代码相似性检查算法简介

这里采用的方法是基于抽象语法树（AST）来进行代码相似性的检查。
具体而言，从代码源文件中解析出 AST，然后比较两个 AST 之间有多“相似”，将这个指标作为两个代码文件有多相似的指标。

关键实现技术以及一些实现细节

在程序实现中，我们使用了 clang 13.0.0 来生成我们的 AST。
由于 clang 缺少针对 Java 语言的接口，所以在这里采用了比较特殊的处理方式来构建 AST 树。
具体而言，这里使用了 Runtime.getRuntime().exec(String cmd) 接口来使用 Java 语言运行命令行命令。同时使用 clang 的-ast-dump=json 命令行选项使 clang 程序输出的 AST 以 Json 格式输出。
那么接受到了 clang 生成的 json 字符串，现在的问题就是 java 如何解析 json 字符串，这里 java 丰富的第三方库就体现出了优势。这里采用了 Jackson 库[1]作为对 json 字符串的解析工具。这个工具可以将 json 以树的形式解析。
那么现在就开始比较了吗？不，由于 C++程序的编译特性，导致了我们现在获得的 AST 树中大量节点实际上都来自于头文件，而实际上不管测试 cpp 文件如何，在我们这个场景下，头文件对应的内容是高度相似的，所以我们需要从树中排除掉这些节点。只保留来自源文件中的 AST。
这里引入一个假设：所有#include 语句都在源文件开头，我们认为这是合理的——确实又将一部分文件（.inc 文件）以#include 方式插入代码某处，但此时我们认为这是代码的一部分。在 clang 生成的 json 中，其中 loc 项记录了对应节点从何处生成，运用此信息我们可以将原有因头文件而巨大的 json 结构消除很大部分。
另外，json 树中还保留了例如 loc 属性这样与 AST 关系不大的属性，那我们在处理 json 树时也可以将这些信息删去，保留更关键的信息。
那么现在的问题就在于我们如何评估两个 AST 的相似程度，或者说我们如何评估两棵树结构的相似程度。一个简单的想法是，类似于两个字符串，我们可以借由编辑距离与两个字符串的长度和比值来定义两个字符串的不相似程度。我们也可以通过两个树的编辑距离以达成这一目标。
所以在这里评估树编辑距离我采用了 APTED 算法[2][3]，它是 RTED 算法[4]的升级。
为了评估树编辑距离，一个自然的想法是递归的处理。然而这就意味着难以接受的指数级的时间复杂度。APTED 算法采用动态规划，有多项式级的时空复杂度。并且 APTED 提供了一个 Java 第三方库[5]。我们在项目中引用了这个第三方库。
这样，我们有了树编辑距离，我们就可以得到两个 AST 有多不相似，那么，反过来就是两个 AST 有多相似，这个相似值就可以作为两个代码相似度的一个评估值。

一些问题

目前，程序 AST 建立过程中仅使用 clang，这就意味着它只能较好的对抗符号层面的混淆，对控制流层面的混淆仍然是比较不敏感的，此外对于一个典型场景：在代码中加入大量死代码，“稀释”重复代码，其完全无法对抗。因为 clang 实际上没有程序静态分析的能力。所以设想了此工具一个更好的用法：在处理相似代码前，先将源文件优化编译，反编译回 cpp 文件，以进行死代码消除。
但是问题是反编译工具并不存在。那么另一条路是考虑 llvm IR 层面的优化，但同样，llvm IR 到 AST 的逆转换工具并不存在。就是说不如考虑在 llvm IR 层面寻找相似性。但这显然就和目前本项目的想法完全不同了。
总之，尝试在 AST 层面对抗死代码这类可以依靠静态分析解决的问题似乎仍然比较困难。一个可能有效的思路是将代码拆分到更小层次（函数级），将两个文件变为两个文件组来两两对比，寻找高匹配的代码片段。

参考资料

[1]. https://github.com/FasterXML/jackson
[2]. M. Pawlik and N. Augsten. Tree edit distance: Robust and memory- efficient. Information Systems 56. 2016.
[3]. M. Pawlik and N. Augsten. Efficient Computation of the Tree Edit Distance. ACM Transactions on Database Systems (TODS) 40(1). 2015.
[4]. M. Pawlik and N. Augsten. RTED: A Robust Algorithm for the Tree Edit Distance. PVLDB 5(4). 2011.
[5]. https://github.com/DatabaseGroup/apted