

第二部分 软件测试

2.14 对变量的测试





- 针对循环的测试
 - 单个循环节点的测试
 - 循环的初值边界值
 - 循环的迭代
 - 循环包含的语句执行过程；测试增量的变化；每次循环涉及到的变量的取值是否按预期规律发生变化；重复多次循环是否导致误差累积
 - 串联循环节点测试
 - 每个循环节点进行测试



- 嵌套循环节点的测试
 - 先测试最内层的循环，然后逐步外推，直到测试到最外层的循环
 - 考虑4种特殊组合：内外层组合，小小，大大，大小，小大
- 非结构化循环结构：
 - 首先，建议修改代码
 - 其次，如果不能修改代码，则先对单次循环体进行测试，兼顾嵌套循环条件下对循环次数的多种特殊组合



- 理解对变量的测试背景知识
- 掌握对变量测试的分析
- 重难点：掌握对变量测试的分析及方法



1 对变量的测试背景知识

2 对变量的测试分析

3 对变量的测试总结

对变量测试的背景知识



- 为什么进行变量测试
 - 对路径和循环的测试中，最难测试的就是非结构化的程序结构
 - 测试人员，无权修改代码，仍可针对重要路径和循环结构进行测试
 - 除此之外，还需从变量的角度，重点关注对代码中计算最为复杂的变量
- 重点关注计算结果对函数返回值影响最大的变量
- 最可能出错的变量

对变量测试的背景知识



- 变量的3种缺陷定义：
 - 变量在使用前从**未定义过**（在编译时，会有提示，不需人工查找）
 - 变量被定义，但**从未使用过**
 - 变量在使用前，**被多次定义**
 - 这种缺陷通过编译器无法发现的，在测试过程中需要重点关注
 - 寻找这种错误的过程，是一种静态分析的过程，不需要设计测试用例

对变量的测试相关概念

• 定义节点:

- 若被测变量 v 的值在某条包含该变量的语句 n 处发生改变, 则称该语句是关于变量 v 的定义节点, 记做 $DEF(v, n)$
- 输入语句、赋值语句 (对该变量赋值)、循环控制语句 (循环变量) 都是定义节点

• 使用节点:

- 若被测变量 v 的值在某条包含该变量的语句 n 处被使用, 则称该语句是关于变量 v 的使用节点, 记做 $USE(v, n)$
- 输出语句、赋值语句 (变量 v 对其他变量的赋值)、条件语句、循环控制语句 都是使用节点
- 若变量在赋值语句中使用, 称计算使用节点
- 若变量在条件判定表达式中被使用, 则为谓词使用节点

对变量的测试相关概念

- 定义/使用节点对

- 由被测变量 v 的一对定义节点和使用节点构成的一个二元组称为该变量的定义/使用节点对

- 定义/使用路径

- 从被测变量 v 的一个定义节点开始执行，到该变量的某个使用节点结束的一条路径称为该变量的一条定义/使用路径，记做du-path

- 定义/清除路径

- 若被测变量 v 的一条定义/使用路径中不包含该变量的其他定义节点，则该路径称为定义清除路径，记做dc-path



1 对变量的测试背景知识

2 对变量的测试分析

3 对变量的测试总结

对变量的测试分析

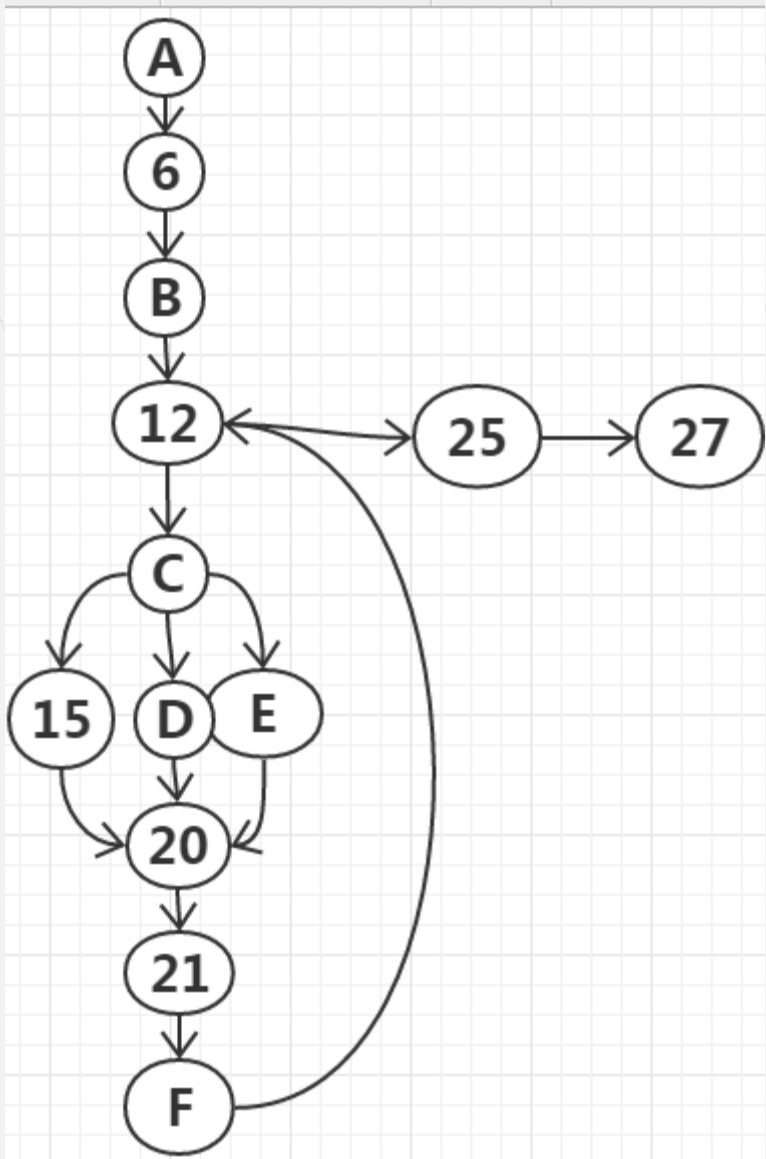
- 对关键变量的数据流测试的一般步骤如下：
 - 确定需要重点测试的变量 v
 - 确定变量 v 的所有定义节点和使用节点
 - 确定变量 v 的所有定义节点和使用节点对
 - 假设变量 v 有 p 个定义节点，分别为 $DEF(v, n1)$, $DEF(v, n2)$ $DEF(v, np)$ ；有 q 个使用节点，分别为 $USE(v, m1)$, $USE(v, m2)$ $USE(v, mq)$ ，则检查每所有定义/使用节点对有 pq 个
 - 判断每条定义/使用路径是否为高风险路径
 - 条定义/使用路径，若该路径不是定义清除路径，则认为该路径是高风险路径，应重点测试

对变量的测试分析



变量	定义节点	使用节点	定义/使用节点
alcoholPrice	3	13	
redPrice	4	13	
beerPrice	5	13	
totalSales	6,21	21,25	21
totalCommission	7,22	22,25,26	22
monthSales	8,13	14,15,16,17,19,20,21	
Commission	9,15,17,19	20,22	
Input	10	13,20	
month	11,23	12,13,20,23	23

对变量的测试分析



- A 表示3--5
- B 表示7--11
- C 表示13--14
- D 表示16--17
- E 表示18--19
- F 表示 22--24

对变量的测试分析



- 关于变量alcoholPrice, redwinePrice和beerPrice的定义和使用情况较为简单，只有一条路径，不做过多分析
- 关于变量totalSales的数据流分析：
 - totalSales的定义/使用对共有4对，分别是：6—21, 6—25, 21—21, 21—25
 - 6—21：
 - 其执行路径有3种可能，但均不包含totalSales, 对变量不产生影响，后续将此3节点合并15/D/E; 此路径只包含变量的一个定义和使用节点，为定义清除路径，对应一次进入循环体，并完整执行判定体的情况

对变量的测试分析



- 6—25:

- 由于该路径包含21这个定义焦点，而且可以看到，此路径执行过程穿越了整个循环结构，变量被多次（6次）赋值，由此得出，此路径是一条高风险路径，需要**重点测试**

- 21—21:

执行路径为：21, F, 12, C, 15/D/E, 20, 21, 这种路径可看做一种退化路径，这种路径在测试中不予考虑

- 21—23:

- 该路径仅包含一次定义和使用，为定义清除路径，不予考虑

对变量的测试分析



- 由以上分析可知：
 - 路径6—25为高风险路径，需要**重点测试**
 - 同时又一次验证循环是最容易导致定义引用异常缺陷，需要重点测试
 - 其他变量可以以此方法为例进行分析，如：
`monthSales, month`等



1 对变量的测试背景知识

2 对变量的测试分析

3 对变量的测试总结

对变量的测试总结



- 大多数情况下不需要逐一分析程序中的所有变量，一般根据**重要变量**进行数据流分析
- 数据流分析主要关注单个变量，**特别关注变量接受赋值和使用**的点，该法主要关注变量从定义到使用过程的执行路径是否可靠



- 对变量测试的背景知识
 - 为什么对变量进行测试
- 对变量测试的分析
 - **相关概念**：定义、使用、定义/使用对、定义/使用路径、定义清除路径
 - 由变量使用造成的高风险路径
- 对变量测试的总结



Question