程序分析大作业

1 设计目标

实现一个针对 Java 程序的指针分析系统，包括域敏感、过程间分析。

2 项目结构

使用 Eclipse 直接导入即可。lib 中存放程序需要的 SOOT 包。src/core 中有 4 个文件，Test.java 用于测试；AnalysisEntrance.java 是分析的入口，接收需要分析的类；Analysis.java 是分析核心，对单个函数进行分析；AnswerPrinter.java 用于输出结果到 result.txt。

3 设计思路

核心代码在 Analysis.java 中，Analysis 继承了 ForwardFlowAnalysis，采用并操作，是类似于 Anderson 方式的前向数据流分析。采用程序模拟执行的方式，首先从 main 函数入手，每个 Analysis 对象分析单个函数，当遇到函数调用语句，则 new 一个 Analysis 对象，继续进行分析。过程中记录函数调用栈 funcstk，防止递归，避免分析无法结束。

3.1 数据格式

Analysis 中 result 维护了变量的指向情况，类型是 Map<String, Set<String>>，其中 Key 存储局部变量、返回值变量、堆中变量等；Value 是堆中位置的集合。Key 的命名格式：局部变量是“函数.变量名”；返回值变量是“函数.@.return”；堆中变量是“#.内存分配的数字.属性”；形式参数是“函数.@.形参在参数列表中的索引”；this 参数是“函数.@.this”。

3.2 函数调用

Analysis 中 functionCall 处理函数调用语句，接收函数表达式 ie 与当前各变量的指向结果为参数，以 Set<String>作为返回值。若包含 Benchmark 类的 alloc(int)，则记录位置；若包含 test(int, Object)，则保存被测变量到 queries（Map<String, String>类型）中。

若是其他情况。先构造一个 Map<String, Set<String>>类型的 init 集合。其中放入<形式参数，形参对应实参的当前分析结果>、<所有的堆中变量，对应的当前分析结果>，如果是 InstanceInvokeExpr 的话，还要放入<this 参数，调用对象的当前分析结果>。之后将被调函数压栈 funcstk，new 一个 Analysis 对象 pa，构造函数传参有 init 集合，funcstk 等。pa 分析结束。合并 pa.queries 到当前 queries；记录 pa.result 中的“函数.@.return”到 ret 中，用于 functionCall 函数返回值；将 pa.result 的堆中变量、queries 中的变量、返回值变量的分析结果并到当前 result 中；若需要的话，将形参、this 的分析结果并到 result 实参中。

3.3 赋值语句

先分析右侧，再分析左侧，把右侧的并入左侧。若右侧是 ParameterRef、ThisRef 注意记录右侧与左侧的对应关系到 paraMap 中，将形参的分析情况并入 rightSet；若右侧是 NewExpr，allocId 并入 rightSet；若右侧是 Local，Local 的分析结果并入 rightSet；若右侧是 InstanceFieldRef，即 base.field 形式，找到 base 指向的所有位置 entry，将“#.entry.field”的指向并入 rightSet；若右侧是 InvokeExpr，将 functionCall 返回值并入 rightSet。若左侧是 Local，直接将 rightSet 并入 result 的 Local 变量指向结果中；若左侧是 InstanceFieldRef，找到 base 指向的所有位置 entry，将 rightSet 并入“#.entry.field”中。

3.4 返回值语句

只处理有返回值的。将<“函数.@.return”，返回的 Local 的指向结果>并入 result 中。

4 成员分工

云昊（201828015059015）
李为（201828015070005）
牛海行（201828015059031）