**BEN**

1. **发布时间：2015**

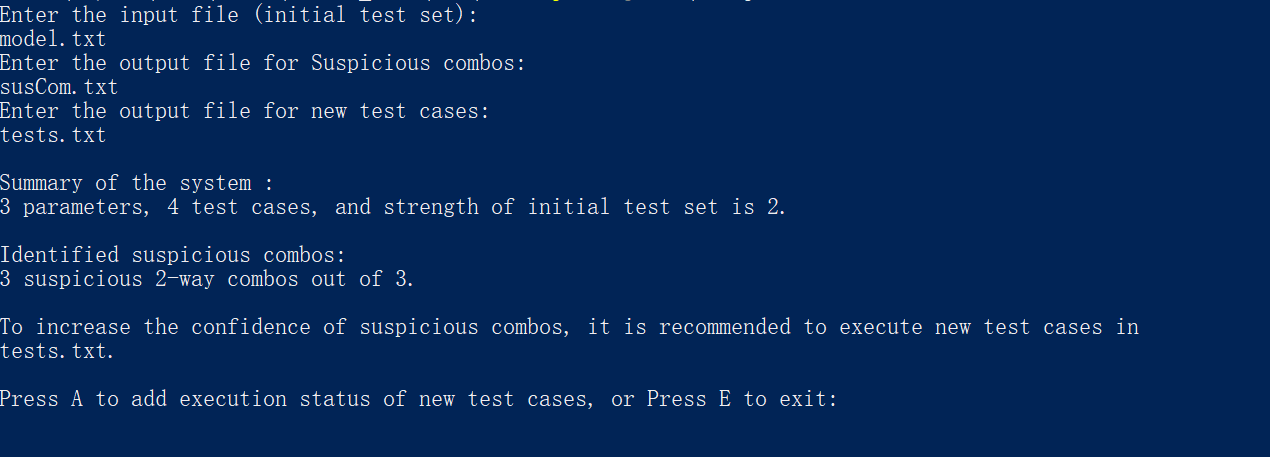
文章BEN: A Combinatorial Testing-Based Fault Localization Tool (2015)提出BEN工具

1. **工具使用形式和跨平台特性**

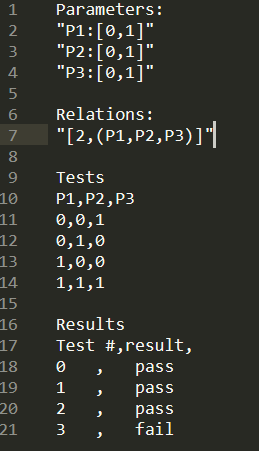
Desktop (GUI) & Desktop (CLI)

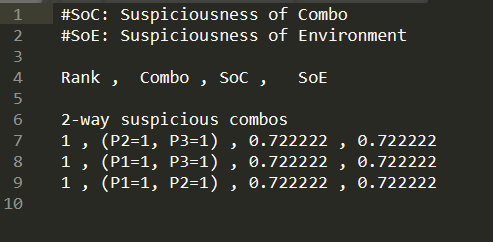
跨平台 (Java)

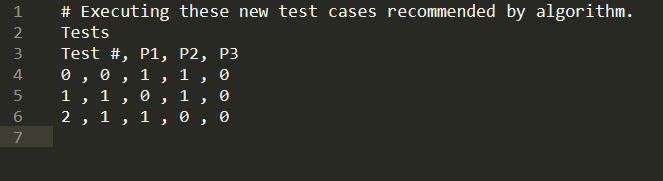
1. **测试生成及其输入输出格式**
   1. **生成算法：**
   2. **支持维度：**
   3. **可变力度：**
   4. **种子：**
   5. **约束：**
   6. **建模语言：**
   7. **输出格式（包括抽象测试用例、自然语言描述测试计划、代码片段等）：**
2. **其它测试活动**
   1. **测试用例优化：**No
   2. **测试执行：**No
   3. **覆盖率评估：**No
   4. **故障定位：**Yes



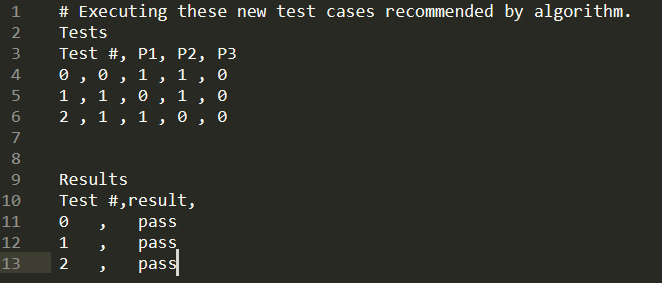
组合测试模型和测试用例输入文件、可疑组合输出文件、下一步测试用例集输出文件如下：

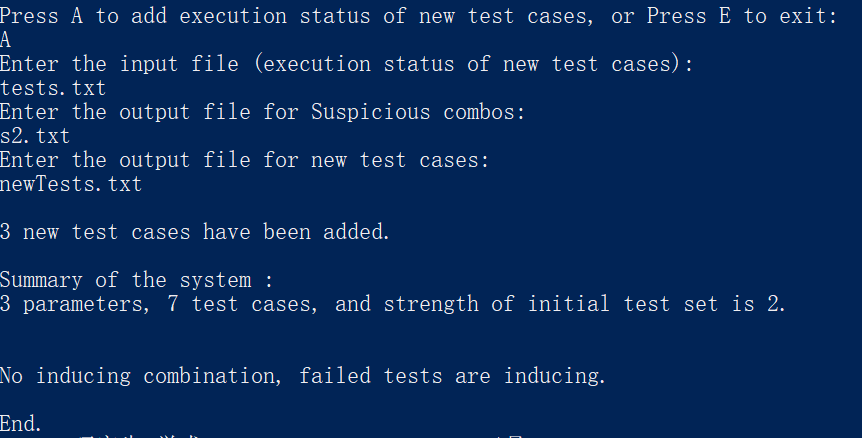




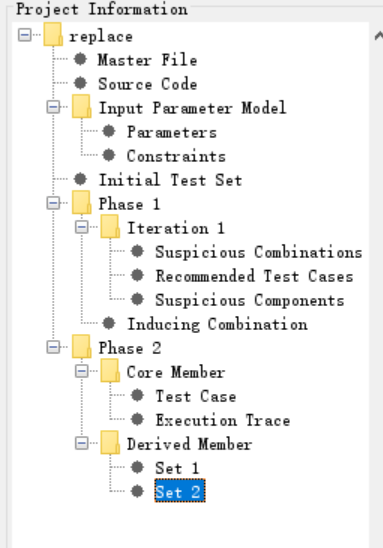


可以执行推荐的测试用例集，继续故障定位。例如下图是三条测试用例的执行结果，全部通过





图形界面版本借助了工具网站提供的replace示例(http://sir.csc.ncsu.edu/portal/index.php)，创建项目时导入模型文件和源代码。模型文件见”工具/inputfile.csv”，给出参数集合、每个参数的取值集合、参数交互关系、约束、测试用例集和测试用例的执行结果。图形界面工具故障定位可分为两个阶段，第一阶段和命令行工具类似，首先计算可疑组合，给出候选测试用例，然后为候选测试用例确定执行结果，确认引发故障的组合，如果不能确认则更新可疑组合，继续迭代。图形界面工具第二阶段定位触发故障的源码语句，该阶段有两个测试用例集，一是包含故障元组的测试用例集，二是和第一部分测试用例集相似，但是未触发故障的测试用例。用户提供这些测试用例集的执行路径，对于那些只出现在故障测试用例中的代码语句，认为可能触发故障，同时出现在故障和非故障测试用例中的语句可疑度稍微低一些；仅出现在未触发故障测试用例中的语句可疑度为0。工具给出代码语句的可疑程度排序，测试人员可根据可疑程度由高到低排查。但是需要提供测试用例的执行路径，工具网站没有给出实例，可疑语句排序功能未使用。



1. **额外功能**
2. **和其它测试工具的兼容性：**No
3. **工具演化：**

No