

## Lab2 说明

软件分析技术 2025 秋 教学团队

截止时间: 2025/12/23 23:59:59

Lab2 是单人完成的作业, 使用 Github Classroom 下发文件、自动测评, 在平台上的测试时间限制可以参考仓库中的 classroom.yml 文件, 请加入 Classroom 并将身份绑定为自己的学号, 请保持 repo 为 private, Classroom 地址为: <https://classroom.github.com/a/NBXav-fq>

(原地址 <https://classroom.github.com/a/D6anuCjC> 已弃用, 请同学们重新加入新的 classroom, 已经提交的代码我会私发给同学本人). 请注意: 只有提交到 main 分支上的代码才会进行有效评分.

在开始 Lab2 之前, 需要安装所需环境, 建议使用 anaconda/miniconda 来管理 Python 环境. 推荐的 Python 版本为 3.10. 我们在根目录提供了一个 requirements.txt 文件, 包含了 Lab2 所需的 Python 包. 你可以在 Lab2 的根目录运行以下命令安装所需的包:

```
pip install -r requirements.txt
```

在安装完成后, 请按照 MIT 6.566 lab3 (<https://css.csail.mit.edu/6.858/2024/labs/lab3.html>) 的 Getting started 部分, 检查环境是否配置正确.

我们为 Lab2 提供了评分脚本, 在运行评分脚本之前, 需要在根目录创建名为 tmp 的空目录. 在根目录运行以下命令可以查看除了隐藏测试和需要助教评分的部分以外的得分:

```
make check-lab3
```

在 singly\_linked\_list 目录运行以下命令可以查看任务 2 的得分详情:

```
python3 test_sym_singly_linked.py
```

### 任务 1.

1. 完成 MIT 6.566 lab3 的第 2 到第 9 个 Exercise. 要求见 MIT 6.566 lab3 说明.
2. 在 MIT 6.566 lab3 的第 9 个 Exercise 中, 已经对若干个数据库不变式进行了检查. 请你自行设计并实现一个新的数据库不变式检查, 并在提交代码的同时提交 pdf 文档, 说明你设计和实现思路.

### 评分细则.

- 占 Lab2 分数的 65%. 在这 65% 中, 完成 MIT 6.566 lab3 的第 2 到第 9 个 Exercise 占 60%, 自行设计并实现一个新的数据库不变式检查占 5%.
- MIT 6.566 lab3 的第 2 到第 9 个 Exercise 的得分由评分脚本自动计算. 评分脚本随 Lab2 一起下发.
- 设计并实现一个新不变式的得分由助教评判.

&lt;

### 任务 2.

1. 实现对 linked list 的 concolic testing.

在评分函数中, 会以 `concolic_execs_with_list(teststaff_sum, maxiter=MAXITER, verbose=1, max_list_length=MAXLEN)` 的方式调用你实现的 `concolic_execs_with_list` 函数, 其中 `teststaff_sum` 是一个对 linked list 进行操作的测试函数. 在 `teststaff_sum` 中, 会使用 `mk_list` 来创建表示 linked list 的符号值, 用来代替原本作为函数输入的具体 linked list. 详细信息请阅读相关代码.

你需要实现 `mk_list` 函数和 `concolic_execs_with_list` 函数. 使得符号执行能够遍历测试函数的所有可能分支. 请不要修改用于测试的文件.

在下发文件中, 已经包含了 `mk_list` 和 `concolic_execs_with_list` 的部分实现, 但是此实现无法通过测试, 你可以在此基础上修改, 也可以完全重写.

2. 保证 concolic testing 过程中产生的 linked list 结构上是合法的.

预先定义的 `SafeSinglyLinkedList` 类, 保证了在满足 Python 程序编写惯例的情况下 (将以下划线开头的属性 “视为” 私有属性), 不会有环、不同的 `SafeSinglyLinkedList` 对象之间不会有共享节点.

`teststaff_construct_correct` 和 `teststaff_cycle_trap` 这两个测试可以检查你实现的 concolic testing 是否会产生不合法的 linked list 结构. 请阅读这两个测试的代码, 确保你的实现能够通过这两个测试.

3. 在此任务中, 存在部分隐藏测试.

◀

评分细则.

- 占 Lab2 分数的 35%. 在这 35% 中, 公开测试点占 15%, 隐藏测试点占 20%.
- 分数由评分脚本自动计算, 公开测试点的评分脚本随 Lab2 一起下发.

▷