

软件分析测试第一次作业

1. 研究问题

构造某一语言程序(本作业针对 java 语言)AST, 计算任意两棵抽象语法树的相似度, 并以此为基础, 分析某一项目特定模块的 work flow 之中的 AST 变化情况, 具体来说, 包括以下内容:

a) RQ1.连续的 Commit 间, 模块 AST 的变化可能有多大?

b) RQ2.连续的发布版本间, 模块 AST 的变化可能有多大?

Bug-fix 版本间, 模块 AST 的变化可能有多大?

功能版本间, 模块 AST 的变化可能有多大?

c) RQ3.在什么情况下, 相近的两次 Commit 会使得模块 AST 发生大的变化?

d) RQ4.开发过程中, 模块 AST 有没有可能发生连续的大的变化?

2. 评估准则

a. 调整同类型节点代码顺序应当不影响 AST 相似度

b. 改变字面量具体的值应当不影响 AST 相似度

.....

3. 实验对象

实验对象来自于开源项目: emoji-java

项目地址: <https://github.com/vdurmont/emoji-java>

Commit 历史概览

```
75811ba - Release v4.0.0
99f4681 - Add source code string for emojis
98dd599 - Fix few emojis with missing/out-of-order codepoints (#88)
18335af - Updated the Json Library version in the dependency list. (#81)
07d8df2 - update readme with full list of emoticons (#86)
163553b - Release v3.3.0
b41cdfa - Fix Travis CI config
95c2eca - Add family_man_woman_girl_boy emoji (#77)
c6cb218 - Freva/detect emoji with fitzpatrick modifier (#76)
b1e44dd - Fixed some emoji has wrong fitzpatrick flag set. (#68)
7f74d36 - method added to extract emojis from text (#67)
d75b072 - Fix several duplicate aliases + add test so we don't add anymore duplicates
340811e - Release v3.2.0
91281b3 - Fix #14: add an isOnlyEmojis method
107534d - Fix #42: missing fitzpatrick modifiers
a5124f9 - Fix #48: couple_kiss_woman_woman was invalid
9d1ff59 - Fix #59: nail polish should support fitzpatrick
3e36a68 - Fix #58: duplicate alias for squared_ng and nigeria flag
07c8c02 - Fix #62: Add emojis from Apple iOS 10.2
ca4a06e - Added slight_smile as variation of slightly_smiling (#57)
```

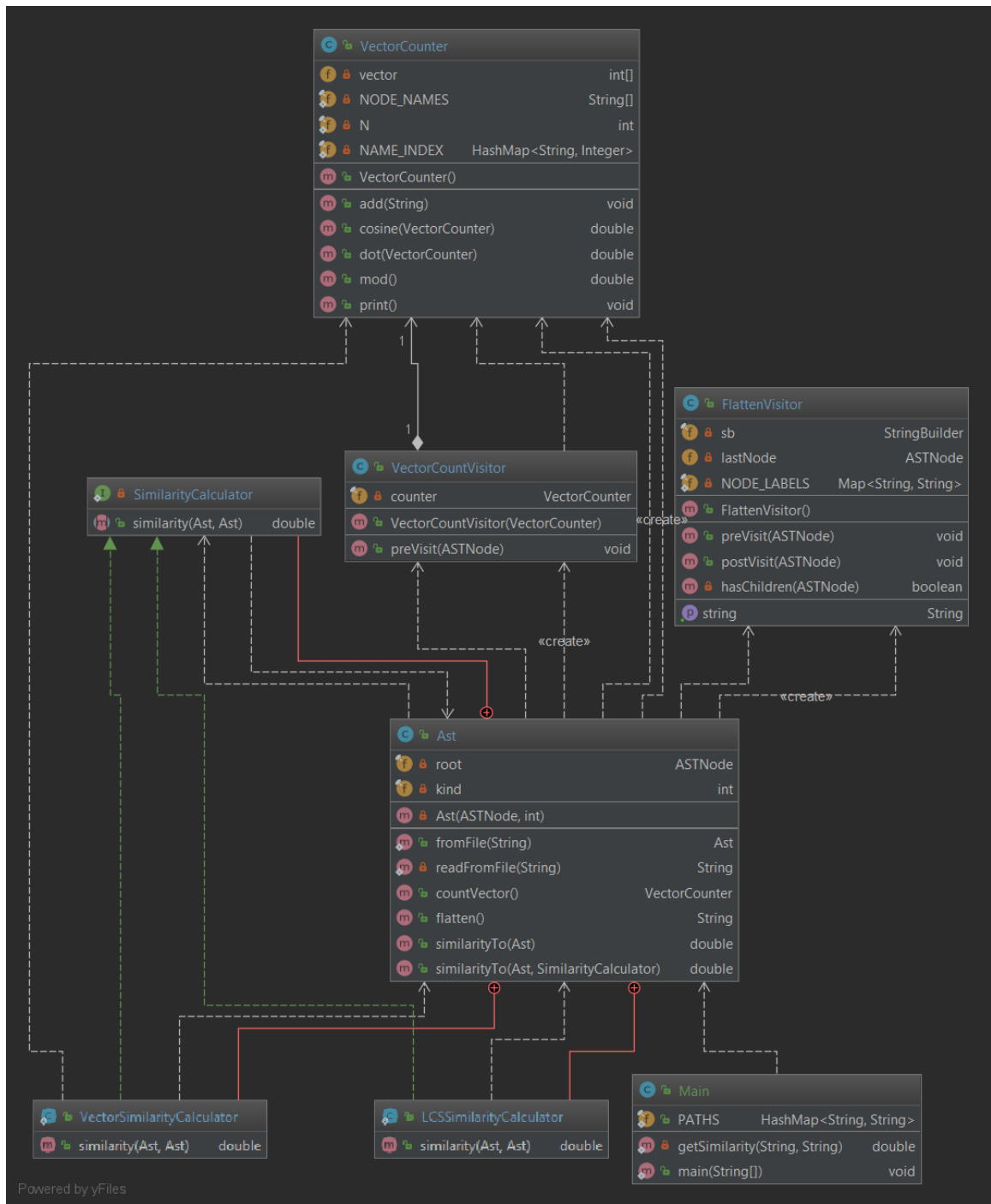
版本历史概览

```
v1.0.0
v1.0.1
v1.1.0
v1.1.1
v2.0.0
v2.1.0
v2.2.0
v2.2.1
v3.0.0
v3.1.0
v3.1.1
v3.1.2
v3.1.3
v3.2.0
v3.3.0
v4.0.0
```

4. 实验步骤

本实验采用 Java 作为主要的开发语言

程序 UML 图如下：



a) AST 构建

AST 的构建工作主要借助于

`org.eclipse.jdt.core.dom.AST`

`org.eclipse.jdt.core.dom.ASTNode`

`org.eclipse.jdt.core.dom.ASTParser`

来完成

b) AST 之间相似度算法设计

将 AST 转化为字符串形式，之后利用 LCS 算法计算其相似度

$$\text{Similarity} = \text{Length}(\text{LCS}) / \text{MAX}(\text{Length}(\text{AST1}), \text{Length}(\text{AST2}))$$

模块 AST 间相似度计算方法：

以文件之间 AST 相似度计算为基础，每个文件的权重为 1 (因为我们不能确定各个文件之间的重要程度，所以假设为 1)，取平均：

$$\text{ModuleSimilarity} = \sum(\text{Similarity}(\text{file})) / \text{Count}(\text{files})$$

c) 连续 commit 间，文件及模块 AST 变化计算

简单来说，通过 git log，git checkout 以及相关 shell 命令完成不同 commit id 之间的模块内单个代码文本文件之间的比较，而后汇总模块内文件相似度，计算连续 commit 之间模块级别的相似度。

d) 连续发布版本间，文件及模块 AST 变化计算

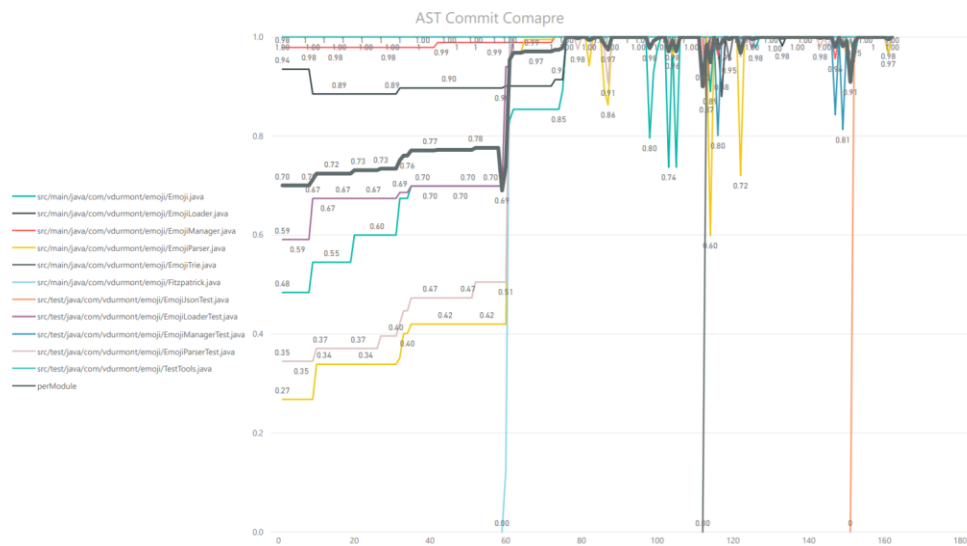
同 c 中的操作，切换不同的版本，而后再具体得到所需结果。

e) 结果描述与分析

见第 5 小节

5. 结果呈现与分析

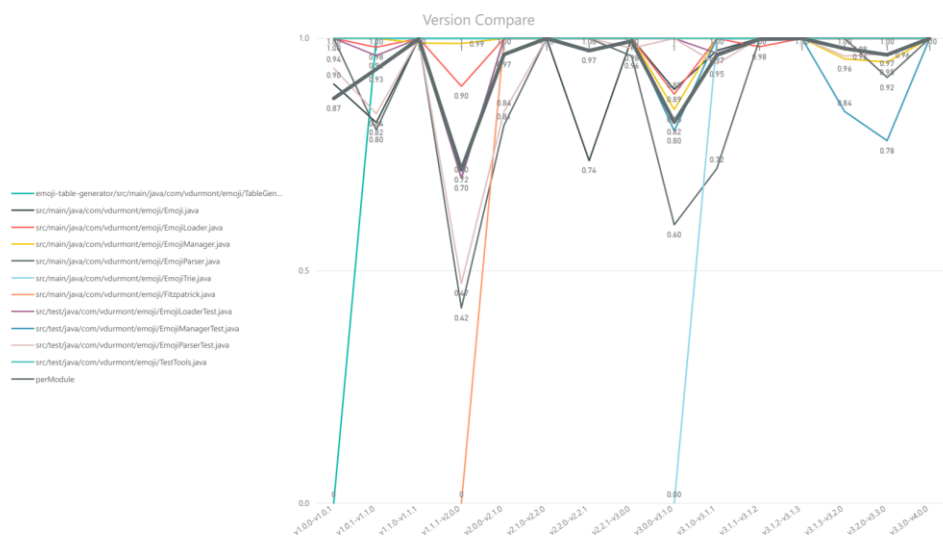
a) RQ1.连续的 Commit 间，模块 AST 的变化可能有多大？



连续的 Commit 间，模块 AST 的变化

图 5-1

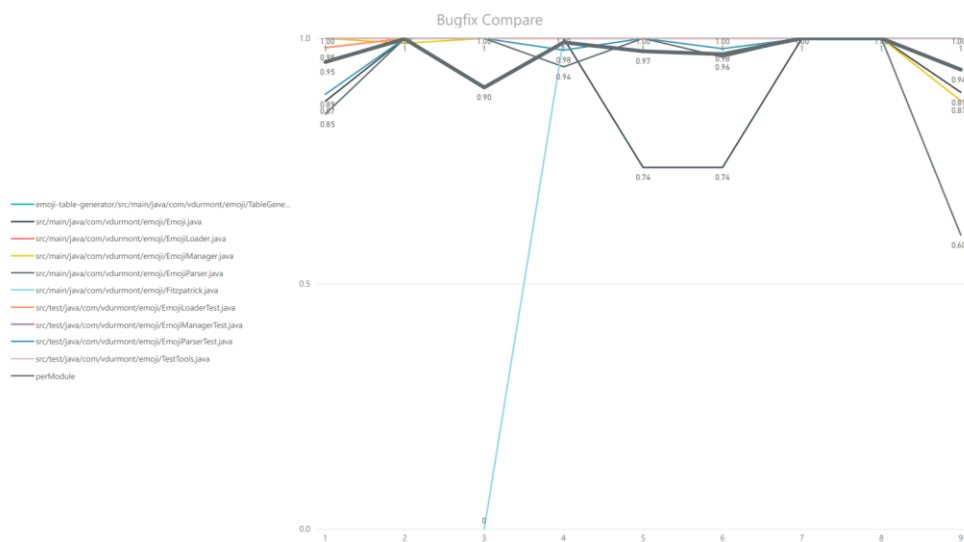
b) RQ2.a 连续的发布版本间，模块 AST 的变化可能有多大？



连续的发布版本间，模块 AST 的变化

图 5-2

c) RQ2.b bugfix 之间，模块 AST 的变化可能有多大？



Bug-fix, 模块 AST 的变化

图 5-3

d) RQ3.在什么情况下，相近的两次 Commit 会使得模块 AST 发生大的变化？

由图 5-1 可知，早期开发阶段，模块 AST 变化较大，初步推断是因为早期模块尚处于非稳定阶段，功能增减和 bug 修复比较频繁，另外受本实验模块 similarity 计算策略的影响，当在连续两次 commit 之间有代码文件新建时，模块 AST 变化较大。

e) RQ4.开发过程中，模块 AST 有没有可能发生连续的大的变化？

由我们的实验结果即图 5-1 可知，在模块开发早期阶段，有可能。

6. 小结

通过本次实验，了解了 AST 的具体构建过程以及对如何计算 AST 之间相似度有了一定的认识。

REF:

<https://github.com/nettee/ASTComparator>