

第十二章 黑盒测试

- ◆ 1 黑盒测试的基本概念
- ◆ 2 等价类划分
- ◆ 3 边界值分析法
- ◆ 4 因果图法
- ◆ 5 功能图法
- ◆ 6 测试方法选择策略
- ◆ 7 黑盒测试工具介绍
- ◆ 小结



1 黑盒测试的基本概念

- 黑盒测试也称功能测试，通过测试来检测每个功能是否都能正常使用。
- 测试中把程序看作一个不能打开的黑盒子，在完全不考虑程序内部结构和内部特性的情况下，在程序接口进行测试。
- 只检查程序功能是否按照需求规格说明书的规定正常使用，程序是否能适当地接收输入数据而产生正确的输出信息。
- 黑盒测试着眼于程序外部结构，不考虑内部逻辑结构，主要针对软件界面和软件功能进行测试。
- 黑盒测试试图发现以下类型的错误：
 - 功能错误或遗漏；
 - 性能错误；
 - 界面错误；
 - 初始化和终止错误。
 - 数据结构或外部数据库访问错误；



第十二章 黑盒测试

- ◆ 1 黑盒测试的基本概念
- ◆ 2 等价类划分
 - 划分等价类
 - 划分等价类的方法
 - 设计测试用例:
- ◆ 3 边界值分析法
- ◆ 4 因果图法
- ◆ 5 功能图法
- ◆ 6 测试方法选择策略
- ◆ 7 黑盒测试工具介绍
- ◆ 小结



2 等价类划分

- ◇ 等价类划分法是一种黑盒测试的技术
- ◇ 不考虑程序的内部结构，把所有可能的输入数据，即程序的输入域划分成若干部分（子集），然后从每一个子集中选取少数具有代表性的数据作为测试用例。
- ◇ 该方法是一种重要的，常用的黑盒测试用例设计方法。



2 等价类划分

——划分等价类

等价类划分可有两种不同的情况：有效等价类和无效等价类

➤ 有效等价类

- 指对于程序的规格说明来说是合理的，有意义的输入数据构成的集合。
- 利用有效等价类可检验程序是否实现了规格说明中所规定的功能和性能。

➤ 无效等价类

- 与有效等价类的定义恰巧相反，不符合需求规格说明书。



2 等价类划分

——划分等价类的方法

➤ 确定等价类的六条原则

- 在输入条件规定了取值范围或值的个数的情况下，则可以确立一个有效等价类和两个无效等价类。
- 在输入条件规定了输入值的集合或者规定了“必须如何”的条件的情况下，可确立一个有效等价类和一个无效等价类。
- 在输入条件是一个布尔量的情况下，可确定一个有效等价类和一个无效等价类。
- 在规定了输入数据的一组值（假定 n 个），并且程序要对每一个输入值分别处理的情况下，可确立 n 个有效等价类和一个无效等价类。
- 在规定了输入数据必须遵守的规则的情况下，可确立一个有效等价类（符合规则）和若干个无效等价类（从不同角度违反规则）。
- 在确知已划分的等价类中各元素在程序处理中的方式不同的情况下，则应再将该等价类进一步的划分为更小的等价类。



2 等价类划分

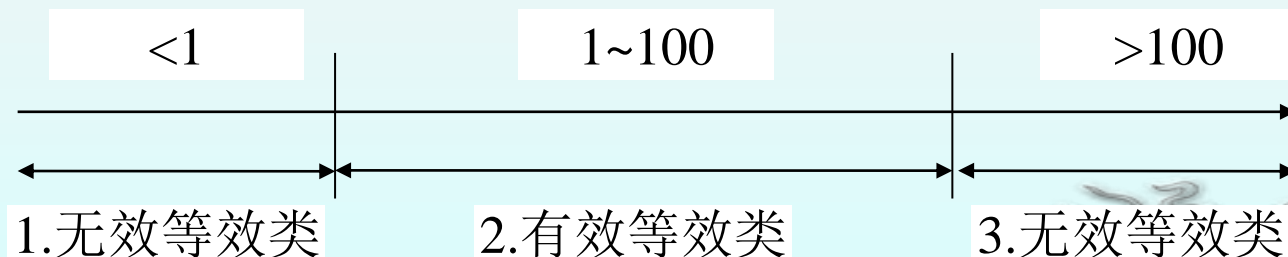
——设计测试用例

➤ 乘法器测试用例

- 已知乘数1有1-100共计100个取值，乘数2也有1-100共计100个取值，请设计乘法器程序的测试用例

用例编号	乘数1	乘数2	乘积
1	1	1	1
2	1	2	2
3	1	3	3
...

- 先根据输入条件确定有效等价类和无效等价类



2 等价类划分

——设计测试用例

➤ 乘法器测试用例

- 然后从划分出的等价类中按以下三个原则设计测试用例
 - ◆ 每一个等价类规定一个唯一的编号。
 - ◆ 设计一个新的测试用例，使其尽可能多地覆盖尚未被覆盖地有效等价类，重复这一步。直到所有的有效等价类都被覆盖为止。
 - ◆ 设计一个新的测试用例，使其仅覆盖一个尚未被覆盖的无效等价类，重复这一步，直到所有的无效等价类都被覆盖为止。



2 等价类划分

——设计测试用例

➤ 乘法器测试用例

- 然后从划分出的等价类中按以下三个原则设计测试用例

用例编号	所属等价类	乘数1	乘数2	乘积
1	2	3	20	60
2	1	-10	2	提示“请输入1~100之间的整数”
3	3	200	3	提示“请输入1~100之间的整数”



2 等价类划分

——设计测试用例

➤ 三角形计算测试用例

输入条件	有效等价类	无效等价类
是否三角形的3条边		
是否等腰三角形		
是否等边三角形		



2 等价类划分

——设计测试用例

➤ 三角形计算测试用例

输入条件	有效等价类	无效等价类
是否三角形的3条边	$(A > 0)$, (1) $(B > 0)$, (2) $(C > 0)$, (3) $(A + B > C)$, (4) $(B + C > A)$, (5) $(A + C > B)$, (6)	$(A \leq 0)$, (7) $(B \leq 0)$, (8) $(C \leq 0)$, (9) $(A + B \leq C)$, (10) $(B + C \leq A)$, (11) $(A + C \leq B)$, (12)
是否等腰三角形	$(A = B)$, (13) $(B = C)$, (14) $(C = A)$, (15)	$(A \neq B)$ and $(B \neq C)$ and $(C \neq A)$, (16)
是否等边三角形	$(A = B)$ and $(B = C)$ and $(C = A)$, (17)	$(A \neq B)$, (18) $(B \neq C)$, (19) $(C \neq A)$, (20)

2 等价类划分

——设计测试用例

➤ 三角形计算测试用例

用例编号	【A, B, C】	覆盖等价类	输出
1	【3, 4, 5】	(1), (2), (3), (4), (5), (6)	一般三角形
2	【0, 1, 2】	(7)	不能构成三角形
3	【1, 0, 2】	(8)	
4	【1, 2, 0】	(9)	
5	【1, 2, 3】	(10)	
6	【1, 3, 2】	(11)	
7	【3, 1, 2】	(12)	



2 等价类划分

——设计测试用例

➤ 三角形计算测试用例

用例编号	【A, B, C】	覆盖等价类	输出
8	【3, 3, 4】	(1), (2), (3), (4), (5), (6), (13)	等腰三角形
9	【3, 4, 4】	(1),(2),(3),(4),(5), (6), (14)	
10	【3, 4, 3】	(1),(2),(3),(4),(5),(6),(15)	
11	【3, 4, 5】	(1),(2),(3),(4),(5), (6),(16)	非等腰三角形
12	【3, 3, 3】	(1),(2),(3),(4),(5),(6),(17)	是等边三角形
13	【3, 4, 4】	(1),(2),(3),(4),(5),(6),(14),(18)	非等边三角形
14	【3, 4, 3】	(1), (2),(3),(4),(5),(6), (15), (19)	
15	【3, 3, 4】	(1),(2),(3),(4),(5), (6),(13),(20)	



第十二章 黑盒测试

- ◆ 1 黑盒测试的基本概念
- ◆ 2 等价类划分
- ◆ 3 边界值分析法
- ◆ 4 因果图法
- ◆ 5 功能图法
- ◆ 6 测试方法选择策略
- ◆ 7 黑盒测试工具介绍
- ◆ 小结



3 边界值分析法

- ◇ 边界值分析法就是对输入或输出的边界值进行测试的一种黑盒测试方法。
- ◇ 通常边界值分析法是作为对等价类划分法的补充，这种情况下，其测试用例来自等价类的边界。



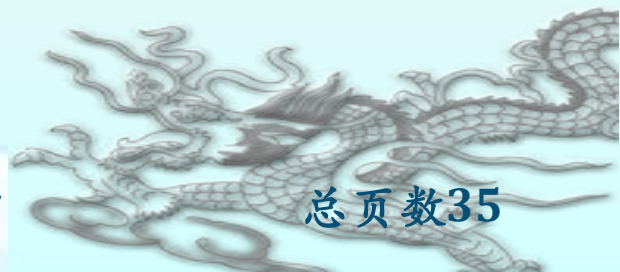
3 边界值分析法

- 对边界值设计测试用例，应遵循以下几条原则：
- 如果输入条件规定了值的范围，则应取刚达到这个范围的边界的值，以及刚刚超越这个范围边界的值作为测试输入数据。
 - 如果输入条件规定了值的个数，则用最大个数、最小个数、比最小个数少1、比最大个数多1的数作为测试数据。
 - 根据规格说明的每个输出条件，使用前面的原则①。
 - 根据规格说明的每个输出条件，应用前面的原则②。
 - 如果程序的规格说明给出的输入域或输出域是有序集合，应选取集合的第一个元素和最后一个元素作为测试用例。
 - 如果程序中使用了一个内部数据结构，则应当选择这个内部数据结构边界上的值作为测试用例。
 - 分析规格说明，找出其他可能的边界条件。



第十二章 黑盒测试

- ◆ 1 黑盒测试的基本概念
- ◆ 2 等价类划分
- ◆ 3 边界值分析法
- ◆ 4 因果图法
 - 因果图设计方法
 - 因果图测试用例
- ◆ 5 功能图法
- ◆ 6 测试方法选择策略
- ◆ 7 黑盒测试工具介绍
- ◆ 小结



4 因果图法

——因果图设计方法

适合于检查程序输入条件的各种组合情况

- 等价类划分方法和边界值分析方法，着重考虑输入条件，未考虑输入条件之间的联系。
- 如果在测试时必须考虑输入条件的各种组合，可能的组合数将是天文数字。因此必须考虑使用一种适合于描述对于多种条件的组合，相应产生多个动作的形式来考虑设计测试用例，这就需要利用因果图。
- 因果图方法最终生成的就是决策表。它适合于检查程序输入条件的各种组合情况。



4 因果图法

——因果图设计方法

➤ 利用因果图导出测试用例需要经过以下几个步骤：

- 分析软件规格说明描述中，哪些是原因（即输入条件或输入条件的等价类），哪些是结果（即输出条件），并给每个原因和结果赋予一个标识符。
- 分析程序规格说明的描述中语义的内容，并将其表示成连接各个原因与各个结果的“因果图”。
- 标明约束条件。由于语法或环境的限制，有些原因和结果的组合情况是不可能出现的。为表明这些特定的情况，在因果图上使用若干个标准的符号标明约束条件。
- 把因果图转换成决策表。
- 为决策表中每一列表示的情况设计测试用例。



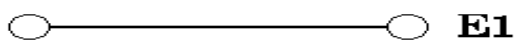
4 因果图法

——因果图设计方法

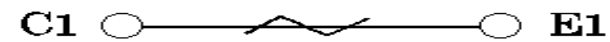
➤ 因果图的基本图形符号

- 恒等：若原因出现，则结果出现；若原因不出现，则结果也不出现。
- 非(\sim)：若原因出现，则结果不出现；若原因不出现，则结果出现。
- 或(\vee)：若几个原因中有1个出现，则结果出现；若几个原因都不出现，则结果不出现。
- 与(\wedge)：若几个原因都出现，结果才出现。若其中有1个原因不出现，则结果不出现。

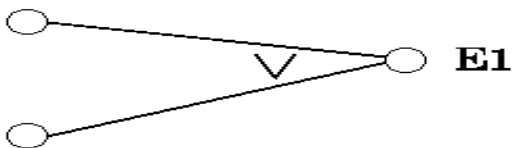
(a) 恒等



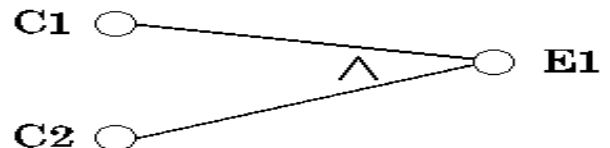
(b) 非



(c) 或



(d) 与

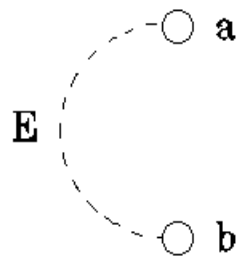


4 因果图法

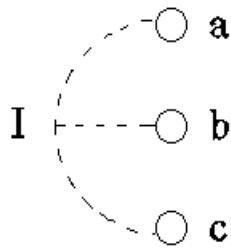
——因果图设计方法

➤ 因果图的约束符号

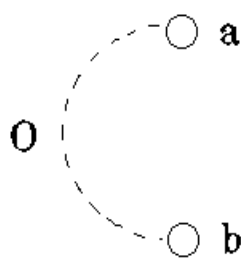
- E(互斥): 表示a、b两个原因不会同时成立,两个中最多有一个可能成立。
- I(包含): 表示a、b、c这3个原因中至少有一个必须成立。
- O(惟一): 表示a和b当中必须有一个,且仅有一个成立。
- R(要求): 表示当a出现时b必须也出现。a出现时不可能b不出现。
- M(屏蔽): 表示当a是1时,b必须是0.而当a为0时,b的值不定。



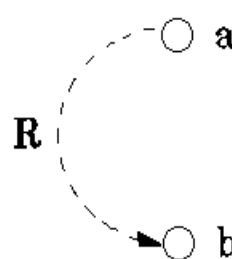
(1) E (互斥·排他)



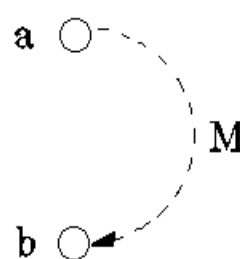
(2) I (包含·或)



(3) O (唯一)



(4) R (要求)



(5) M (屏蔽)

4 因果图法

——因果图测试用例

有一个处理单价为5角钱的饮料的自动售货机软件测试用例的设计。其规格说明如下：

- “若投入5角钱或1元钱的硬币，押下『橙汁』或『啤酒』的按钮，则相应的饮料就送出来。
- 若售货机没有零钱找，则一个显示『零钱找完』的红灯亮，这时在投入1元硬币并押下按钮后，饮料不送出来而且1元硬币也退出来；
- 若有零钱找，则显示『零钱找完』的红灯灭，在送出饮料的同时退还5角硬币。



4 因果图法

——因果图测试用例

➤ 分析这一段说明，列出原因和结果

■ 原因：

- ◆ 1. 售货机有零钱找
- ◆ 2. 投入1元硬币
- ◆ 3. 投入5角硬币
- ◆ 4. 押下橙汁按钮
- ◆ 5. 押下啤酒按钮

■ 结果：

- ◆ 21. 售货机〔零钱找完〕灯亮
- ◆ 22. 退还1元硬币
- ◆ 23. 退还5角硬币
- ◆ 24. 送出橙汁饮料
- ◆ 25. 送出啤酒饮料



4 因果图法

——因果图测试用例

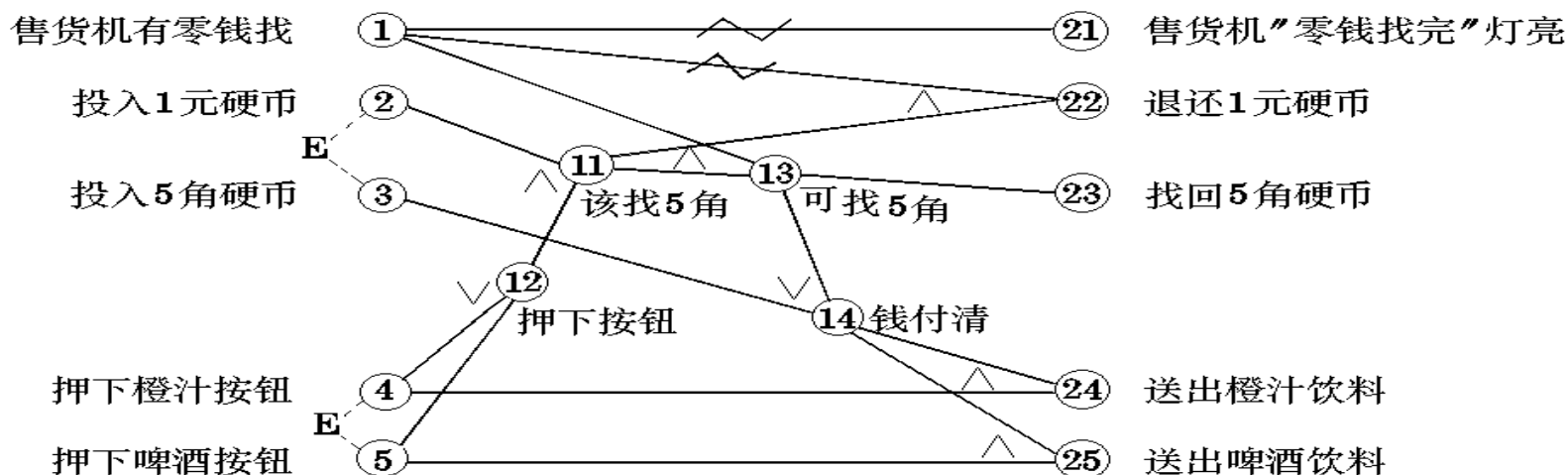
➤ 画出因果图

- 建立中间结点，表示处理的中间状态

- ◆ 11. 投入1元硬币且押下饮料按钮
- ◆ 12. 押下〔橙汁〕或〔啤酒〕的按钮
- ◆ 13. 应当找5角零钱并且售货机有零钱找
- ◆ 14. 钱已付清

- 由于2与3，4与5不能同时发生，分别加上约束条件E。

- 1. 售货机有零钱找
- 2. 投入1元硬币
- 3. 投入5角硬币
- 4. 押下橙汁按钮
- 5. 押下啤酒按钮



4 因果图法

——因果图测试用例

➤ 转换成决策表

序号		1	2	3	4	5	6	7	8	9	10	1	2	3	4	5	6	7	8	9	20	1	2	3	4	5	6	7	8	9	30	1	2	
条 件	①	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
	②	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	
	③	1	1	1	1	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1	1	0	0	0	0	1	1	1	1	0	0	0	0
	④	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0	0
	⑤	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	0
中 间 结 果	⑪						1	1	0		0	0	0		0	0	0							1	1	0		0	0	0		0	0	0
	⑫						1	1	0		1	1	0		1	1	0							1	1	0		1	1	0		1	1	0
	⑬						1	1	0		0	0	0		0	0	0							0	0	0		0	0	0		0	0	0
	⑭						1	1	0		1	1	1		0	0	0							0	0	0		1	1	1		0	0	0
结 果	⑳						0	0	0		0	0	0		0	0	0							1	1	1		1	1	1		1	1	1
	㉑						0	0	0		0	0	0		0	0	0							1	1	0		0	0	0		0	0	0
	㉒						1	1	0		0	0	0		0	0	0							0	0	0		0	0	0		0	0	0
	㉓						1	0	0		1	0	0		0	0	0							0	0	0		1	0	0		0	0	0
	㉔						0	1	0		0	1	0		0	0	0							0	0	0		0	1	0		0	0	0
测试用例							Y	Y	Y		Y	Y	Y		Y	Y							Y	Y	Y		Y	Y	Y		Y	Y		

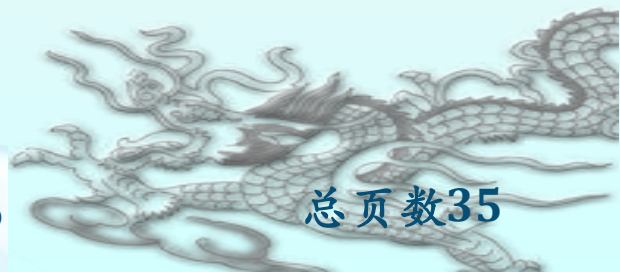
① 决策表中，阴影部分表示因违反约束条件的不可能出现的情况，删去

② 第16列与第32列因什么动作也没做，也删去

③ 最后可根据剩下的16列作为确定测试用例的依据

第十二章 黑盒测试

- ◆ 1 黑盒测试的基本概念
- ◆ 2 等价类划分
- ◆ 3 边界值分析法
- ◆ 4 因果图法
- ◆ 5 功能图法
 - 功能图设计方法
 - 功能图法生成测试用例
- ◆ 6 测试方法选择策略
- ◆ 7 黑盒测试工具介绍
- ◆ 小结



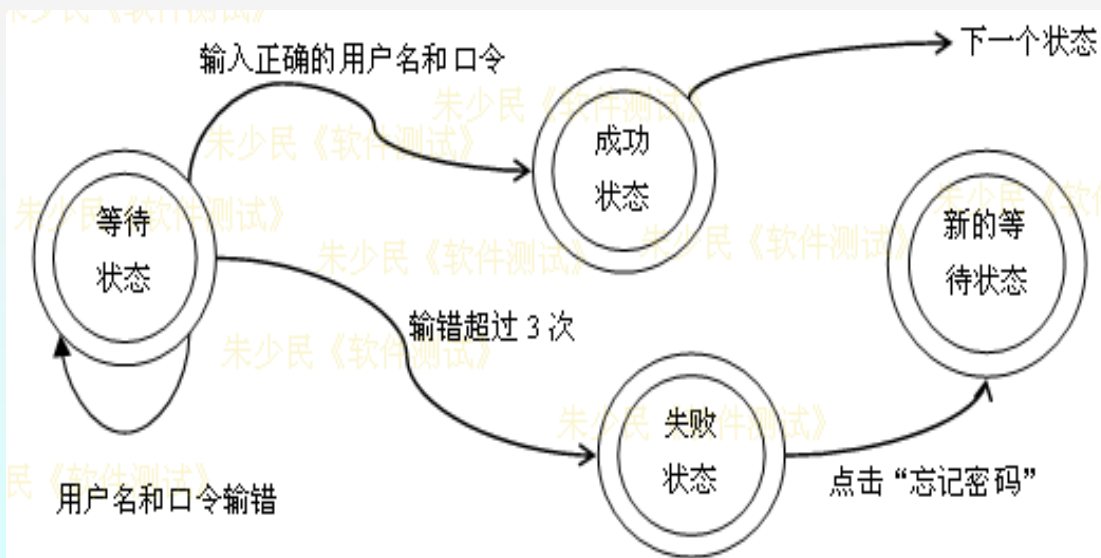
5 功能图法

——功能图设计方法

功能图方法是一种黑盒、白盒混合用例设计方法

- 用功能图形象地表示程序的功能说明，并机械地生成功能图的测试用例。
- 功能图模型由状态迁移图和逻辑功能模型构成。

■ 状态迁移图



- ① 状态指出数据输入的位置或时间，而迁移则指明状态的改变。
- ② 用节点表示状态，用弧度代替迁移，则状态迁移图就可转化为一个程序的控制流程图形式。

5 功能图法

——功能图设计方法

■ 逻辑功能模型

- ◆ 表示在状态中输入条件和输出条件之间的对应关系。
- ◆ 逻辑功能模型只适合于描述静态说明，输出数据仅由输入数据决定
- ◆ 测试用例则是由测试中经过的一系列状态和在每个状态中必须依靠输入/输出数据满足的一对条件组成。

逻辑功能模型的决策表形式（1—成功，0—失败）

输入	正确的用户名	错误的用户名	错误的用户名	正确的用户名
	错误的密码	正确的密码	错误的密码	正确的密码
输出	0	0	0	1
	错误提示	错误提示	错误提示	
状态	等待重新输入	等待重新输入	等待重新输入	进入新的状态



5 功能图法

——功能图法生成测试用例

➤ 从功能图生成测试用例的过程如下。

■ 生成局部测试用例：

- ◆ 在每个状态中，从因果图生成局部测试用例。
- ◆ 局部测试库由原因值（输入数据）组合与对应的结果值（输出数据或状态）构成。

■ 测试路径生成：

- ◆ 利用上面的规则生成从初始状态到最后状态的测试路径。

■ 测试用例合成：

- ◆ 合成测试路径与功能图中每个状态的局部测试用例。
- ◆ 结果是视状态到最后状态的一个状态序列，以及每个状态中输入数据与对应输出数据组合。

■ 测试用例的合成算法：

- ◆ 采用条件构造树



第十二章 黑盒测试

- ◆ 1 黑盒测试的基本概念
- ◆ 2 等价类划分
- ◆ 3 边界值分析法
- ◆ 4 因果图法
- ◆ 5 功能图法
- ◆ 6 测试方法选择策略
- ◆ 7 黑盒测试工具介绍
- ◆ 小结



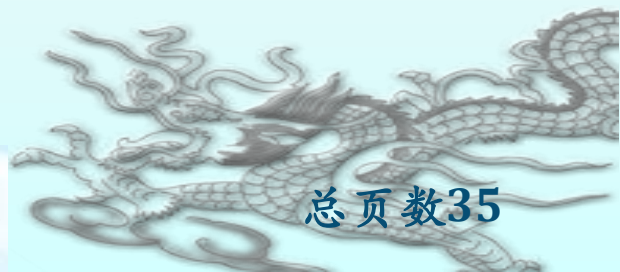
6 测试方法选择策略

- 首先进行等价类划分，包括输入条件和输出条件的等价划分，将无限测试变成有限测试，这是减少工作量和提高测试效率最有效的方法。
- 在任何情况下都必须使用边界值分析方法。经验表明，用这种方法设计出的测试用例发现程序错误的能力最强。
- 可以用错误推测法追加一些测试用例，这需要依靠测试工程师的智慧和经验。
- 对照程序逻辑，检查已设计出的测试用例的逻辑覆盖程度。如果没有达到要求的覆盖标准，应当再补充足够的测试用例。



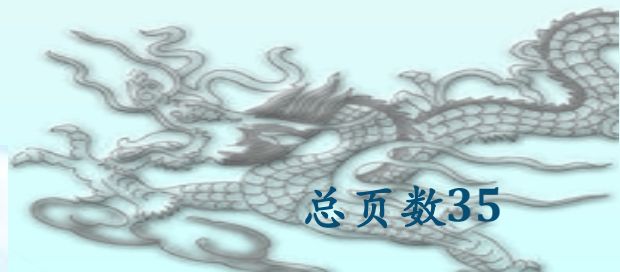
6 测试方法选择策略

- 如果程序的功能说明中含有输入条件的组合情况，则一开始就可选用因果图法和判定表驱动法。
- 对于参数配置类的软件，要用正交试验法选择较少的组合方式达到最佳效果。
- 功能图法也是很好的测试用例设计方法，我们可以通过不同时期条件的有效性设计不同的测试数据。
- 对于业务流清晰的系统，可以利用场景法贯穿整个测试案例过程，在案例中综合使用各种测试方法。



第十二章 黑盒测试

- ◆ 1 黑盒测试的基本概念
- ◆ 2 等价类划分
- ◆ 3 边界值分析法
- ◆ 4 因果图法
- ◆ 5 功能图法
- ◆ 6 测试方法选择策略
- ◆ 7 黑盒测试工具介绍
- ◆ 小结



7 黑盒测试工具介绍

- LoadRunner
- QuickTest Pro



小结

- ◆ 本章主要讲解了等价类划分法，边界值分析法，因果图法，功能图分析法，综合的案例分折，同时对黑盒测试的几种方法进行的比较，最后介绍了常用的几种黑盒测试的工具。
- ◆ 本章的实践性较强，希望举一反三，将这些测试技术和平时的软件开发和测试工作结合起来。

