

# 回归测试工具-KTCase

## 回归测试现状:

回归测试是软件生存周期中十分重要,但又耗时费力的一个阶段,由于缺乏适用、有效的回归测试工具,使得回归测试质量难以保证,进度和费用难以控制,对软件质量造成了很大影响,如何设计既能对软件进行充分测试又能减少经费开支的回归测试用例是测试人员追求的目标。

当前主流的回归测试工具是 WINRUNNER (简称 WR),采用录制/回放技术,通过录制鼠标动作和键盘输入,建立起自己的测试脚本,在回归测试时,通过在新版应用程序中回放测试脚本,对软件进行测试,在回放时无需测试人员介入,因此能够减轻测试人员的工作量,缩短测试时间。但是,如果软件中采用了非标准的控件,将导致测试脚本在回放时经常出错,需要测试人员反复调试脚本,无疑又增加了测试人员的工作量,同时,对测试人员的技术要求也比较高,使得工具的应用效果大打折扣。

另外一种回归测试策略是采用测试用例选择技术,通过不同的方法,从已有测试用例集中选择部分测试用例进行回归测试。KTCASE 回归测试用例设计工具就是其中之一。KTCASE 回归测试用例设计工具能够根据软件修改情况,通过软件变更影响分析,找出因软件变更可能受到影响的部分,并从原有的测试用例中挑选出能够覆盖受影响部分的测试用例,有效解决回归测试中的无效测试用例剔除、冗余测试用例剔除、测试用例遗漏等难题。

KTCASE 有以下功能:

软件功能图自动生成;

软件功能变更标识;

软件功能变更影响分析;

程序流程图自动生成;

程序代码变更标识;

程序变更影响分析;

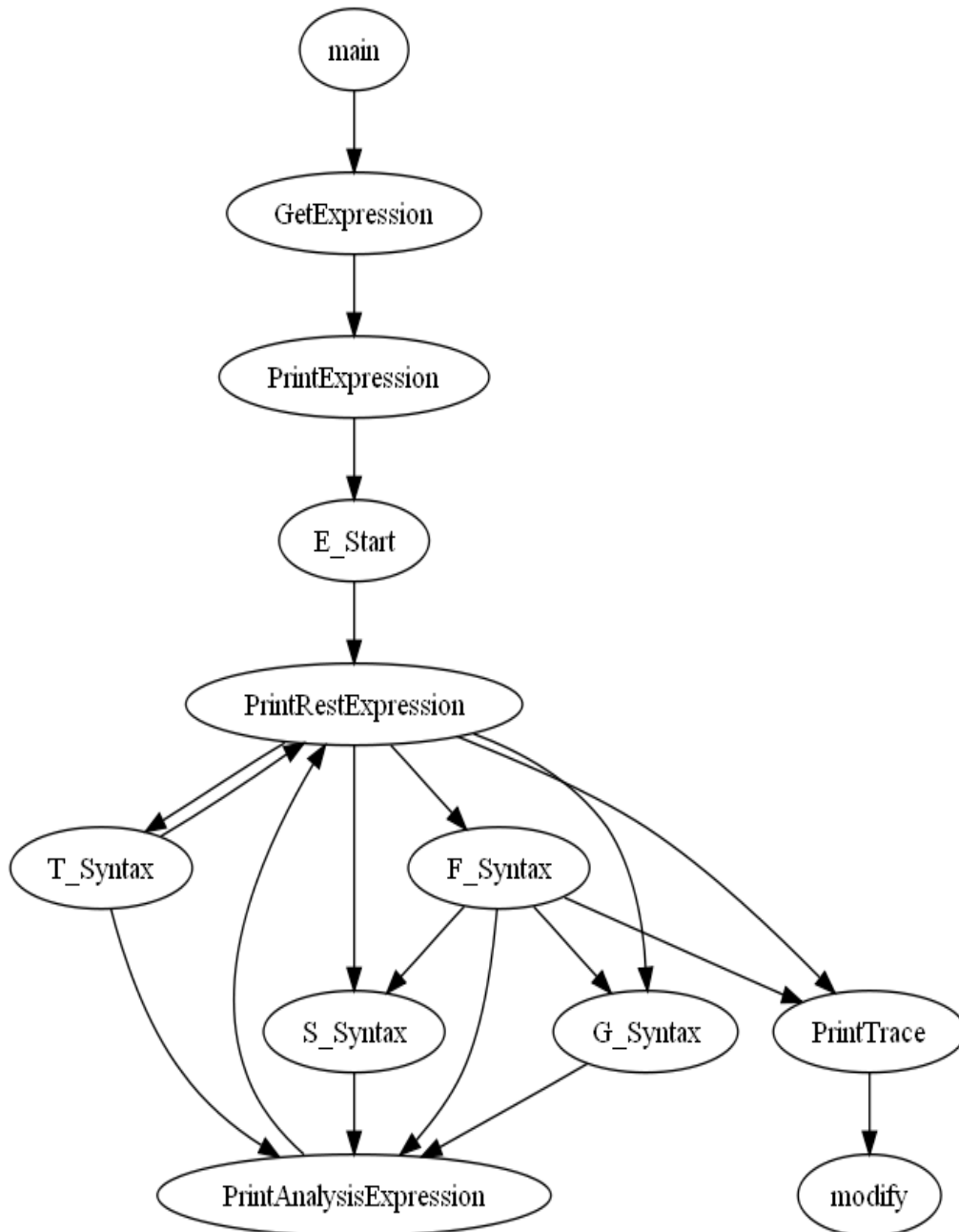
测试用例优化;

回归测试用例选择。

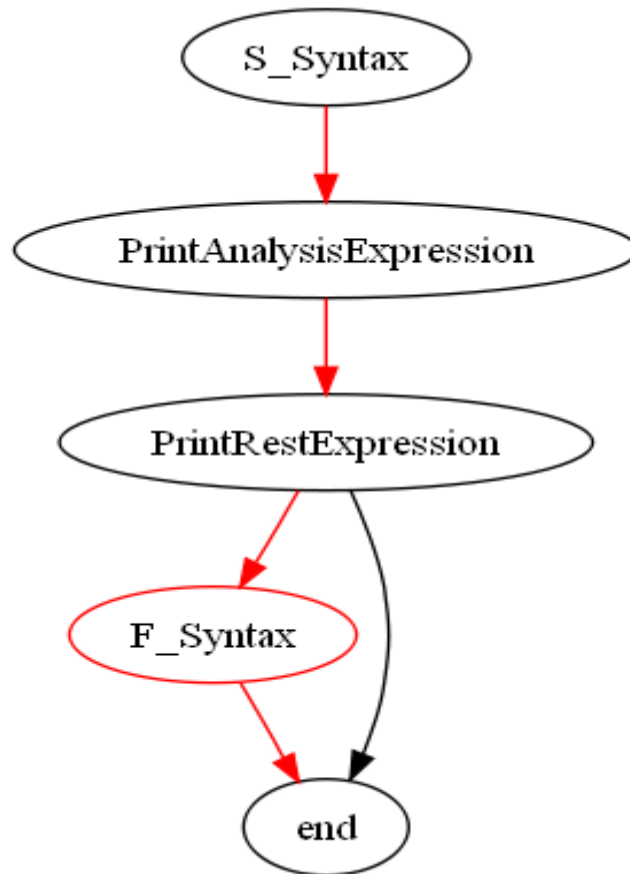


## KTCase 主要功能:

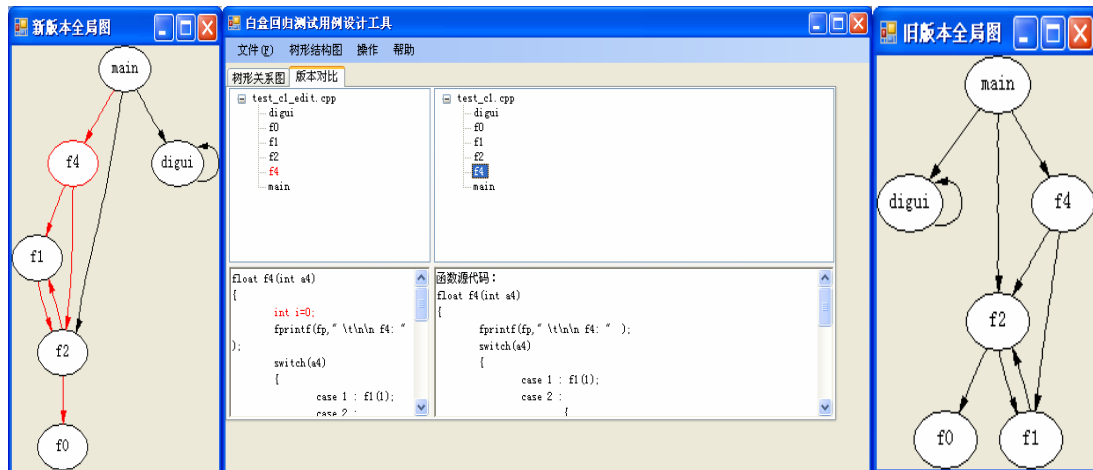
1. 程序流程图自动生成: 通过分析源程序代码, 生成程序流程图;



2. 程序代码变更标识: 通过比较软件修改前后的功能图, 从源程序代码和程序流程图两个不同的侧面标识出程序的变更部分 (用红线标出变更);



3. 程序代码变更标识。通过比较程序修改前后的流程图，找出软件变更所对应的结点和边以及源程序代码（图中红色部分）



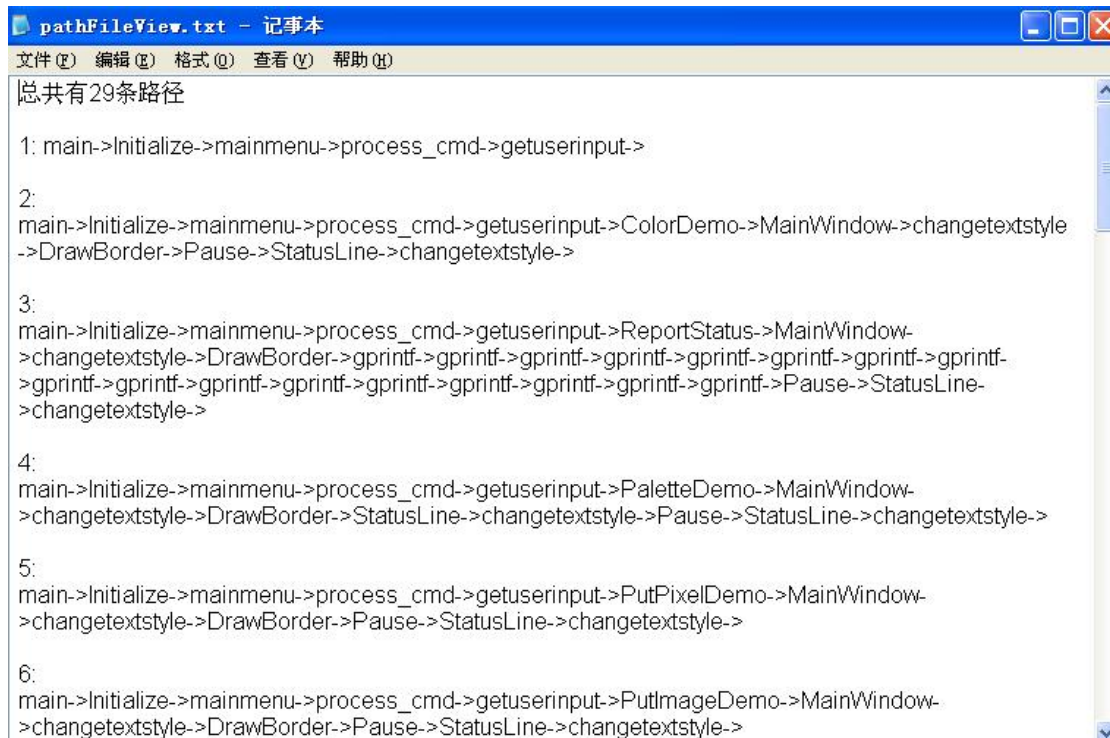
4. 程序变更影响分析

1) 静态路径

工具能精确的计算出所有的静态路径（一条不多一条不少）

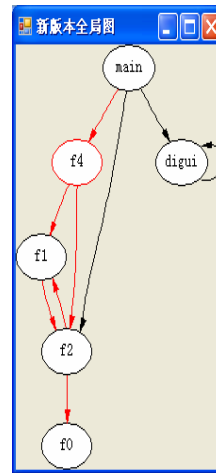
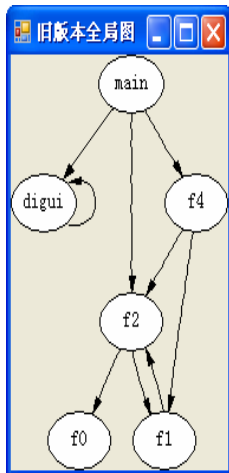


以文本形式显示出所有的静态路径。



## 2) 程序变更影响分析

根据程序变更情况和程序所包含静态路径，自动找出因程序变更而受到影响的静态路径（图中红线部分所示）；

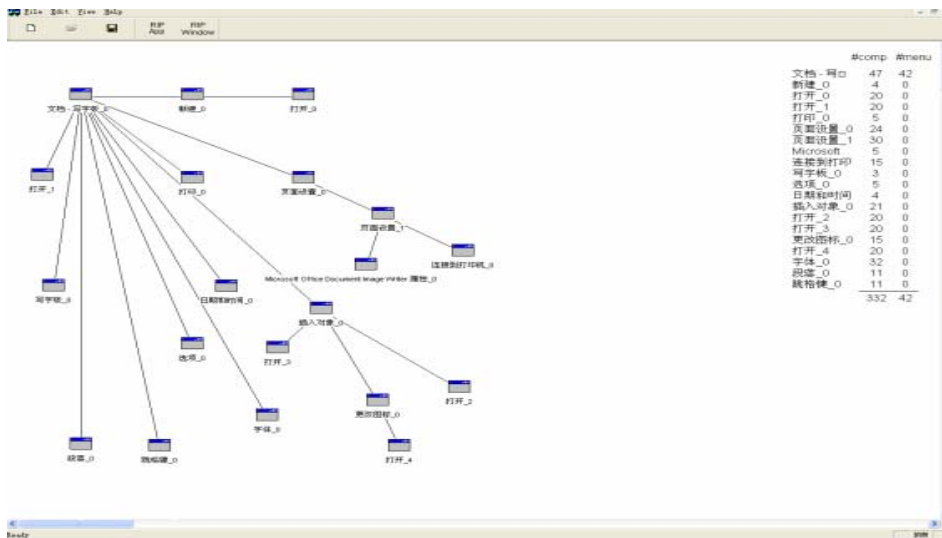


5. 测试用例优化。根据插装程序给出的测试用例对代码覆盖情况，通过读取测试用例覆盖文件，对测试用例进行优化，剔除冗余和覆盖率较低的测试用例，得到满足覆盖要求最简测试用例。

6. 回归测试用例选择。

根据最简测试用例和回归测试需要覆盖的程序路径，找出能够满足覆盖要求的测试用例。

7. 软件功能图自动生成：通过分析软件执行代码，生成软件功能图；



8. 软件功能变更标识

通过比较软件修改前后的功能图，自动识别软件的功能变更情况，并以图形方式显示；

9. 软件功能变更影响分析

根据软件功能变更情况，给出因软件变更而受到影响的功能路径