PDGs for the project

PDG 1 PDG 2  PDG n

Modified list of PDGs

PDG’ 1  PDG’ 2  PDG’ n/2

Check for clone

PDG i  PDG’ j

PDG’ j

PDG i  PDG k
Advantages

1. Compile-time very fast generation of PDGs.
2. No need of extra analysis for dependencies between compilation modules.
3. High accuracy (above 90 %).
4. Scalable to analyze million lines of source code (C/C++).
5. Possibility to detect clones within list of projects.
6. Possibility for parallel run.
7. Opportunity of automatic clones generation for testing.
## Results: comparison of tools

All tests are clones. One original file was modified to obtain all 3 types of clones [1].

<table>
<thead>
<tr>
<th>Test Name</th>
<th>CCFinder(X)</th>
<th>MOSS</th>
<th>CloneDR</th>
<th>CCD</th>
</tr>
</thead>
<tbody>
<tr>
<td>copy00.cpp</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
</tr>
<tr>
<td>copy01.cpp</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
</tr>
<tr>
<td>copy02.cpp</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
</tr>
<tr>
<td>copy03.cpp</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
</tr>
<tr>
<td>copy04.cpp</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
</tr>
<tr>
<td>copy05.cpp</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
</tr>
<tr>
<td>copy06.cpp</td>
<td>no</td>
<td>no</td>
<td>yes</td>
<td>yes</td>
</tr>
<tr>
<td>copy07.cpp</td>
<td>no</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
</tr>
<tr>
<td>copy08.cpp</td>
<td>no</td>
<td>no</td>
<td>no</td>
<td>yes</td>
</tr>
<tr>
<td>copy09.cpp</td>
<td>no</td>
<td>no</td>
<td>yes</td>
<td>yes</td>
</tr>
<tr>
<td>copy10.cpp</td>
<td>no</td>
<td>no</td>
<td>yes</td>
<td>yes</td>
</tr>
<tr>
<td>copy11.cpp</td>
<td>no</td>
<td>no</td>
<td>no</td>
<td>yes</td>
</tr>
<tr>
<td>copy12.cpp</td>
<td>no</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
</tr>
<tr>
<td>copy13.cpp</td>
<td>no</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
</tr>
<tr>
<td>copy14.cpp</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
</tr>
<tr>
<td>copy15.cpp</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
</tr>
</tbody>
</table>

1. Chanchal K. Roy: Comparison and evaluation of code clone detection techniques and tools: A qualitative approach

- **yes** — test was detected as clone with original code.
- **no** — test was not detected

---

### Accuracy

![Accuracy Graph](accuracy_graph.png)

**Accuracy**

- CCFinder(X)
- MOSS
- CloneDR
- CCD

- Accuracy
Results: PDGs’ generation

Intel core i3, 8GB Ram.

Source code lines

PDGs’ generation time

Size of dumped PDG
Results: clones detection

Similarity level higher 95%, minimal clone length 25.
Intel core i3, 8GB Ram.

Clone detection time

Number of detected clones
Results
Thank You.