西北工業大學
NORTHWESTERN POLYTECHNICAL UNIVERSITY

第三章　黑盒测试方法

授课教师：　郑炜

第三章黑盒测试方法

- 3.1 测试用例综述
  - 3.1.1 测试用例设计原则
  - 3.1.2 测试用例设计步骤
  - 3.1.3 测试用例的构成
- 3.2 等价类设计方法
  - 3.2.1 等价类划分
  - 3.2.2 等价类划分方法
  - 3.2.4 等价类划分的测试运用

- 3.3 边界值设计方法
  - 3.3.1 边界值分析法原理
  - 3.3.2 边界值分析原则
  - 3.3.3 健壮性分析
  - 3.3.4 边界值分析法的测试运用
- 3.4 因果图和决策表设计方法
  - 3.4.1 因果图原理
  - 3.4.2 因果图法应用
  - 3.4.3 决策表法及其应用

- 3.5 正交试验设计方法
  - 3.5.1 正交试验设计法原理
  - 3.5.2 利用正交试验法设计测试用例
- 3.6 组合测试方法
  - 3.6.1 基本概念
  - 3.6.2 构造方法
  - 3.6.3 组合测试工具的使用

- 3.1 测试用例综述
  - 3.1.1 测试用例设计原则
  - 3.1.2 测试用例设计步骤
  - 3.1.3 测试用例的构成

3.1.1 测试用例设计原则

- **单个测试用例最小化原则**

  测试用例覆盖边界更清晰，测试结果对软件缺陷指向性更强，用例间耦合度低，彼此间干扰低；

- **测试用例替代产品文档功能原则**

  测试用例忠实反应产品功能；重点考虑测试代码的可读性和描述性；

- **单次投入成本和多次投入成本原则**

  测试用例设计应对测试成本的考虑全面地体现在测试的设计、执行和维护的各个阶段；

- **测试结果分析和调试最简单化原则**

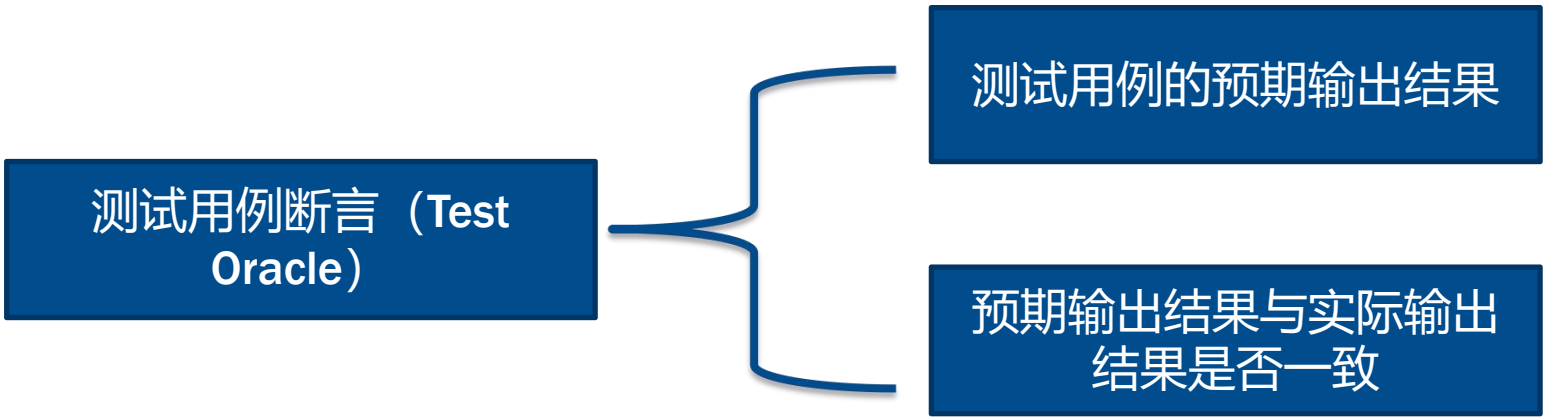
  使测试结果分析和测试调试更简单；

3.1.2 测试用例设计步骤

测试需求分析	明确测试软件、模块需求，测试需求
业务流程分析	明确软件主流程，条件备选流程，数据流向，关键判断条件
测试用例设计	确定测试套件，测试场景，针对测试场景，确定测试用例，增加测试数据，完成测试用例
测试用例评审	确保测试过程和方法的正确性，以及是否有遗漏的测试点
测试用例更新完善	

3.1.3 测试用例的构成


测试输入
执行条件
预期输出结果


测试用例断言（Test Oracle）
测试用例的预期输出结果
预期输出结果与实际输出结果是否一致

- 3.2 等价类设计方法
  - 3.2.1 等价类划分
  - 3.2.2 等价类划分方法
  - 3.2.4 等价类划分的测试运用

3.2.1 等价类划分

- 基本概念

等价类划分法是把所有可能的输入数据，即程序的输入划分成若干部分（子集），然后从每一个子集中选取少量具有代表性的数据作为测试用例；

- 等价类的划分的两种情况

有效等价类

是指对程序的规格说明而言，合理且有意义的输入数据构成的集合。

无效等价类

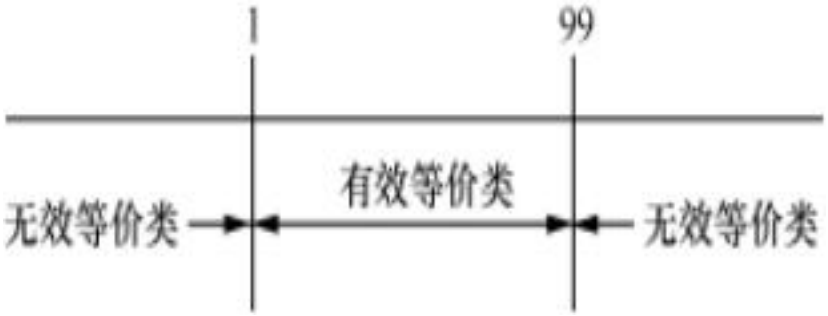
是指对程序的规格说明而言，不合理的、无意义的输入数据构成的集合。

此处应注意，正确的术语应该是“无效值的等价类”，而不是“无效等价类”，因为等价类本身并不是无效的，只是这个等价类对于某个特定的输入值是无效的。

3.2.2 等价类划分方法

- 如果一个输入条件规定了输入值的范围，那么可以得到三个等价类：一个有效等价类和两个无效等价类。

例：规定输入值的范围是1~99,如图所示，那么可以得到3个等价类：一个合法等价类{1,...,99}；两个非法等价类{x|x<1}和{x|x>99}。



3.2.2 等价类划分方法

- 如果输入条件规定了必须满足的情形，那么生成两个等价类：一个为有效等价类；另一个为无效等价类。

例：输入的第一个字符必须是一个数字，那么得到两个等价类—一个为有效等价类{s|s的第一个字符是数字}；另一个为无效等价类{s|s的第一个字符不是数字}。

3.2.2 等价类划分方法

- 如果处理每个有效输入的方式都不相同，那么为每个有效输入生成一个有效等价类。

例：如果在一个菜单中选择菜单项作为输入，那么应该为每个菜单项定义一个等价类。

- 如果输入条件规定了有效输入的数量（假定为N），那么为正确的输入数量定义一个有效等价类，同时定义两个无效等价类。

3.2.2 等价类划分方法

- 如果输入条件规定了必须满足的情形，那么生成两个等价类：一个为有效等价类；另一个为无效等价类。

例：输入的第一个字符必须是一个数字，那么得到两个等价类—一个为有效等价类{s|s的第一个字符是数字}；另一个为无效等价类{s|s的第一个字符不是数字}。

- 如果一个等价类中的元素被程序处理的方式不同，那么就把该等价类分割为更小的等价类。一种直观的识别方式是简单值、普通值、极端值和典型值等。

测试用例的选择原则

- 为每个划分好的等价类规定唯一编号
- 设计一个新的测试用例，使其尽可能多地覆盖尚未被覆盖的有效等价类，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。

3.2.2 等价类划分方法

等价类划分的四个类别

类型	缺陷假设类型	是否考虑异常区域
弱一般等价类	单软件缺陷假设	不考虑
强一般等价类	多软件缺陷假设	不考虑
弱健壮等价类	单软件缺陷假设	考虑
强健壮等价类	多软件缺陷假设	考虑

单软件缺陷假设是指“失效极少是由两个或两个以上的软件缺陷同时发生引起的”。多软件缺陷假设，则是指“失效是由两个或两个以上软件缺陷同时作用引起的”，要求在选取测试用例时同时让多个变量取极值。

3.2.2 等价类划分方法

- **弱一般等价类测试**

  不考虑无效数据，测试用例使用每个等价类中的一个值。它选取的测试用例只需要覆盖到有效等价类，

- **强一般等价类测试**

  每个等价类至少要选择一个测试用例。它不考虑无效等价类，选取测试用例时，各个有效区间的组合都要覆盖到

- **弱健壮等价类测试**

  对于有效输入，使用每个有效等价类的一个值；对于无效输入，测试用例只使用一个无效值，其余值都是有效的。

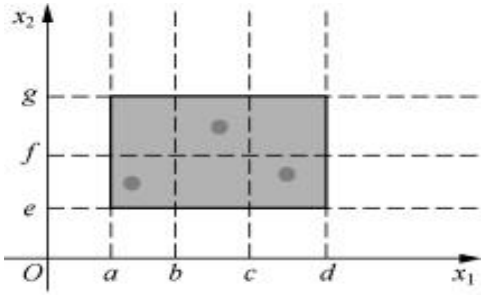
- **强健壮等价类测试**

  每个无效等价类和有效等价类的组合都要覆盖到，考虑所有的有效和无效情况，

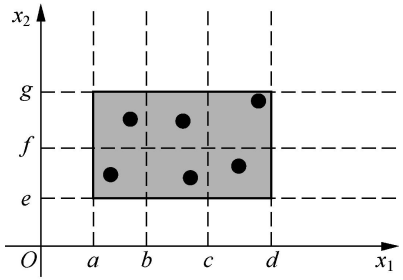
3.2.2 等价类划分方法

例： 某函数F有两个输入变量x1、x2，要求两个输入变量的取值范围如下：a≤x1≤d，区间为[a,b],(b,c),[c,d]；e≤x2≤g，区间为[e,f),[f,g]，求各类等价类测试示例。

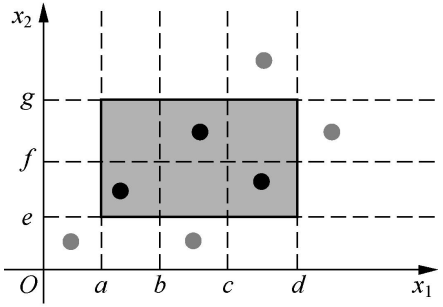
分析 x1和x2的无效区间为：x1＜a；x2＞d；x2＜e；x2＞g。



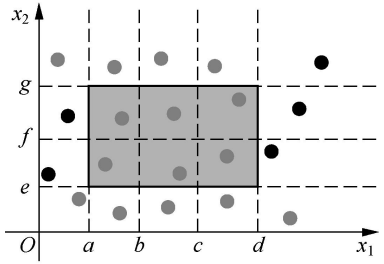
弱一般等价类测试示例



强一般等价类测试示例



弱健壮等价类测试示例



强健壮等价类测试示例

3.2.3 等价类划分的测试运用

1、三角形问题的等价类测试

例：输入3个整数a、b和c分别作为三角形的3条边，通过程序判断这3条边的组成情况是等边三角形、等腰三角形、一般三角形，还是不构成三角形。

三角形问题的等价类

有效等价类	编号	无效等价类	编号
整数	1	一边为非整数 两边为非整数 三边均为非整数	4 5 6
3个数	2	只有一条边 只有两条边 多于三条边	7 8 9
1≤a≤100 1≤b≤100 1≤c≤100	3	一边为0 两边为0 三边为0	10 11 12
		一边＜0 两边＜0 三边＜0	13 14 15
		一边＞100 两边＞100 三边＞100	16 17 18

覆盖无效类的测试用例表

测试用例	输入*a*, *b*, *c*	期望输出	覆盖等价类
Test2	1.5, 4, 5	提示“请输入1～100的整数”	4
Test3	3.5, 2.5, 5	提示“请输入1～100的整数”	5
Test4	2.5, 4.5, 5.5	提示“请输入1～100的整数”	6
Test5	3	提示“请输入3条边长”	7
Test6	4, 5	提示“请输入3条边长”	8
Test7	2, 3, 4, 5	提示“请输入3条边长”	9
Test8	3, 0, 8	提示“边长不能为0”	10
Test9	0, 6, 0	提示“边长不能为0”	11
Test10	0, 0, 0	提示“边长不能为0”	12
Test11	−3, 4, 6	提示“边长不能为负”	13
Test12	2, −7, −5	提示“边长不能为负”	14
Test13	−3, −4, −5	提示“边长不能为负”	15
Test14	101, 4, 5	提示“请输入1～100的整数”	16
Test15	3, 101, 102	提示“请输入1～100的整数”	17
Test16	101, 104, 105	提示“请输入1～100的整数”	18

2、NextDate函数问题

NextDate函数包含3个变量，即月份（month）、日期（day）和年份（year），函数的输出为输入日期的后一天。例如，输入为2017年4月25日，则函数的输出为2017年4月26日。函数要求输入变量均为整数值，并且满足下列条件：

（1）1≤month≤12；
（2）1≤day≤31；
（3）1912≤year≤2050。

使用简单等价类划分方法，可以划分为以下有效等价类：

M1 = {month:1≤month≤12}
D1 = {day:1≤day≤31}
Y1 = {year:1912≤year≤2050}

若M1、D1、Y1这3个条件中的任意一个无效，那么NextDate函数都会产生一个输出，其无效等价类：

M2 = {month:month＜1}　　D3 = {day:day＞31}
M3 = {month:month＞12}　　Y2 = {year:year＜1912}
D2 = {day:day＜1}　　Y3 = {year:year＞2050}

3.2.3 等价类划分的测试运用

西北工业大学 NORTHWESTERN POLYTECHNICAL UNIVERSITY

NextDate函数的一般等价类测试用例

测试用例	输入			期望输出
	day	*month*	*year*	
TestCase1	25	4	2017	2017年4月26日

NextDate函数的弱健壮等价类测试用例

测试用例	输入			期望输出
	day	*month*	*year*	
TestCase1	25	4	2017	2017年4月26日
TestCase2	25	0	2017	*month*不在1～12中
TestCase3	25	13	2017	*month*不在1～12中
TestCase4	0	4	2017	*day*不在1～31中
TestCase5	32	4	2017	*day*不在1～31中
TestCase6	25	4	1911	*year*不在1912～2050中
TestCase7	25	4	2051	*year*不在1912～2050中

弱健壮等价类测试中的无效测试用例则只包含一个无效值，其他都是有效值，即含有单软件缺陷假设。

3.2.3 等价类划分的测试运用

NextDate函数的强健壮等价类测试用例

测试用例	输入			期望输出
	day	*month*	*year*	
TestCase1	25	−1	2017	*month*不在1～12中
TestCase2	−25	4	2017	*day*不在1～31中
TestCase3	25	4	1900	*year*不在1912～2050中
TestCase4	−1	−4	2017	变量*day*、*month*无效，变量*year*有效
TestCase5	−1	4	1900	变量*day*、*year*无效，变量*month*有效
TestCase6	25	−4	1911	变量*month*、*year*无效，变量*day*有效
TestCase7	−25	−4	2051	变量*day*、*month*、*year*无效

强健壮等价类测试考虑了更多的无效值情况。强健壮等价类测试中的无效测试用例可以包含多个无效值，即包含多个软件缺陷假设。因为NextDate函数有3个变量，所以对应的强健壮等价类测试用例可以包含一个无效值，2个无效值或3个无效值。

3.2.3 等价类划分的测试运用

改进等价类划分测试NextDate函数

关于每个月的天数问题，可以详细划分为以下等价类。

M1 = {month:month有30天}
M2 = {month:month有31天，除去12月}
M3 = {month:month是2月}
M4 = {month:month是12月}
D1 = {day:1≤day≤27}
D2 = {day:day=28}
D3 = {day:day=29}
D4 = {day:day=30}
D5 = {day:day=31}
Y1 = {year:year是闰年}
Y2 = {year:year不是闰年}

NextDate函数的改进等价类划分测试用例

测试用例	输入			期望输出
	day	*month*	*year*	
TestCase1	30	6	2017	2017年7月1日
TestCase2	31	8	2017	2017年9月1日
TestCase3	27	2	2017	2017年2月28日
TestCase4	28	2	2017	2017年3月1日
TestCase5	29	2	2016	2016年3月1日
TestCase6	31	12	2017	2018年1月1日
TestCase7	31	9	2017	不可能的输入日期
TestCase8	29	2	2017	不可能的输入日期
TestCase9	30	2	2017	不可能的输入日期
TestCase10	9	15	2017	变量*month*无效
TestCase11	35	9	2017	变量*day*无效
TestCase12	9	9	2100	变量*year*无效

- 3.3 边界值设计方法
  - 3.3.1 边界值分析原理
  - 3.3.2 边界值分析原则
  - 3.3.3 健壮性分析
  - 3.3.4 边界值分析法的测试运用

3.3.1 边界值分析法原理

1．边界值测试原理

边界值分析法的基本思想是在等价类的极端情况下考虑软件测试工作，因为错误很容易发生在输入值的关键点，即从合法变为非法的那一点。

2．边界条件

边界条件即输入定义域或输出值域的边界，而不是内部。通常情况下，软件测试所包含的边界检验有以下几种类型：数字、字符、位置、质量、大小、速度、方位、尺寸、空间等。

3．边界值分析法的优缺点

边界值分析法 在测试范围的边界值上进行考虑，相对来说是比较简便易行的，生成测试数据的成本也很低。但对于整个测试过程来说，它的测试用例不够充分，也不能发现测试变量之间的依赖关系，因此只能作为初步测试用例使用。

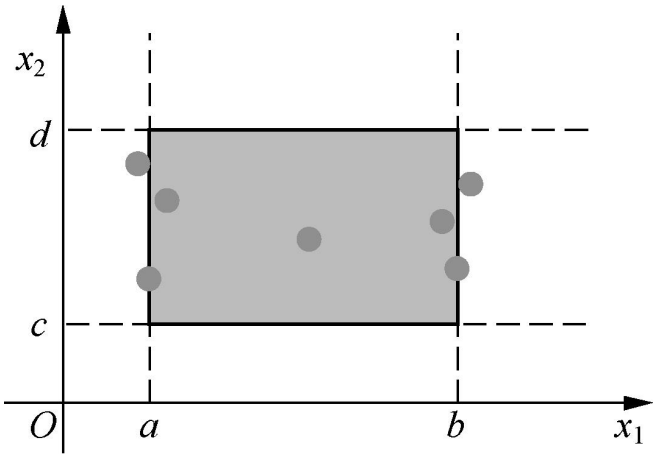
3.3.2 边界值分析原则

- 如果输入条件规定了一个取值范围，就从范围的边界生成合法测试用例，并为恰好超出边界的输入值生成非法测试用例。
- 如果输入条件规定了值的个数，就用最大个数、最小个数、比最小个数少1、比最大个数多1的个数创建测试用例。
- 如果输出条件规定了一个取值范围，就为输出范围的边界生成合法测试用例，并为恰好超出边界的输出值生成非法测试用例。
- 如果程序的输入或输出是一个有序的集合（如顺序文件、线性列表、表格等），就需要关注第一个和最后一个元素。
- 寻找其他边界条件。例如，如果合法范围是−999～999，那么测试用例包括合法测试用例−999、合法测试用例999、非法测试用例−1000、非法测试用例1000。

3.3.3 健壮性分析

健壮性指软件在发生异常的情况下还能正常运行的能力。健壮性测试是边界值分析的一种简单扩展。除了对变量的5个边界分析取值，还要考虑略超过最大值（max+）和略小于最小值（min-）时的情况。

对于一个有两个输入变量的程序而言，健壮性边界值测试用例选取如图所示



在进行健壮性测试时，需要注意以下几点：

（1）健壮性测试需要注意预期的输出。

（2）健壮性测试的主要价值是观察异常情况的处理。

（3）软件质量要素的衡量标准包括软件的容错性。

（4）软件的容错性度量可以是软件从非法输入中恢复的能力。

3.3.4边界值分析法的测试运用

1．三角形问题的边界值分析测试用例设计

在三角形问题中，除了要求边长是整数，没有给出其他限制条件。假设输入在1～100取值，则边长下界为1，上界为100。

三角形问题的边界值分析测试用例

测试用例	*a*	*b*	*c*	预期输出
Test1	60	60	1	等腰三角形
Test2	60	60	2	等腰三角形
Test3	60	60	60	等边三角形
Test4	50	50	99	等腰三角形
Test5	50	50	100	非三角形
Test6	60	1	60	等腰三角形
Test7	60	2	60	等腰三角形
Test8	50	99	50	等腰三角形
Test9	50	100	50	非三角形
Test10	1	60	60	等腰三角形
Test11	2	60	60	等腰三角形
Test12	99	50	50	等腰三角形
Test13	100	50	50	非三角形

3.3.4 边界值分析法的测试运用

在使用等价类划分进行测试时，还需要注意以下问题

（1）如果实现的语言是强类型语言（无效值输入会引起系统运行时出错），则没有必要使用健壮等价类测试。

（2）如果错误输入检查非常重要，则应进行健壮等价类测试。

（3）如果输入数据以离散区间或集合的形式定义，则等价类测试是合适的，当然也适用于变量值越界会造成故障的系统。

（4）在发现合适的等价关系之前，可能需要许多次的尝试。

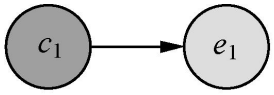
- 3.4 因果图和决策表设计方法
  - 3.4.1 因果图原理
  - 3.4.2 因果图法应用
  - 3.4.3 决策表法及其应用

3.4.1因果图原理

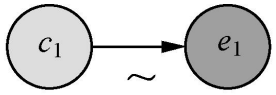
因果图使用简单的逻辑符号，用直线连接左右节点。左节点表示输入状态，是因果图的原因；右节点表示输出状态，是结果。因果图中的4种符号分别表示 4种因果关系。其中，ci表示原因，通常位于图左；ei表示结果，通常位于图右。ci与ei可以取值0或1，0表示状态不出现，1表示状态出现。

4 种因果关系

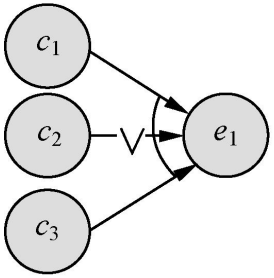
（1）恒等：若c1为1，则e1也为1。
（2）非：若c1为1，则e1为0。
（3）或：c1或c2或c3为1，则e1为1；否则e1为0。或可以有任意个输入。
（4）与：若c1和c2都为1，则e1为1；否则e1为0。与可以有任意个输入。



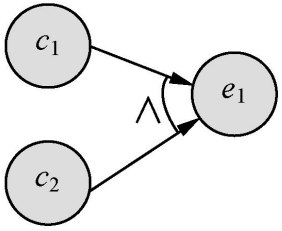
（a）恒等



（b）非



（c）或

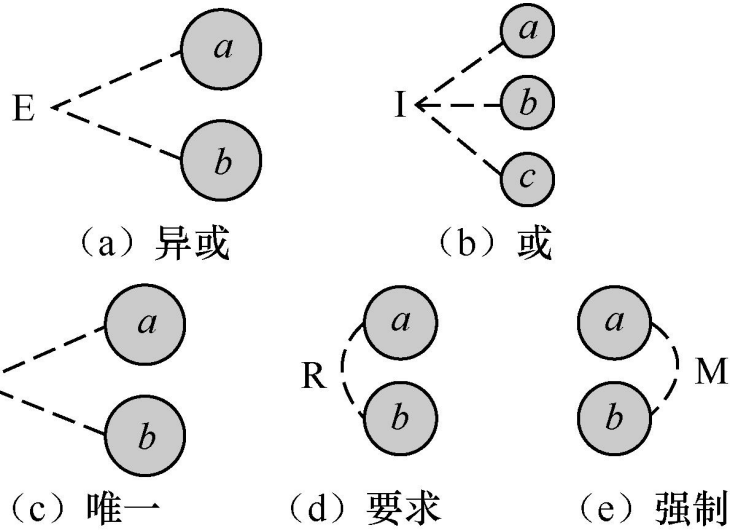


（d）与

3.4.1因果图原理

因果图的约束的类别

（1）异或：a和b最多有一个为1，即a和b不能同时为1。
（2）或：a和b至少有一个为1，即a和b不能同时为0。
（3）唯一：a和b有且仅有一个为1。
（4）要求：a为1时，b必须为1。
（5）强制：强制约束是关于输出条件的约束。若结果a是1，则结果b强制为0。



（a）异或　（b）或　（c）唯一　（d）要求　（e）强制

3.4.1因果图原理

因果图法设计测试用例的步骤

1、根据程序规格说明书识别出输入条件或输入条件的等价类（原因）以及输出条件（结果），并给每个原因和结果赋予一个标识符。

2、找出原因与原因之间、原因与结果之间的对应关系，将其表示成连接各个原因与各个结果之间的“因果图”。

3、由于语法及环境限制，有些原因与原因之间、原因与结果之间的组合情况是不可能出现的，在因果图上用记号表明约束或限制条件。

4、将因果图转换成决策表。

5、根据决策表中每一列设计测试用例。

3.4.2因果图法应用

例： 需要设计这样一个软件，其程序的规格要求如下：
输入的第1个字符必须是“#”或“*”，第2个字符必须是一个数字，在此情况下进行文件的修改；如果第1个字符不是“#”或“*”，则给出信息N；如果第2个字符不是数字，则给出信息M。

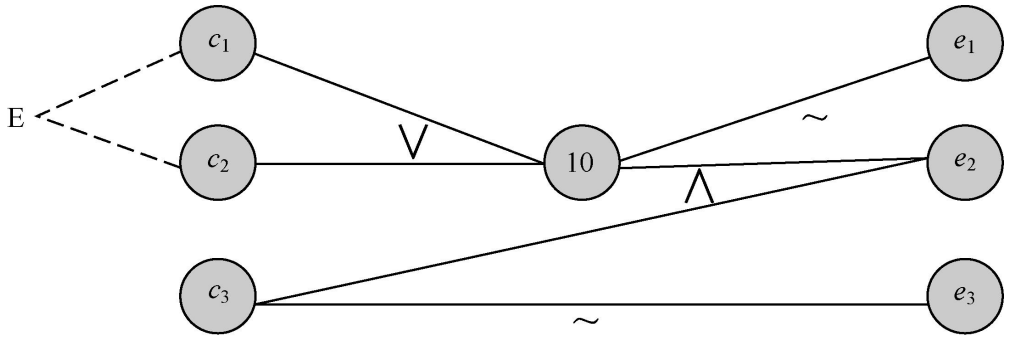
步骤一：对程序设计要求进行分析，分别给出原因和结果，并对每一个原因对应的结果都给定一个标识。

原因	结果
c1：第1个字符是“#”	e1：给出信息“N”
c2：第1个字符是“*”	e2：修改文件
c3：第2个字符是一个数字	e3：给出信息“M”

3.4.2因果图法应用

步骤二：找出原因与原因之间、原因与结果之间的对应关系，将其表示成连接各个原因与各个结果之间的“因果图”。

将得出的原因和结果进行连接，可以得到程序的因果图，如图3-11所示。其中编号为10的中间节点是导出结果的进一步原因。绘制过程中，发现原因c1与c2不可能同时为1，即第一个字符不可能既是“#”又是“*”，所以在因果图上加上E约束。



因果图表示

3.4.2因果图法应用

步骤三：将因果图转换成决策表

		1	2	3	4	5	6	7	8
条件	c_1	1	1	1	1	0	0	0	0
	c_2	1	1	0	0	1	1	0	0
	c_3	1	0	1	0	1	0	1	0
	10			1	1	1	1	0	0
动作	e_1							√	√
	e_2			√		√			
	e_3				√		√		√
	不可能	√	√						

根据因果图建立的决策表

3.4.3决策表法及其应用

1．决策表

决策表（也称为判定表）是分析和表达多逻辑条件下执行不同操作情况的工具。在所有的黑盒测试方法中，基于决策表的测试是最为严格、最具有逻辑性的测试方法。

决策表的4个组成部分

① 条件桩—列出问题的所有条件；
② 条件项—针对条件桩给出的条件列出所有的可能值；
③ 动作桩—列出问题规定的可能采取的操作；
④ 动作项—指出在条件项的各组取值情况下应采取的动作。

动作项和条件项紧密相关，指出在条件项的各组取值情况下应采取的动作。

2．构建决策表的步骤

步骤一：确定规则的个数。一般来说，有n个条件的决策表有2n个规则。

步骤二：列出所有的条件桩和动作桩。

步骤三：填入条件项。

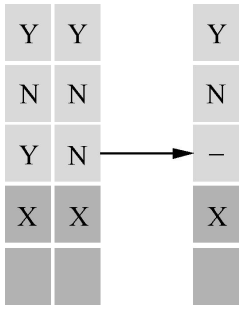
步骤四：填入动作项，得到初始决策表。

步骤五：简化决策表，合并相似规则。

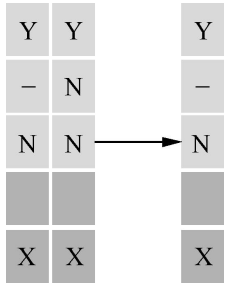
3.4.3决策表法及其应用

当决策表较烦琐时，会对它进行简化。有两条或多条规则具有相同的动作，并且其条件项之间存在着极为相似的关系，就可以将规则合并，合并后的条件项用符号“—”表示

当对规则进行进一步合并时，若需要被合并的条件项分别为“—”和一个任意值，可以直接合并为“—”。



两条规则合并为一条



两条规则进一步合并

3．以三角形问题为例构建决策表

①、列出问题的条件桩为

C1：a、b、c能构成三角形吗？
C2：a=b？
C3：a=c？
C4：b=c？

②、计算这个问题的规则数

共有4个条件，每个条件的取值为“是”或“否”，因此共有24=16条规则。

③、列出动作桩

A1：非三角形
A2：不等边三角形
A3：等腰三角 形
A4：等边三角形
A5：不构成三角形

三角形问题的决策表

		1	2	3	4	5	6	7	8	9
条件	*a*、*b*、*c*能构成三角形吗?	N	Y	Y	Y	Y	Y	Y	Y	Y
	a=*b*?	—	Y	Y	Y	Y	N	N	N	N
	a=*c*?	—	Y	Y	N	N	Y	Y	N	N
	b=*c*?	—	Y	N	Y	N	Y	N	Y	N
动作	非三角形	√								
	不等边三角形									√
	等腰三角形					√		√	√	
	等边三角形		√							
	不构成三角形			√	√		√			

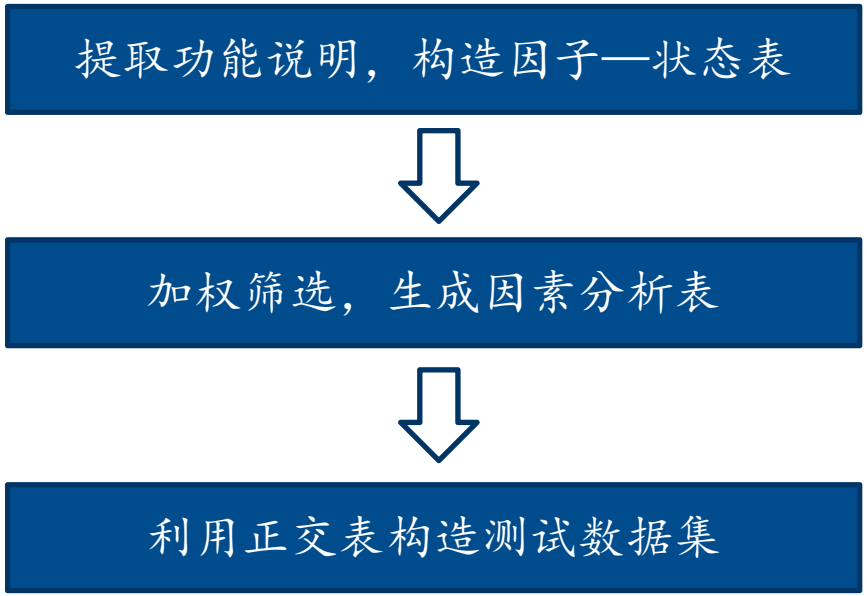
- 3.5 正交试验设计方法
  - 3.5.1 正交试验设计法原理
  - 3.5.2 利用正交试验法设计测试用例

3.5.1正交试验设计法原理

1．正交试验法

正交试验设计方法是依据伽罗瓦（Galois）理论，从大量的试验数据（测试用例）中挑选适量的、有代表性的点（例），从而合理地安排试验（测试）的一种科学试验设计方法。

2．利用正交试验设计测试用例的步骤



3.5.1正交试验设计法原理

3．正交表的构成

（1）行数（Rows）：正交表中行的数量，即试验的次数，也是我们通过正交试验法设计的测试用例的个数。

（2）因素数（Factors）：正交表中列的数量，即我们要测试的功能点。

（3）水平数（Levels）：任何单个因素能够取得的值的最大个数。

正交表示例

	列号							
		1	**2**	**3**	**4**	**5**	**6**	**7**
行号	1	1	1	1	1	1	1	1
	2	1	1	1	0	0	0	0
	3	1	0	0	1	1	0	0
	4	1	0	0	0	0	1	1
	5	0	1	0	1	0	1	0
	6	0	1	0	1	0	1	0
	7	0	0	1	1	0	0	1
	8	0	0	1	0	1	1	0

正交表中包含的值为从0到“水平数-1”或从1到“水平数”，即要测试功能点的输入条件。

正交表一般使用L行数（水平数因素数）来表示，如L8(27)，代表的是8行7列，水平数为2的正交表，如表3-15所示，其中加粗数字为因素，而底纹所在处为几个水平值

4.交互作用

每张正交表后都附有相应的交互作用表，它是专门用来安排交互作用试验的。

安排交互作用的试验时，是将两个因素的交互作用当作一个新的因素，占用一列，为交互作用列，

L8(27)交互作用表

列号	1	2	3	4	5	6	7
	(1)	3	2	5	4	7	6
		(2)	1	6	7	4	5
			(3)	7	6	5	4
				(4)	1	2	3
					(5)	3	2
						(6)	1

交互作用表中可查出L8(27)正交表中的任何两列的交互作用列。表中带“（）”的为主因素的列号，它与另一主因素的交互列为第1个列号从左向右，第2个列号顺次由下向上，二者相交的号为二者的交互作用列。

3.5.1正交试验设计法原理

正交试验的表头设计是正交设计的关键，它承担着将各因素及交互作用合理安排到正交表的各列中的重要任务，正交试验的表头设计的主要步骤如下：

- 确定列数
- 确定各因素的水平数
- 选定正交表
- 表头安排
- 组织实施方案

3.5.2正交试验法设计测试用例

1．用正交表设计测试用例的步骤

（1）有哪些因素（变量）。
（2）每个因素有哪几个水平（变量的取值）。
（3）选择一个合适的正交表。
（4）把正交表中变量的值映射到表中。
（5）把正交表中每行的各因素水平的组合作为一个测试用例。
（6）加上可疑且没有在正交表中出现的组合。

2．如何选择正交表

（1）考虑因素（变量）的个数。
（2）考虑因素水平（变量的取值）的个数。
（3）考虑正交表的行数。
（4）取行数最少的一个。

3．选择正交表的基本原则

（1）先看水平数。若各因素全是二水平，就选用L(2*)表；若各因素全是三水平，就选L(3*)表。若各因素的水平数不相同，就选择适用的混合水平表。

（2）每一个交互作用在正交表中应占1列或2列。

（3）要看试验精度的要求。若要求高，则宜取试验次数多的L表。

（4）若试验费用很昂贵、试验的经费很有限或人力和时间都比较紧张，则不宜选试验次数太多的L表。

（5）按原来考虑的因素、水平和交互作用去选择正交表，若无正好适用的正交表可选，简便且可行的办法是适当修改原定的水平数。

（6）对某因素或某交互作用的影响是否确实存在没有把握的情况下，选择L表时常为该选大表还是选小表而犹豫。若条件许可，应尽量选用大表，让影响存在的可能性较大的因素和交互作用各占适当的列。

- 3.6 组合测试方法
  - 3.6.1 基本概念
  - 3.6.2 构造方法
  - 3.6.3 组合测试工具的使用

正交试验设计方法存在的问题

- 构造正交表需要输入参数的可能取值数保持一致，并且输入参数的个数与参数的取值数需要满足一定的条件。实际测试时，抽取的输入参数和参数的可能取值很难满足该条件

- 测试人员基于自身的测试经验，可能希望部分参数之间实现更高强度的组合覆盖或部分参数之间的约束关系，正交试验设计方法难以满足。

3.6.1组合测试基本概念

组合测试（Combinatorial Testing）方法则可以有效地解决上述问题， 组合测试步骤如下:

- 对被测项目进行分析，借助之前介绍的等价类划分和边界值分析等方
- 法，识别出被测项目的输入参数，并依次确定每个参数的可能取值。
- 确定组合测试要使用的t-way组合覆盖准则。
- 将可以覆盖t-way组合覆盖准则的测试用例称为组合测试用例集。

3.6.1组合测试基本概念

假设需要对部署的系统做兼容性测试

- 识别出影响到系统兼容性的4个因素，分别是数据库服务器、Web 服务器、浏览器和路由器。
- 依次分析出每个因素的可能取值

系统兼容性测试的输入

数据库服务器	Web服务器	浏览器	路由器
Oracle	Apache	IE	Cisco
SQL Server	JBoss	Firefox	Huawei
MySQL	WebSphere	Chrome	Alcatel

- 得到最优成对组合测试用例集

数据库服务器	Web服务器	浏览器	路由器
Oracle	Apache	IE	Cisco
Oracle	JBoss	Firefox	Huawei
Oracle	WebSphere	Chrome	Alcatel
SQL Server	Apache	Firefox	Alcatel
SQL Server	WebSphere	IE	Huawei
SQL Server	JBoss	Chrome	Cisco
MySQL	Apache	Chrome	Huawei
MySQL	JBoss	IE	Alcatel
MySQL	WebSphere	Firefox	Cisco

组合测试的优点

- 组合测试通过仅关注部分参数间的组合覆盖可以有效减小构造出的测试用例集规模；
- 组合测试技术可以有效地检测出由于参数间交互所触发的各种软件缺陷；
- 部分实证研究也表明，在一些测试场景下，组合测试技术可以用小规模的测试用例集完成高质量的软件测试；

组合测试的缺点

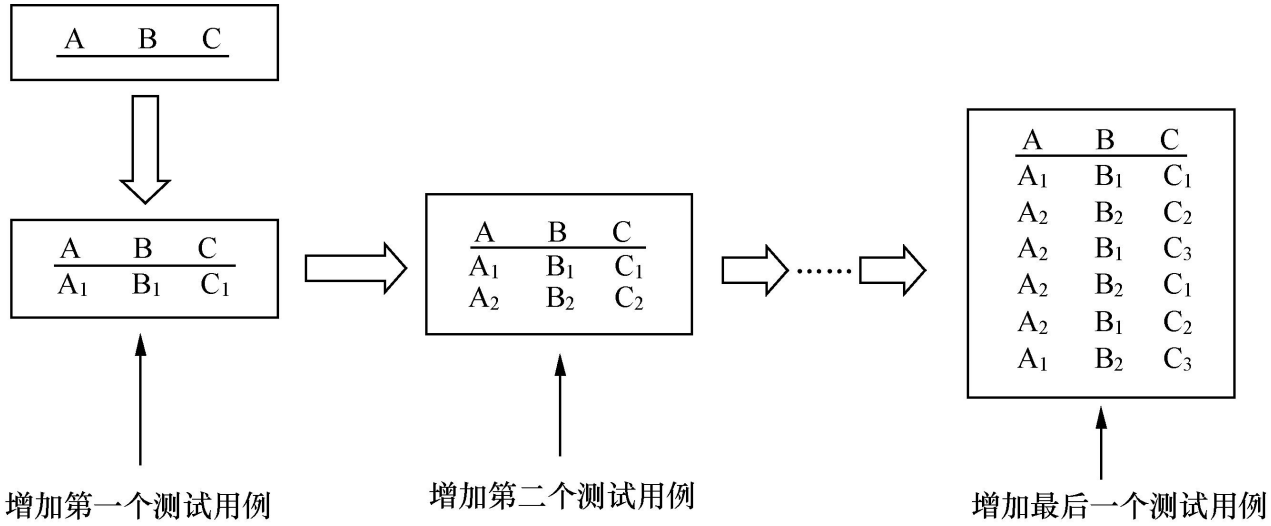
- 被测程序的测试用例设计不能仅仅依靠组合测试技术，因为组合测试仅是一种基于规约的测试用例生成技术，要设计出充分性更高的测试用例集需要结合使用其他软件测试技术；
- 组合测试仅仅关注测试用例输入的构造，但测试用例预言仍需要采用手工方式进行构造，该不足使得组合测试很难实现自动化；
- 组合测试的软件缺陷检测效果与测试人员分析出的输入因素和因素取值构成密切相关；

3.6.2构造方法

构造最优组合测试用例集问题是一个NP难问题，目前主要基于贪心算法和进化算法（如遗传算法、模拟退火等）来构造组合测试用例集。

假设被测项目包含3个参数：A、B和C。其中参数A包括两个参数取值A1和A2，参数B包括两个参数取值B1和B2，参数C包括3个参数取值C1、C2和C3。

一次生成一个测试用例（one test at a time）的策略



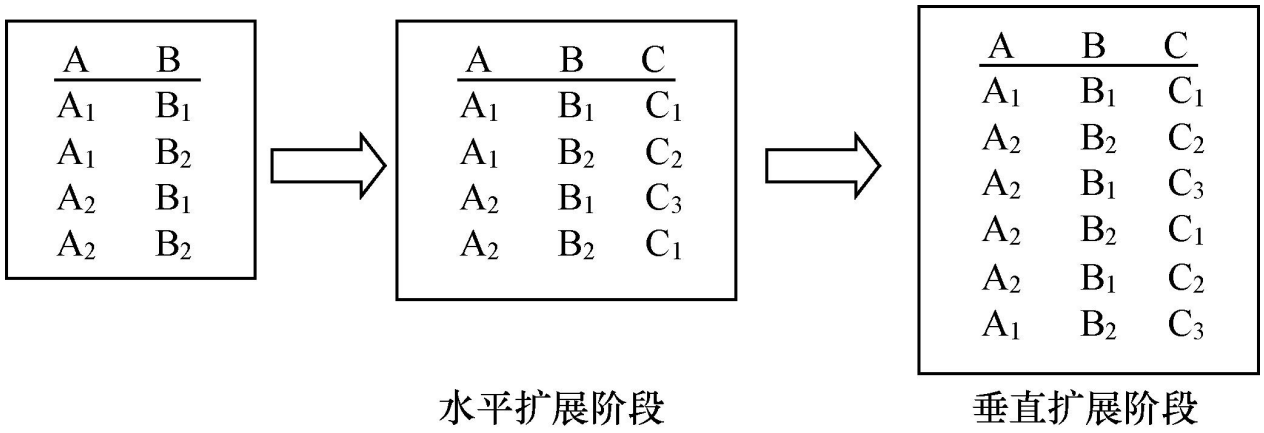
IPO策略

该策略在每次迭代时包含两个阶段：水平扩展阶段和垂直扩展阶段。

首先生成测试用例满足前两个参数A和B的两两组合覆盖

随后扩展至第三个参数C，先进入垂直扩展阶段，

当填充完第4个测试用例的取值后，如果还有前3个参数尚未覆盖的两两组合集合Uncover，则进入水平扩展阶段



水平扩展阶段　　垂直扩展阶段

3.6.3组合测试工具的使用

目前使用较多的组合测试工具是微软开发的PICT工具。该工具可以指定需要使用的组合覆盖准则，并可以支持组合之间约束处理。

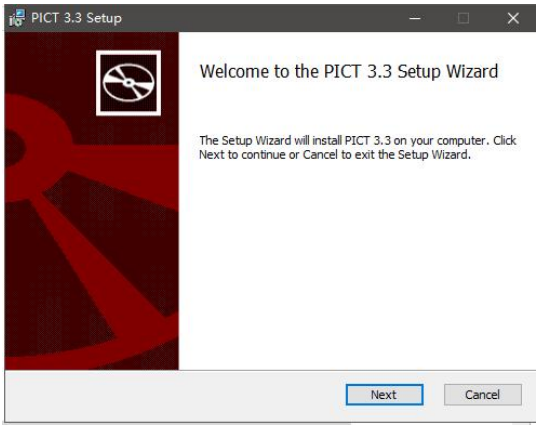
1．PICT简介

成对独立组合测试（Pairwise Independent Combinatorial Testing，PICT）会根据输入描述，自动生成满足条件的组合测试用例集，从而提高测试效率。

2．PICT安装

（1）通过官网可以下载到PICT的安装包，双击“pict33.msi”安装程序，进入图3-17所示的安装界面。

（2）一路单击“Next”按钮，直到可以给PICT设置安装路径后继续单击“Next”按钮，接下来的界面会出现“Install”按钮，单击“Install”按钮进行安装，再单击“Finish”按钮即可完成安装。



3．实例讲解

（1）准备以下一个输入实例。

操作系统包括Windows 7、Windows 8、Windows 8.1、Windows 10。

打印机的型号有HP 2621、HP 3636、HP 2132。

打印模式有黑白、彩色两种。

按钮包括确定和取消。

（2）在安装PICT的目录中新建一个TXT文件并把上面的实例输入进去，这里我们将输入文件命名为“test.txt”。

（3）使用cmd命令进入test文件所在的目录中

（4）使用命令“pict test.txt”，

（5）如果我们想把结果保存为Excel文档并使用Excel进行后续操作，我们可以将输入流指向一个Excel文件，使用的命令为“pict test.txt>output_test.xls”。

本章小结

- 测试用例综述
- 等价类设计方法
- 边界值设计方法
- 因果图和决策表设计方法
- 正交试验设计方法
- 组合测试方法