高级程序设计II 大作业实验报告

项目地址：GitHub

代码贡献

- 农钧翔 2019201407：
  - 实现迭代数据流算法 & 构建`Flatten`图（Task2）
  - 生成关联规则及置信度（Task3）
  - 库函数对接与格式化输出（Task4）
- 于倬浩 2019201409：
  - 构建`Transaction`图 & 规范化指令（Task2）
  - 实现`Apriori`算法生成项集（Task3）
  - 图形化界面的实现 & 与动态链接库的输出对接（Task5）
- 汪元森 2019201420：
  - 初始化数据流算法（Task2）
  - 实现`calcu_support`计算项集的支持度（Task3）
  - 缺陷检测算法的实现（Task4）

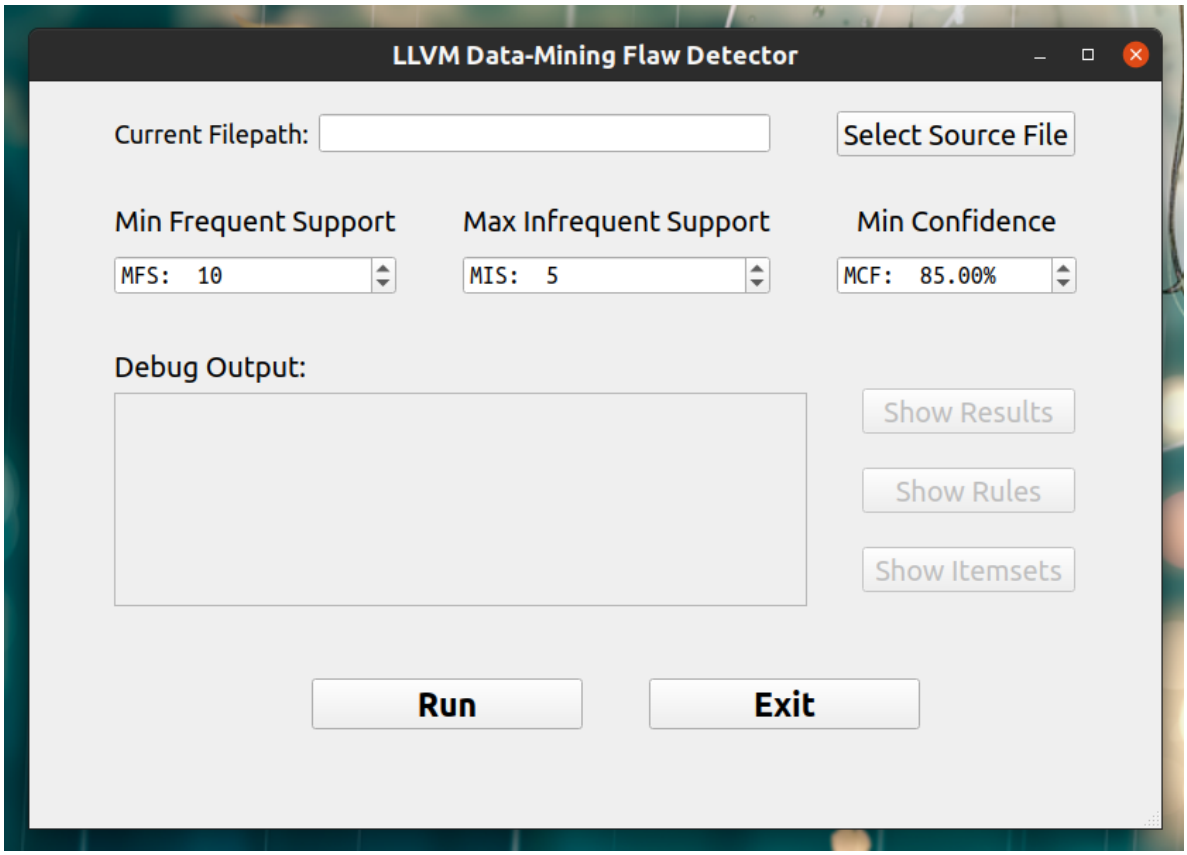
实验环境

操作系统：`Ubuntu 20.04 LTS`

```
$ clang --version
clang version 10.0.0
Target: x86_64-pc-linux-gnu
Thread model: posix
InstalledDir: /usr/bin
$ qmake --version
QMake version 3.1
Using Qt version 5.12.8 in /usr/lib/x86_64-linux-gnu
```

简单介绍

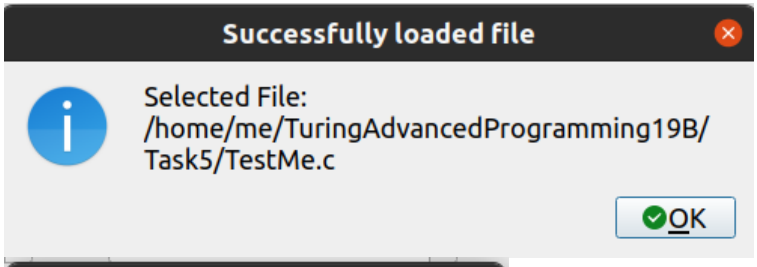
本次UI的实现采用了QT5，实现了一个简单的多窗口程序，主窗口初始状况下如下图所示：

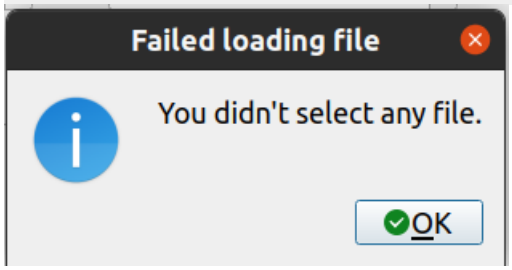


支持指定需要检测的源文件位置，指定MFS、MIS、MCF的值。

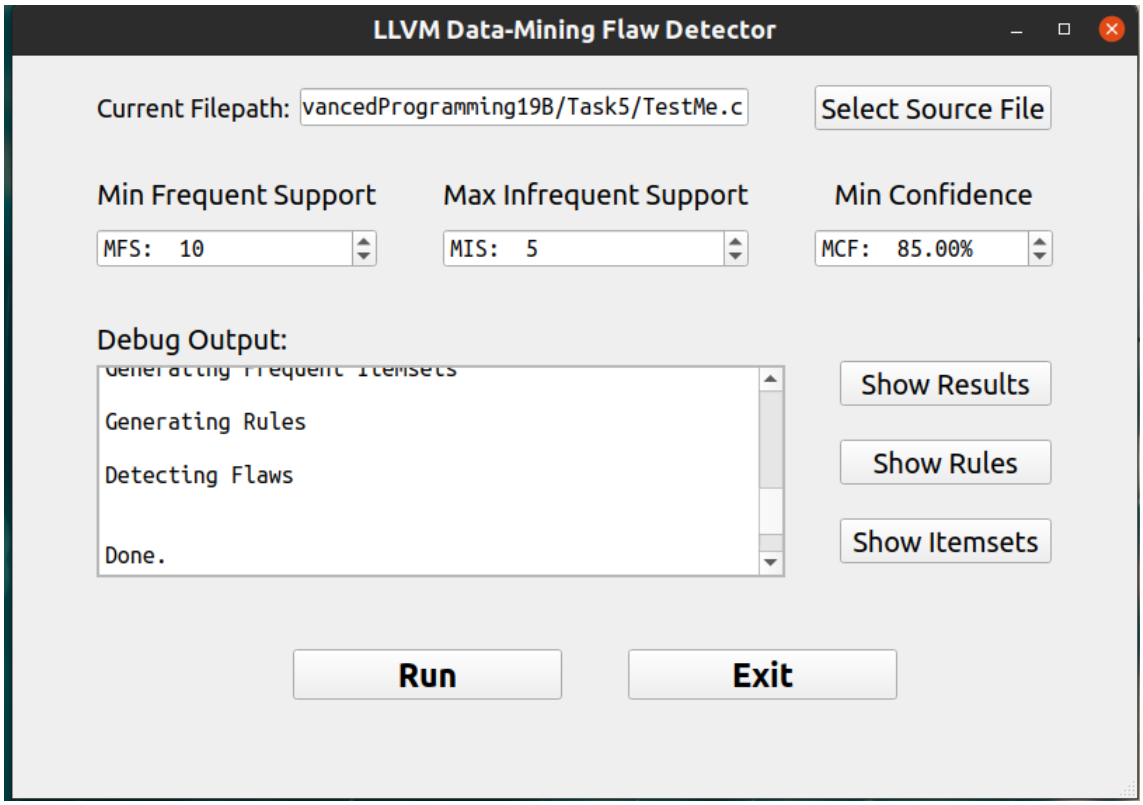
对于三个指定值的输入，选用了spinbox，以便控制输入值的上下界、步长等。

选择文件的PushButton通过调用QFileDialog，调用系统选择文件的窗口，较为美观，且限定用户只能选择C/C++源代码文件。文件选择结束后，调用QMessageBox给出反馈，同时将文件目录更新至主窗口的lineEdit中。





选定源文件，指定参数后，单击"Run"即可运行监测。运行过程中，所有设定参数的按钮均被禁用，运行结束后重新启用这些按钮。



动态链接库MyPass.so会将调试信息输出到stderr中，并实时更新至"Debug Output"中。对于结果和项集的传递，MyPass.so将这些信息输出至stdout，使用不同的token区分输出类型，MyPass在成功执行结束后，在stdout中返回一个token"$$SUCCESS"，UI在读取到这个token后，才会启用右侧的三个按钮。可以保证在编译错误/运行错误/MyPass运行结束前，阻止用户点击右侧的按钮。同时在文件更新后，亦会禁用展示结果的按钮。

在实现上，我使用了两个不同的槽函数，对接动态链接库的两个输出流，需要注意的是，由于输出流自带缓冲区，在每一次输出后必须清空缓冲区，否则会导致部分字符串丢失。

单击"Show ..."，会打开一个新窗口，展示项集数据/生成的规则/缺陷检测的结果。对于项集和规则，使用字符串列表的形式展示，每行一个规则或项集，使用等宽字体展示每条规则。对于缺陷检测结果，以只读纯文本的形式展示，灵活性更好。在实现上，我在MyPass.so的标准输出中，加入了不同的token，表示不同类

型的数据，槽函数读取时，便可较为容易的将输出的纯文本数据分类处理。下面同时展示了三个数据窗口的内容：