171240524 张灵毓
171240524@smail.nju.edu.cn

codesim 实验报告

Version 1
2020 年 11 月 23 日

Overview

本实验采用的方法是通过对待测程序文本处理得到一些 fingerprints，然后对比两个程序的 fingerprints，从而判断它们之间的相似度。具体采用的是和 MOSS 工具相同的方法。其中最主要的有两部分，一是通过词法语法分析识别出待检测代码中的 identifiers(如变量名，函数名，类名等) 然后用统一的符号去替换，从而避免变量名和函数名的更改导致的相似度的降低。二是将待检测源程序文本通过 n-gram 处理并 hash 后，如何从这些 hash value 中选取若干个当作这个程序的 fingerprints，这里采用的是 WINNOWING 算法。

Replace Indentifiers

这一步中的词法分析和语法分析部分使用 libclang 完成，实现的过程可以大致分为三步：

1 通过词法分析识别出被标识为 id 的 tokens, 但这一步我们无法知道每一个 id 具体是函数还是其他类别.

2 通过遍历语法树得到源程序中定义的 id 和其类别一一对应的映射，这一步实际上我们只需要源程序中自己定义的 id，所以我们可以注释掉头文件，这既能够保证我们能找到所有我们需要的 id，也能很大程度缩小语法树的大小。

3 根据前两步的结果将源程序中的 id 替换成相同的符号。

WINNOWING algorithm

在使用 WINNOWING 算法选取 fingerprints 之前，我们需要将已经完成上面一步的源程序处理成不包含空格的字符串，并使用 n-gram 将其处理成一个字符串列表，列表中每个字符串长度都是 n，然后对这些字符串进行 hash 得到 hash value.

为了从这些 hash value 中选择出 fingerprints，WINNOWING 算法所采用的策略是：设置了一个窗口大小为 w，窗口在 hash value 的 sequence 上移动，每次都选择窗口中值最小的 hash value 当作 fingerprints，如果存在多个最小的 hash value, 则选取其中在窗口中最右边的那一个。

这个策略能保证任意连续的 w 个 hash value 中一定会至少选择一个当作 fingerprints，既能够选取足够的 fingerprints，又能够有效的控制 fingerprints 的数量在合适的范围。
