

《自动化测试 2020》经典自动化测试大作业

一、项目背景

测试选择是一项常见的回归测试优化手段。而程序分析作为认识程序的常用技术，被广泛应用于测试选择技术的自动化。根据程序分析的手段不同，测试选择可以分为静态选择和动态选择两种。本项目要求同学实现一个静态测试选择工具，能够根据代码变更自动选取出受变更影响的测试（Impacted Tests）。完整的静态测试选择流程大致可以分为三个主要阶段：

1. **变更检测阶段：**以某项指标（文件的检验和或依赖关系）为基准，判定哪些生产代码产生了变更。一般来说，基准发生变化的生产代码就是产生变更的部分；
2. **静态分析阶段：**使用静态分析工具构建生产代码与测试代码之间的联系，得到代码依赖图；
3. **测试选择阶段：**根据代码关系图和变更信息，筛选出受变更的测试。

二、项目要求

1. 总体要求

为了降低作业的复杂度，第 1 阶段（变更检测）**不要求**同学实现。变更信息作为输入给出；选择本项目的同学需要按照实现要求，完成一个可以完成第 2 阶段和第 3 阶段的简易测试选择工具。

2. 实现要求

（1）项目使用 java 语言实现，采用 Maven 进行构建；

（2）静态程序分析采用静态程序分析框架 WALA（ $\leq 1.5.5$ ）实现，WALA 相关依赖可以在 maven 中心库找到（<https://mvnrepository.com/search?q=wala>）；

（3）实现**类**和**方法**两种粒度的依赖分析；

提示：测试选择的粒度其实和静态分析阶段生成的代码关系图的粒度有关。同学们可以利用 WALA 框架提供的 ClassHierarchy 生成 CHACallGraph，再进一步生成类依赖图；方法依赖图则可以通过处理 O-CFA 调用图进一步得出；

（4）生成可以运行的 jar 包。jar 包命名为 testSelection.jar，基本使用方法如下：

① 指令一：java -jar testSelection.jar -c <project_target> <change_info>，执行类级测试选择；

② 指令二：java -jar testSelection.jar -m <project_target> <change_info>。执行方法级测试选择。

（5）针对（4）的几条说明

① <project_target>是一个字符串，表示一个文件路径，指向待测项目的 target 文件

夹（本次大作业采用的待测项目全部采用 maven 构建）。测试选择工具以一个待测项目和一个记录了变更信息的文本文件为输入。在接收到 target 文件夹路径后，测试选择工具应该能够自动读取文件目录下所有的.class 的文件，并把它们添加到 AnalysisScope 中；

② <change_info>是一个字符串，表示文件路径，指向记录了变更信息的文本文件。该文本文件每行一个方法签名，表示生产代码中发生变更的代码。

3. 数据约定

（1）**输入约定：**本项目提供两组输入数据，每组数据包含 5 个符合 maven 结构的待测项目，每个待测项目内含 1 个变更文件（change_info.txt）。其中：

① 第一组数据：学生可见，用于调试项目、保证项目功能的正确性；

② 第二组数据：学生不可见，用于评分。

③ 变更文件内含变更信息，每行一个方法签名，表示发生变更的生产代码；

（2）**输出约定：**工具应该能够将测试选取结果记录到文本文件中，并输出到 jar 包所在的目录下（即“./”）。输出文件每行一个方法签名，表示一个受测试影响的方法。指令一和指令二的输出应该分别命名为 selection-class.txt 和 selection-method.txt。

三、 提交要求

1. 本项大作业分数分配：代码依赖图 10%，代码 60%，jar 包运行情况 30%

2. 整个项目提交到 Github 上，请同学们之后提供仓库的 URL 供我们 Fork。相关格式和目录结构参见：<https://github.com/yushengcheng/AutomatedTesting2020>

3. 目录结构说明：

（1）可运行 jar 包命名为 testSelection.jar，并放在 Demo/目录下。请同学们在自己的电脑上，针对自己的 jdk 路径先生成一个 jar 包，用于调试和运行。再将 wala.properties 中的 java_runtime_dir 属性值修改为 C:/Program Files/Java/jdk1.8.0_231 后生成一个 jar 包，并作为最终作业提交。配置详情见“WALA 操作指南.pdf”；

（2）针对第一组数据生成 10 个代码依赖图（PDF 格式），放置在 Report/目录下。每个项目对应 2 个代码依赖图，分别表示类级和方法级的代码依赖。代码依赖图命名格式为：“class-项目名.pdf”和“method-项目名.pdf”；

提示：可以将转换成.dot 文件的格式并输出，在使用 graphviz 将其转化成.pdf 的形式。该任务不要求自动化，可以编写代码生成.dot 文件后通过命令行生成依赖图。

Graphviz 官网：<https://graphviz.org/>

（3）项目代码放置在 Project/目录下，要求代码逻辑清晰、注释完整。

四、 未尽事宜发邮件咨询

赵 源 zhaoyuan@smail.nju.edu.cn

张犬俊 quanjun.zhang@smail.nju.edu.cn

钱瑞祥 qrx@smail.nju.edu.cn