



## 8.5 软件测试技术

### 白盒测试

- 白盒测试的概念
- 控制流程图
- 路径测试
- 语句测试
- 分支测试
- 条件组合测试
- 循环的测试

#### 白盒测试：

白盒测试也称结构测试或逻辑驱动测试，它是知道产品内部工作过程，可通过测试来检测产品内部动作是否按照规格说明书的规定正常进行，按照程序内部的结构测试程序，检验程序中的每条通路是否都有能按预定要求正确工作，而不顾它的功能。

白盒测试的主要方法有逻辑覆盖、基本路径测试等，主要用于软件验证。

问题：白盒测试依据什么？



# 白盒测试

- 白盒测试的概念
- 控制流程图
- 路径测试
- 语句测试
- 分支测试
- 条件组合测试
- 循环的测试

## 依据程序的逻辑结构-白盒测试技术

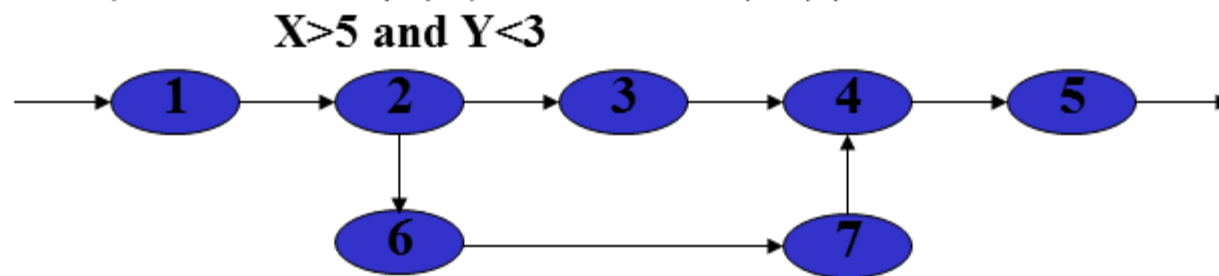
### (1) 关于建立被测对象模型

控制流程图:一种表示程序控制结构的图形工具,其基本元素是节点、判定、过程块。

其中:过程块是既不能由判定、也不能由节点分开的一组程序语句;

判定:是一个程序点,此处控制流可以分叉;

节点:是一个程序点,此处控制流可以结合。



其中: 1、3、5、6、7为过程块

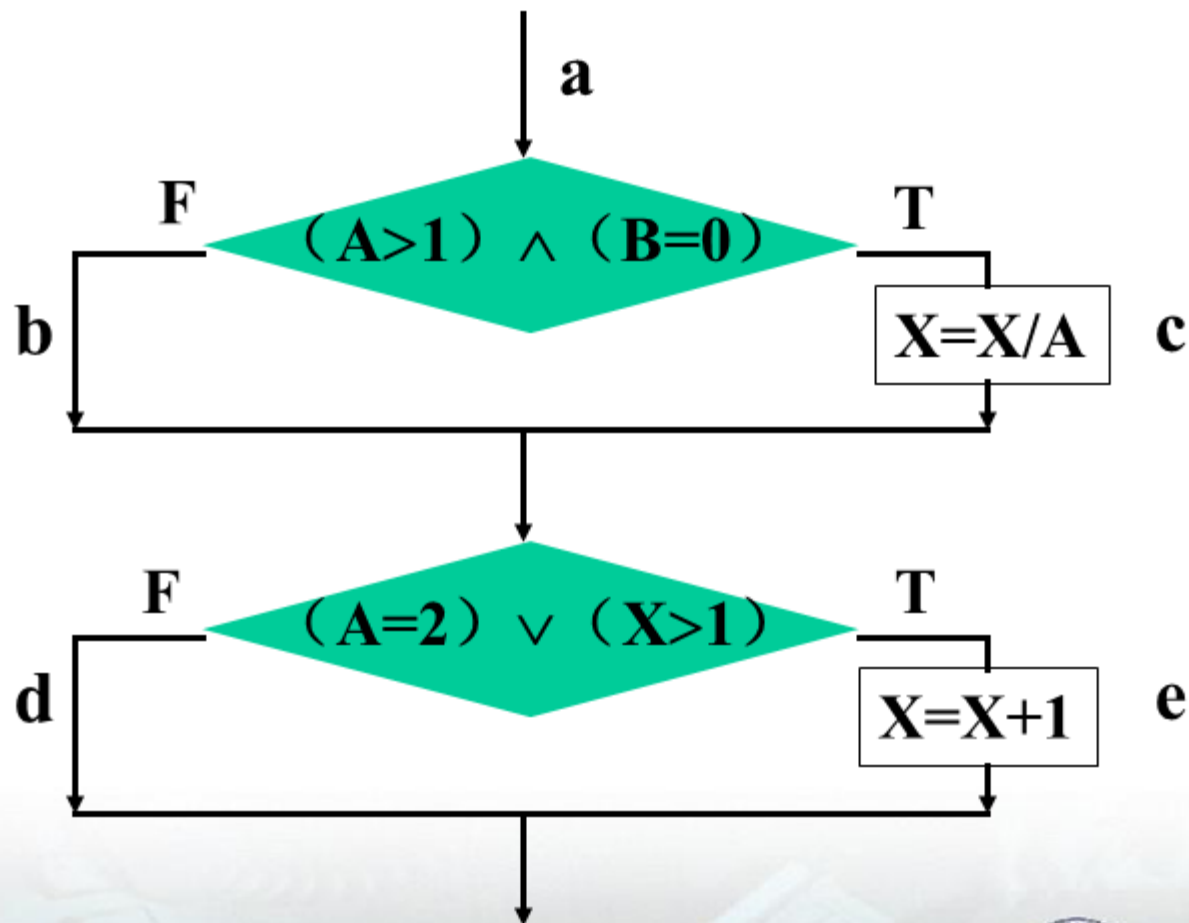
2为判定, 4为节点



# 白盒测试

- 白盒测试的概念
- 控制流程图
- 路径测试
- 语句测试
- 分支测试
- 条件组合测试
- 循环的测试

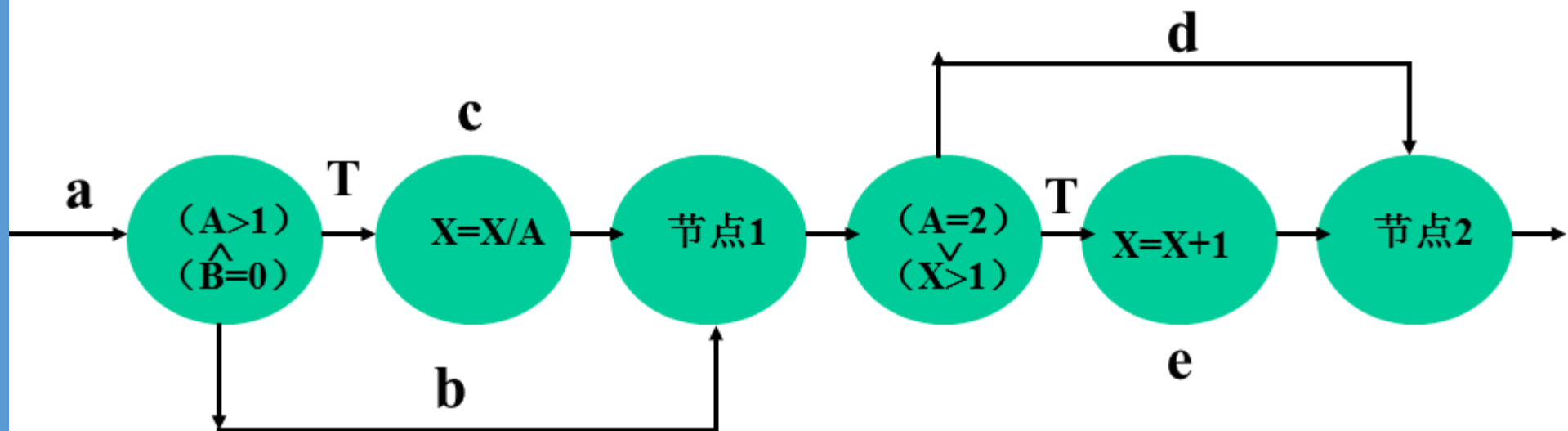
例如：以下为一个程序流程图，其中该例子中有两个判断，每个判断都包含复合条件的逻辑表达式。



# 白盒测试

- 白盒测试的概念
- 控制流程图
- 路径测试
- 语句测试
- 分支测试
- 条件组合测试
- 循环的测试

其控制流程图为：



# 白盒测试

- 白盒测试的概念
- 控制流程图
- 路径测试
- 语句测试
- 分支测试
- 条件组合测试
- 循环的测试

## (2) 各种测试方法

该控制流程图有4条不同的路径。4条路径可表示为：

L1 ( $a \rightarrow c \rightarrow e$ ) 简写ace 、 L2 ( $a \rightarrow b \rightarrow d$ ) 简写abd

L3 ( $a \rightarrow b \rightarrow e$ ) 简写abe 、 L4 ( $a \rightarrow c \rightarrow d$ ) 简写acd

- **路径测试 (PX)：**执行所有可能的穿过程序的控制流程路径。

一般来说，这一测试严格地限制为所有可能的入口/出口路径。如果遵循这一规定，则我们说达到了100%路径覆盖率。在路径测试中，该策略是最强的，但一般是不可实现的。

针对该例子，要想实现路径覆盖，可选择以下一组测试用例（规定测试用例的设计格式为：**【输入的 (A, B, X) ，输出的 (A, B, X) 】**）。

测试用例	覆盖路径
<b>【 (2, 0, 4) , (2, 0, 3) 】</b>	L1
<b>【 (1, 1, 1) , (1, 1, 1) 】</b>	L2
<b>【 (1, 1, 2) , (1, 1, 3) 】</b>	L3
<b>【 (3, 0, 3) , (3, 0, 1) 】</b>	L4





# 白盒测试

- 白盒测试的概念
- 控制流程图
- 路径测试
- 语句测试
- 分支测试
- 条件组合测试
- 循环的测试



- 语句测试 (P1)：至少执行程序中所有语句一次。

如果遵循这一规定，则我们说达到了100%语句覆盖率（用C1表达）。

在该例子中，只要设计一种能通过路径ace的测试用例，就覆盖了所有的语句。所以可选择测试用例如下：

【(2, 0, 4), (2, 0, 3)】 覆盖L1

语句覆盖是最弱的逻辑覆盖准则。

问题：就该程序而言，如果两个判断的逻辑运算写错，例如，第一个判断中的逻辑运算符“ $\wedge$ ”错写成了“ $\vee$ ”，或者第二个判断中的逻辑运算符“ $\vee$ ”错写成了“ $\wedge$ ”，利用上面的测试用例，仍可覆盖其中2个语句，而发现不了判断中逻辑运算符出现的错误。



# 白盒测试

- 白盒测试的概念
- 控制流程图
- 路径测试
- 语句测试
- 分支测试
- 条件组合测试
- 循环的测试

• **分支测试 (P2)**：至少执行程序中每一分支一次。如果遵循这一规定，则我们说达到了100%分支覆盖率（用C2表示）。

分支覆盖是一种比语句覆盖稍强的逻辑覆盖。但若程序中分支的判定是由几个条件联合构成时，它未必能发现每个条件的错误。

例如对于以上例子，如果选择路径L1和L2，就可得到实现分支覆盖的测试用例：

【(2, 0, 4), (2, 0, 3)】 覆盖L1

【(1, 1, 1), (1, 1, 1)】 覆盖L2

如果选择路径L3和L4，还可得另一组可用的测试用例：

【(2, 1, 1), (2, 1, 2)】 覆盖L3

【(3, 0, 3), (3, 1, 1)】 覆盖L4

问题：分支覆盖还不能保证一定能查出在判断的条件中存在的错误。例如，在该例子中，若第二个分支 $X > 1$ 错写成 $X < 1$ ，利用上述两组测试用例进行测试，无法查出这一错误。因此，需要更强的逻辑覆盖准则去检验判定的内部条件。



# 白盒测试

- 白盒测试的概念
- 控制流程图
- 路径测试
- 语句测试
- 条件组合测试
- 循环的测试

## • 条件组合测试

条件组合测试是一种具有更强逻辑覆盖的测试。

条件组合测试，就是设计足够的测试用例，使每个判定中的所有可能的条件取值组合至少执行一次。如果遵循这一规定，则我们说就实现了条件组合覆盖。只要满足了条件组合覆盖，就一定满足分支覆盖。

在条件组合覆盖技术发展过程中，最初，在设计测试用例时，人们只考虑使分支中各个条件的所有可能结果至少出现一次。但发现该测试技术未必能覆盖全部分支。例如，在上图的例子中，程序段中有四个条件： $A > 1$ ， $B = 0$ ， $A = 2$ ， $X > 1$ 。

条件 $A > 1$  取真值标记为T1，取假值标记为F1

条件 $B = 0$  取真值标记为T2，取假值标记为F2

条件 $A = 2$  取真值标记为T3，取假值标记为F3

条件 $X > 1$  取真值标记为T4，取假值标记为F4

在设计测试用例时，要考虑如何选择测试用例实现T1、F1、T2、F2、T3、F3、T4、F4的全部覆盖：





# 白盒测试

- 白盒测试的概念
- 控制流程图
- 路径测试
- 语句测试
- 分支测试
- 条件组合测试
- 循环的测试



例如，可设计如下测试用例实现条件覆盖：

测试用例	通过路径	条件取值	覆盖分支
【 (1, 0, 3) , (1, 0, 4) 】	L3	F1 T2 F3 T4	b, e
【 (2, 1, 1) , (2, 1, 2) 】	L3	T1 F2 T3 F4	b, e

从上面的测试用例，可以看到该组测试用例虽然实现了判定中各条件的覆盖，但没有实现分支覆盖，因为该组测试用例只覆盖了第一个判断的取假分支和第二个判断的取真分支。为此，人们又进一步提出了条件组合覆盖技术。

例如，在该例子中，前一个判定有4种条件组合：

- (1)  $(A > 1)$ ,  $(B = 0)$ , 标记为 T1 、 T2;
- (2)  $(A > 1)$ ,  $(B \neq 0)$ , 标记为 T1 、 F2, ;
- (3)  $(A \leq 1)$ ,  $(B = 0)$ , 标记为 F1 、 T2;
- (4)  $(A \leq 1)$ ,  $(B \neq 0)$ , 标记为 F1 、 F2;

后一个判定又有4种条件组合：

- (5)  $(A = 2)$ ,  $(X > 1)$ , 标记为 T3、 T4;
- (6)  $(A = 2)$ ,  $(X \leq 1)$ , 标记为 T3、 F4;
- (7)  $(A \neq 2)$ ,  $(X > 1)$ , 标记为 F3、 T4;
- (8)  $(A \neq 2)$ ,  $(X \leq 1)$ , 标记为 F3、 F4。



# 白盒测试

- 白盒测试的概念
- 控制流程图
- 路径测试
- 语句测试
- 分支测试
- 条件组合测试
- 循环的测试

可以采用以下四组测试数据，从而实现条件组合覆盖。

测 试 用 例	覆盖条件	覆盖组合号	通过路径
【 (2, 0, 4) , (2, 0, 3) 】	T1 T2 T3 T4	(1) (5)	L1
【 (2, 1, 1) , (2, 1, 2) 】	T1 F2 T3 F4	(2) (6)	L3
【 (1, 0, 3) , (1, 0, 4) 】	F1 T2 F3 T4	(3) (7)	L3
【 (1, 1, 1) , (1, 1, 1) 】	F1 F2 F3 F4	(4) (8)	L2

这组测试用例实现了分支覆盖，也实现了条件的所有可能取值的组合的覆盖。

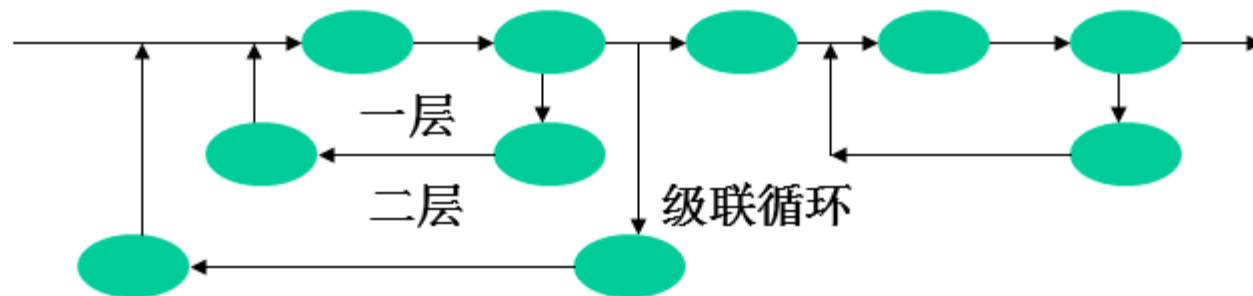
问题：语句覆盖、分支覆盖、条件组合覆盖、路径覆盖之间的关系？



# 白盒测试

- 白盒测试的概念
- 控制流程图
- 路径测试
- 语句测试
- 分支测试
- 条件组合测试
- 循环的测试

## (3) 循环情况的路径选取



嵌套循环

还要考虑循环变量的具体情况

## 关键路径的选取

主要功能路径

没有功能的路径

最短路径

...



北京大学