Codesim

Codesim is a reproduction of Needle: Detecting Code Plagiarism on Student Submissions. It supports performing a similarity check on two single-file C++ programs.

Algorithm

Steps

1. Compile source file with `clang++`;
2. Use `nm` command to list symbols in the object file;
3. Use `objdump` command to read all the function body in the object file;
4. Use Needle algorithm to diff these two sets of functions.

Compile

Codesim will compile input program with the following command.

```
clang++ --std=c++17 -pedantic -O2 {{filename}}
```

And then list symbols in the object file.

```
nm --demangle --defined-only -g -P {{object}}
```

Notice that we will extract all the symbols in the text (code) section, and then filter all the functions whose names start with `std::` and some other special functions like `_start`.

Finally, dump all the compiled functions defined by the user.

```
objdump -d {{object}}
```

Needle

Needle treats a function as a list of instructions (ignore the operands), and define a similarity function between f_1, f_2:

$$\sigma(f_1, f_2) = \max_{k \in \{1,2,...,|f_2|\}} LCS(f_1, f_2[k : k + \omega])$$

This means calculate the max LCS between function f_1 and a window of f_2 with the size of ω.

Then, create a weighted flow network graph like this

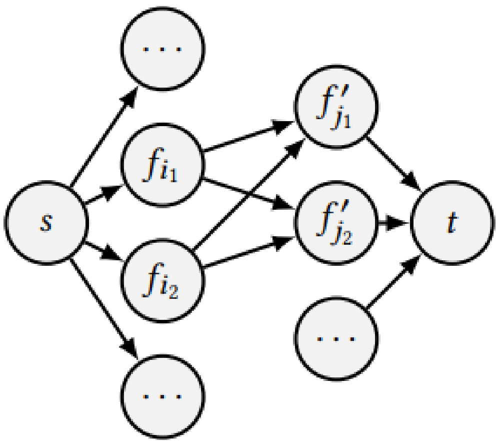

Figure 3: The flow graph G constructed by P_1 and P_2. Each edge (u, v) has capacity $c(u, v)$ and weight $w(u, v)$. The maximum weight network flow from source vertex s to sink vertex t denotes to what extent P_1 can be embedded into P_2.

The maximum cost divided by the total size of the first program is the final result. You can see more details in the original paper.

The differences from the original paper are that we extract the full instructions not ignoring the operands. We use the final the maximum cost divided by the total size multiplied by $1 + \exp\left(-ALPHA + BETA\right)$ (you can enable this by passing argument `--norm`).