回归测试用例选择

Regression Test Selection

yfy

介绍

软件维护的费用占了软件产品总费用的2/3 [Pressman 1987; Schach 1992]。回归测试是软件过程中必不可少的一项活动。回归测试是指修改了原有程序后，重新进行测试以确认程序的正确性，并且没有向原已验证过的部分引入错误。回归测试最多可以占到软件维护费用的一半 [Bezier 1990; Leung and White 1989]。

背景

回归测试

软件测试是一种从程序的输入空间中选取一些输入来执行该程序，以期发现程序中的错误的过程。软件测试是衡量软件质量的一种重要方法。进行软件测试首先需要选取一组输入，分析出期望的程序执行的行为，与实际程序执行的行为进行比较。测试用例包括提供给程序的输入，和期望的运行结果。测试套件指一组测试用例。

为了评价测试套件的质量，我们通常以测试执行的覆盖率作为一项指标。覆盖率指的是程序中一些结构的覆盖率，常见的有语句覆盖，分支覆盖，断言覆盖，路径覆盖。例如语句覆盖，指的是测试执行所覆盖的语句数占可执行的总语句数的比例。覆盖信息有多种方式可以获得。一种方式是代码插装，对于程序P，生成对应的P’，P’在执行一个测试用例t时，可以记录下覆盖相关的信息。另一种方式是修改运行时环境，这样当用t测试P时，运行时环境可以记录哪些结构被覆盖。

CFG

建立CFG

使用一个栈来保存当前的上下文，每一帧保存类型，入口和出口

对于一个程序，建立ProgramEntry结点和ProgramExit结点

对于switch语句，每一个case会连出一条边

if

路径记录

语句前加log，记录结点的id

把所有语句放入block中

过程内分析

过程间分析

前文描述了过程内的回归测试选择技术，现实中，通常程序都不只包含一个过程，下面描述如何把这一技术扩展为过程间的，这样可以适用于整个程序，甚至系统。

当遇到一个函数调用语句S，它对应的S'和S字面上相同，并且这是一个我们建立了CFG的可以找到的函数f，那么跟踪进入这个函数及其对应的函数f'，使用已有的过程内的方法进行比较。比较完成后，记录下f的比较结果，值为{未访问，已访问，全部选择}之一。“未访问”表示尚未比较过f和f'。“全部选择”表示只要进入了f，那么一定需要重新测试，选择全部进入f的测试用例，S之后的语句不需要在分析。“已访问”表示f中存在一条路径从f返回，走这条路径的测试用例不需要重新测试，S之后的语句需要继续分析。

记录下函数比较的结果，不仅是为了去除重复计算，这是必须的，考虑如下程序P：

function hello() {

hello();

}

hello();

P'未做修改。如果不记录函数比较结果，那么比较过程将不会终止。在进入一个函数，我们先把f的比较结果标识为“已访问”，比较完成后，如果需要，再修改为“全部选择”。这样就可以解决这一问题。