

## 5.3 白盒测试

### • 白盒测试概述

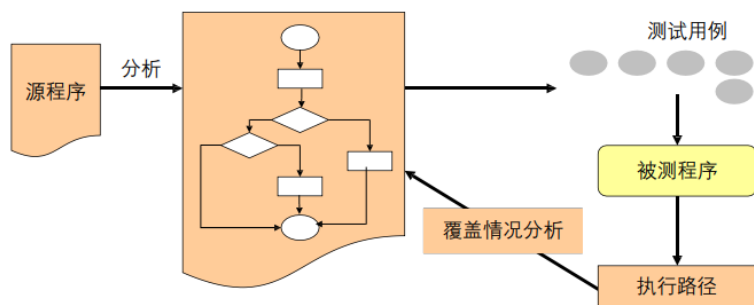
#### • 概念

又称为结构测试或逻辑驱动测试

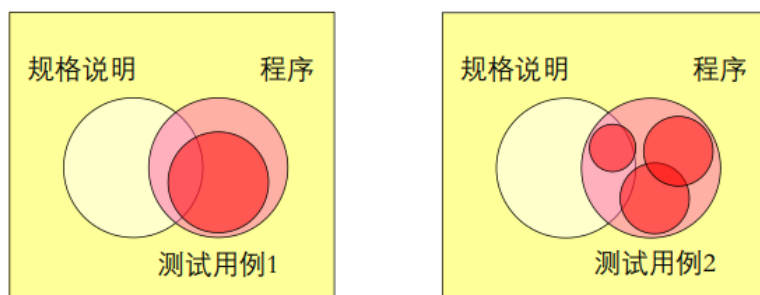
#### • 目的

对路径、分支、循环和内部数据结构进行检查

#### • 测试过程



#### • Venn 图



#### • 测试覆盖标准

##### • 特点

- 以内部逻辑为基础设计测试用例
- 考虑对内部逻辑的覆盖程度

### • 白盒测试方案

#### • 白盒测试方案技术之一：逻辑测试

##### • 语句覆盖

- 使得被测试程序中的每条可执行语句至少被执行一次

##### • 判定覆盖（分支覆盖）

- 每一判定的每个分支至少执行一次

##### • 条件覆盖

- 每一判定的每个条件，分别按照真、假至少各执行一次

##### • 判定/条件覆盖

- 同时满足判定覆盖和条件覆盖的要求

- 条件组合覆盖

- 求出判定中所有条件的各种可能组合值，每一种可能的条件组合至少执行一次

- 白盒测试方案技术之二：控制结构测试

- 基本路径测试

- 基本路径测试是一种白盒测试技术。使用这种技术设计测试用例，首先计算过程设计结果的逻辑复杂度，并以该复杂度为指南定义执行路径的基本集合

- 特点

- 在程序控制图的基础上，通过分析控制构造的环路复杂性，导出基本可执行路径集合，从而设计测试用例
- 设计出的测试用例要保证在测试中程序的每条可执行语句至少执行一次
- 程序中的每个条件至少被测试一次

- 前提条件

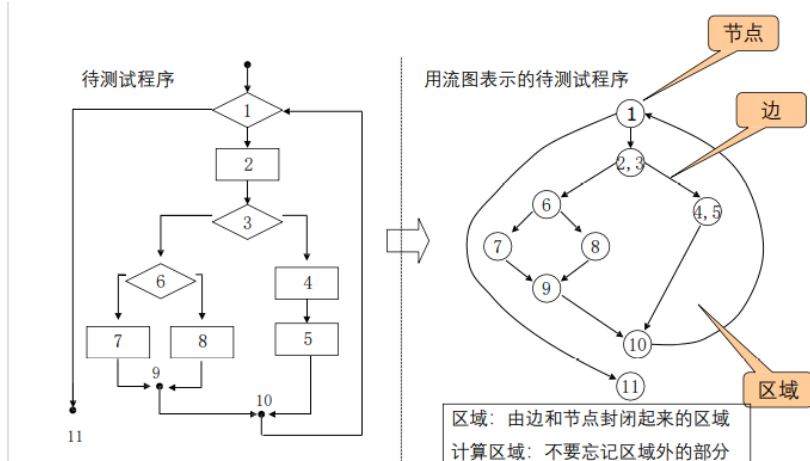
- 测试人员基本明确了测试软件的逻辑结构

- 测试过程

- 设计样例，程序执行，对程序路径进行测试。测试结果是分析实际的测试结果与预期的结果是否一致。

- 步骤

- 以设计或源代码为基础，画出相应的流图



- 确定所的流图的环复杂度

- 环复杂度是一种为程序逻辑复杂性提供定量测度的软件度量，将该度量用于**计算程序的基本的独立路径数目，为确保所有语句至少执行一次的测试数量的上界**

- 但是测试用例越少，可能发现的错误就越少

- 计算方法

- 流图中区域的数量

- 圈复杂度  $V(G) = E - N + 2$ , E 为边的数量, N 是流图中结点的数量

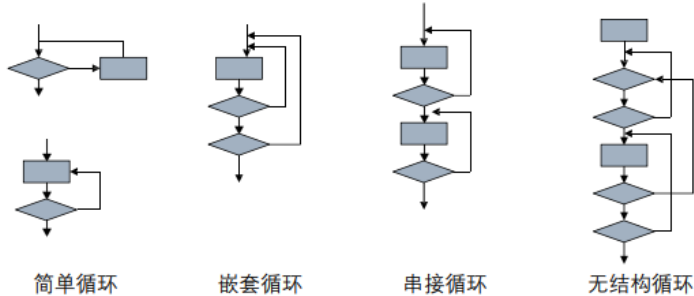
- 圈复杂度  $V(G) = P + 1$ , P 是判定结点的数量

- **确定独立路径的基本集合**

- 独立路径：一条路径，至少包含一条在定义该路径之前不曾用过的边(至少引入程序的一个新处理语句集合或一个新条件)

- **准备测试用例**，执行基本集合中每条路径

- 循环测试



- 简单循环

- 0次循环
- 1次循环
- 2次循环
- m次循环
- n (最大) 次循环, n+1次循环, n-1次循环

- 串接 (连锁) 循环

- 若互相独立，采用与简单循环相同的方法测试
- 否则使用嵌套循环的方法来处理

- 嵌套循环

- 从最内层循环开始，设置所有其他层循环为最小值
- 对最内层循环做简单循环的全部测试。测试时保持所有外层循环的循环变量为最小值。对越界值和非法值做类似的测试
- 逐步外推直到测试完毕
- 对全部各层循环同时取最小循环次数，或者同时取最大循环次数。

- 不规则循环

- 独立路径：一条路径，至少包含一条在定义该路径之前不曾用过的边(至少引入程序的一个新处理语句集合或一个新条件)