课程项目一 JAVA 指针分析系统

一、算法主要设计思想

程序结构大致分为两层，whole program transformer 中主要负责从程序及据其生成的 Jimpe 代码中添加 new 操作和 BenchmarkN 中 alloc 分配的 id 的映射，以及分析程序中执行语句的种类，并分别将其转换成 Anderson 算法中的约束或者 CFL 设计中 new、assign、points-to 等带权的边，并存储下来。

Anderson 类或 CFL 类中，尤其是在其 run 方法中，负责将上一层中获得约束或带权边进行进一步分析，获得指向分析的结果。并把结果通过 Answer Printer 输出到 result.txt 文件中。

二、第一次迭代

1、总体设计：Anderson 基于 Filed-Based 分析

2、算法设计：assign constraint 列表、new constraint 列表，run 方法

3、可执行语句种类扩充：parameters、return value、this Ref

4、新增功能：

在 Anderson 分析方法的基础上，把所有对象的特定域当成一个对象，做出基于 Field-Based 分析，增加类只有一个对象实例时的分析精度。

三、第二次迭代

1、总体设计：基于 CFL 可达性的域敏感分析

2、算法设计：

整体采用前向星的数据结构存储各个节点之间的边关系。在 class Edge Type 中，枚举出 Assign、New、Put、Get、Alias、Flow To、Point To 等边种类，针对 Put、Get 种类的边声明 Object f 存储其操作对象；

connect With 方法负责运算节点之间的关系，并存储至 Graph 类中，包括上游程序分析的节点及其边的种类，CFL 类中提供 whole program transformer 所需的接口，包括添加节点、边、进行基于 CFL 的指针分析等。

3、可执行语句种类扩充：

static Field Ref、virtual Invoke、calling virtual Method、Interface Invoke

4、新增功能：

Feat1：保证 sound，添加 try-catch 和 multi-test；

Feat2：基于一次克隆的上下文敏感分析；

Feat3：增加对继承和多态方法调用的分析精度。寻找合适的 api 去分析方法调用所在类的基类，并将其加入考虑的范畴内，进行一个 sound 的分析，增强程序在多态调用情况下的分析精度；

Feat4：考虑到存在 new 一个数组的情况，程序将数组中所有的对象的访问等同于类中对象的特定域，同样将其中引入 assign、put、get 等边，进行一个 CFL 的分析，保证面对数组分析时的精度和安全性。

四、小组成员及分工

单曦增 2000012990 代码框架构建，算法整体设计及代码实现，边界情况调试

张杰然 1900012159 边界情况调试，测试样例设计，项目报告撰写