

第三部分 黑盒测试

§ 1 黑盒测试概述

§ 2 等价类划分法

§ 3 边界值分析法

§ 4 判定表法

§ 5 因果图

§ 6 域测试模型



§ 1 黑盒测试概述



■ 一、黑盒测试

■ 黑盒测试顾名思义就是将被测系统看成一个黑盒，从外界取得输入，然后再输出。整个测试基于需求文档，看是否能满足需求文档中的所有要求。黑盒测试要求测试者在测试时不能使用与被测系统内部结构相关的知识或经验，它适用于对系统的功能进行测试。



§ 1 黑盒测试概述



- 黑盒测试的优点有：
 - 1) 比较简单，不需要了解程序内部的代码及实现；
 - 2) 与软件的内部实现无关；
 - 3) 从用户角度出发，能很容易的知道用户会用到哪些功能，会遇到哪些问题；
 - 4) 基于软件开发文档，所以也能知道软件实现了文档中的哪些功能；
 - 5) 在做软件自动化测试时较为方便。



§ 1 黑盒测试概述



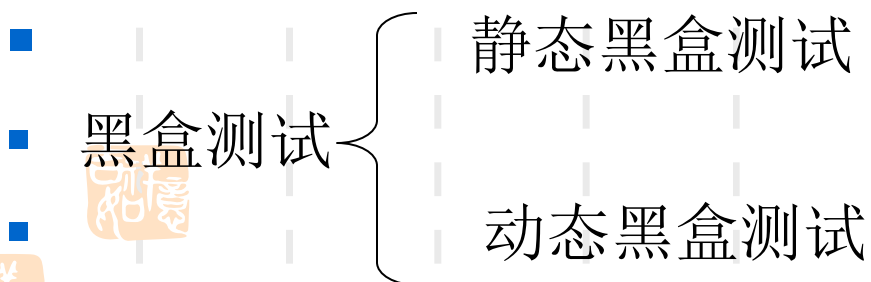
- 黑盒测试的缺点有：
 - 1) 不可能覆盖所有的代码，覆盖率较低，大概只能达到总代码量的30%；
 - 2) 自动化测试的复用性较低。



§ 1 黑盒测试概述

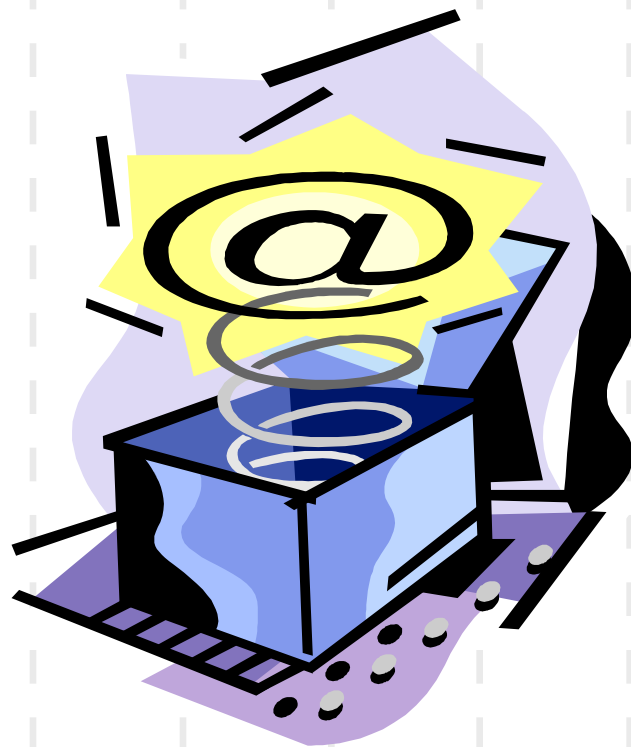
吉祥如意

■ 二、静态与动态黑盒测试



静态测试：只是检查和审阅

动态测试：运行和使用软件



吉祥如意

吉祥如意

吉祥如意

§ 1 黑盒测试概述

吉祥如意

三、静态黑盒测试

对产品说明书进行高级审查

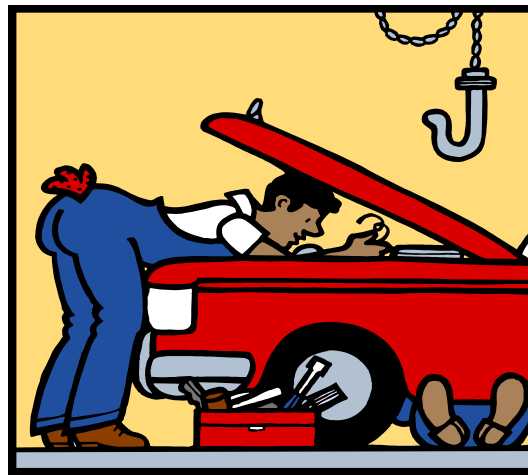
- 设身处地为用户着想
- 研究现有的标准和规范
- 审查和测试同类软件

产品描述属性检查

产品描述用语检查

1、测试标准

- 公司惯用语和约定
- 行业要求
- 国家标准
- 硬件和网络标准
- 图形用户界面（GUI）



§ 1 黑盒测试概述

吉祥如意

2、产品描述属性检查清单

- 完整
- 准确
- 精确
- 一致
- 合理
- 代码无关
- 可测试



吉祥如意

吉祥如意

吉祥如意

吉祥如意

§ 1 黑盒测试概述



3、产品描述用语检查清单

- 是否有绝对、肯定和切实认定的叙述，针对其设计用例。
- 比较模糊的用语。某些、有时……
- 功能清单是否有等等、诸如此类、依此类推，无法测试的词汇。
- 在性能上不确定的说法。
- 隐藏大量需要说明的功能
- 如果……那么……（没有否则）。



§ 2 等价类划分法



- 一、基本概念

- 1、等价类划分法

- 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 等价类划分法



- 2、划分等价类

- 等价类是指某个输入域的子集合.在该子集合中,各个输入数据对于揭露程序中的错误都是等效的。

- 并合理地假定:测试某等价类的代表值就等于对这一类其它值的测试。因此,可以把全部输入数据合理划分为若干等价类,在每一个等价类中取一个数据作为测试的输入条件,就可以用少量代表性的测试数据,取得较好的测试结果。

在该子集合中, 各个输入数据对于揭露程序中的错误都是等效的。



§ 2 等价类划分法



3、有效等价类和无效等价类

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**：是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：是指对于程序的规格说明来说，是不合理的，无意义的输入数据构成的集合。

■ 设计测试用例时，要同时考虑这两种等价类。因为，软件不仅要能接收合理的数据，也要能经受意外的考验。这样的测试才能确保软件具有更高的可靠性。



§ 2 等价类划分法



■ 4、等价类表

- 在确立了等价类后,可建立等价类表,列出所有划分出的等价类:

输入条件	有效等价类	无效等价类
------	-------	-------

...

...

...

...

...

...

注：实际表格比此复杂，见实际例子

§ 2 等价类划分法

- 二、等价类的划分原则(根据输入输出条件)
- ①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。
- ②在输入条件规定了输入值的集合或者规定了“必须如何”的条件的情況下,可确立一个有效等价类和一个无效等价类。
- ③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。

§ 2 等价类划分法



- ④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。
- ⑤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）。
- ⑥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 等价类划分法



- 如：
- 1、数据取值是从1-99（一个有效等价类，两个无效等价类）
- 2、变量名义字母开头（一个有效等价类、一个无效等价类）
- 3、条件 $A > 100$ （一个有效等价类、一个无效等价类）
-
.....



§ 1 等价类划分法



- 三、举例

- 1、三角形的例子

- 某程序规定：“输入三个整数作为三边的边长构成三角形。当此三角形为一般三角形、等腰三角形及等边三角形时，分别做计算...”。试用等价类划分方法为该程序的构成三角形部分进行测试用例设计。

- 请同学们根据上面的划分原则设计测试用例

§ 1 等价类划分法

吉祥如意

表 4.1 例 1 的等价类表

		有效等价类	号码	无效等价类	号码
输入条件	整数	整数	1	$\begin{cases} a \text{ 为非整数} \\ b \text{ 为非整数} \\ c \text{ 为非整数} \end{cases}$	12
				$\begin{cases} a, b \text{ 为非整数} \\ b, c \text{ 为非整数} \\ a, c \text{ 为非整数} \end{cases}$	13
				三边 a, b, c 均为非整数	14
	三个数	三个数	2	$\begin{cases} \text{只给 } a \\ \text{只给 } b \\ \text{只给 } c \end{cases}$	15
				$\begin{cases} \text{只给 } a, b \\ \text{只给 } b, c \\ \text{只给 } a, c \end{cases}$	16
				给出三个以上	17
	非零数	非零数	3	$\begin{cases} a \text{ 为 } 0 \\ b \text{ 为 } 0 \\ c \text{ 为 } 0 \end{cases}$	18
				$\begin{cases} a, b \text{ 为 } 0 \\ b, c \text{ 为 } 0 \\ a, c \text{ 为 } 0 \end{cases}$	19
				三边 a, b, c 均为 0	20
	正数	正数	4	$\begin{cases} a < 0 \\ b < 0 \\ c < 0 \end{cases}$	21
				$\begin{cases} a < 0 \text{ 且 } b < 0 \\ a < 0 \text{ 且 } c < 0 \\ b < 0 \text{ 且 } c < 0 \end{cases}$	22
				三边均 < 0 : $a < 0$ 且 $b < 0$ 且 $c < 0$	23
输出条件	构成一般三角形	$a + b > c$	5	$\begin{cases} a + b < c \\ a + b = c \end{cases}$	24
		$b + c > a$	6	$\begin{cases} b + c < a \\ b + c = a \end{cases}$	25
		$a + c > b$	7	$\begin{cases} a + c < b \\ a + c = b \end{cases}$	26
	构成等腰三角形	$a = b$	8		27
		$b = c$	9		28
		$a = c$	10		29
	构成等边三角形	$a = b = c$	11		30
					31

吉祥如意

吉祥如意

吉祥如意

吉祥如意

吉祥如意

吉祥如意

§ 1 等价类划分法

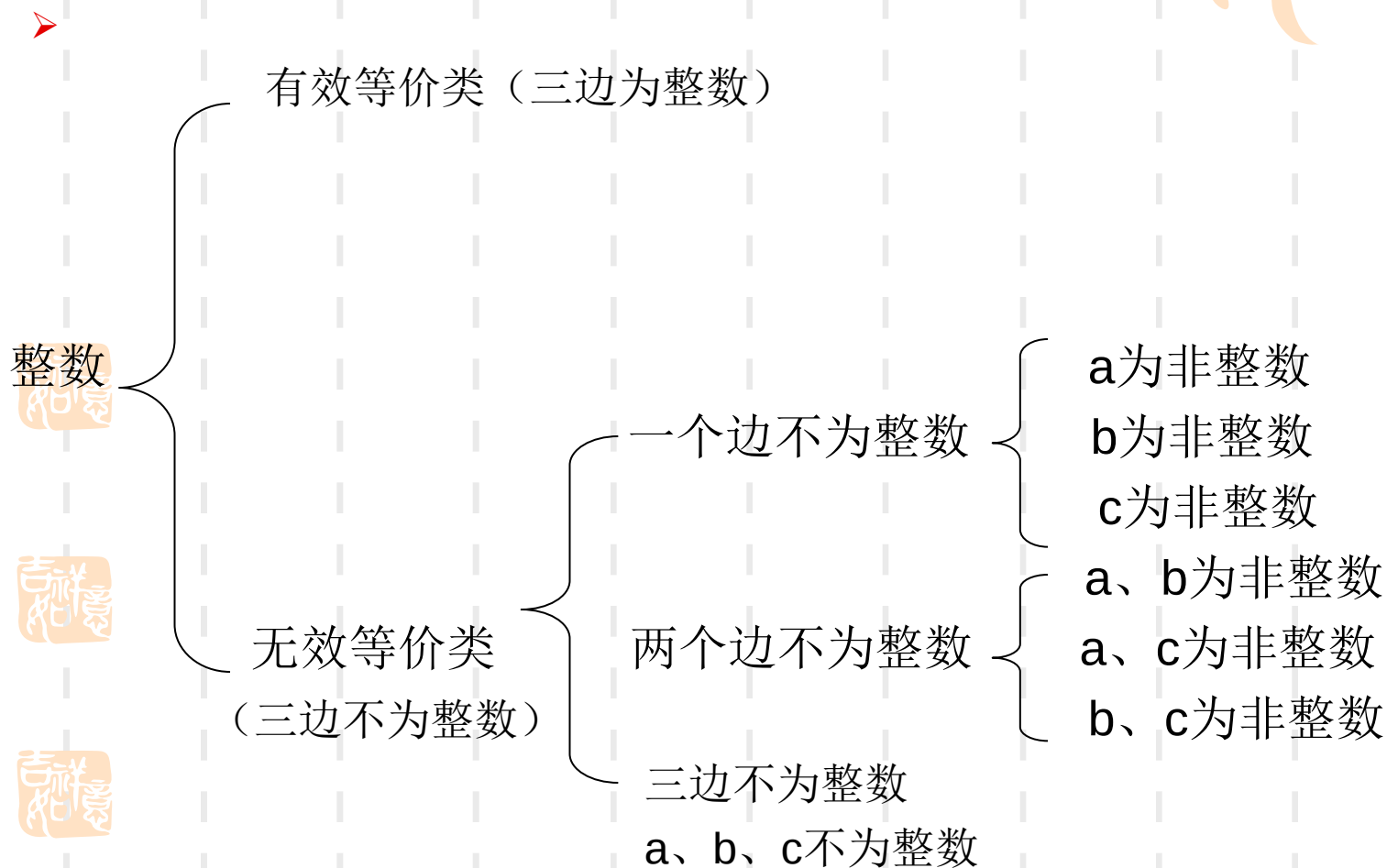


■ 分析:

- (使用等价类划分方法必须仔细分析和推敲题目所给出的要求。)
- 本题的输入条件要求的关键之处有：1) 整数；2) 三个数；3) 非零数；4) 正数。
- 输出条件要求的关键之处有：5) 应满足两边长之和大于第三边边长；6) 等腰；7) 等边。
- 其中，3)、4) 和5) 并没有在题目上明显给出，但这些条件是必要的。



§ 2 等价类划分法



§ 2 等价类划分法



2、电话号码

某城市电话号码由三部分组成。它们的名称和内容分别是：

地区码：空白或三位数字；

前 缀：非 ‘0’或 ‘1’的三位数字；

后 缀：4位数字。

假定被测程序能接受一切符合上述规定的电话号码，拒绝所有不符合规定的电话号码。根据该程序的规格说明，作等价类的划分，并设计测试方案。



§ 2 等价类划分法

输入条件	有效等价类	无效等价类
地区码	1.空白; 2.三位数;	5.有非数字字符; 6.少于三位数字; 7.多于三位数字。
前 缀	3.从200到999之间的三位	8.有非数字字符; 9.起始位为‘0’; 10.起始位为‘1’; 11.少于三位数字; 12.多于三位数字。
后 缀	4.四位数字。	13.有非数字字符; 14.少于四位数字; 15.多于四位数字。

§ 2 等价类划分法



四、测试用例设计

- 从划分出的等价类中按以下三个原则设计测试用例：
 - ①为每一个等价类规定一个唯一的编号。
- ②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。
- ③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 等价类划分法



- 例：1、三角形的测试用例
- 以下分两步进行：
- （1）列出等价类表（表中号码为等价类编号）
- （2）列出覆盖上述等价类的测试用例：

	a	b	c	覆盖等价类号码
■				
■				
■	3	4	5	(1) -- (7)
■	4	4	5	(1) -- (7) , (8)
■	4	5	5	(1) -- (7) , (9)
■	5	4	5	(1) -- (7) , (10)
■	4	4	4	(1) -- (7) , (11)



§ 2 等价类划分法

覆盖无效等价类的测试用例

表 4.2 覆盖无效等价类的测试用例

a	b	c	覆盖等价类号码	a	b	c	覆盖等价类号码
2.5	4	5	12	0	0	5	29
3	4.5	5	13	3	0	0	30
3	4	5.5	14	0	4	0	31
				0	0	0	32
3.5	4.5	5	15				
3	4.5	5.5	16	-3	4	5	33
3.5	4	5.5	17	3	-4	5	34
3.5	4.5	5.5	18	3	4	-5	35
3			19	-3	-4	5	36
	4		20	-3	4	-5	37
		5	21	3	-4	-5	38
				-3	-4	-5	39
3	4		22				
	4	5	23	3	1	5	40
3		5	24	3	2	5	41
3	4	5	25	3	1	1	42
				3	2	1	43
0	4	5	26				
3	0	5	27	1	4	2	44
3	4	0	28	3	4	1	45

2、电话号码的测试用例

方案	内容			输 入	预期输出
	地区码	前缀	后缀		
1	空白（1）	200～999之间的三位数字（3）	四位数字（4）	()276-2345	有效
2	三位数字（2）		四位数字（4）	(635)805-9321	有效
3	有非数字字符（5）			(20A)723-4567	无效
4	少于三位数字（6）			(33)234-5678	无效
5	多于三位数（7）			(5555)345-6789	无效
6		有非数字字符（8）		(345)5A2-3456	无效
7		起始位为‘0’（9）		(345)012-3456	无效
8		起始位为‘1’（10）		(345)132-3456	无效
9		少于三位数字（11）		(345) 92-3456	无效
10		多于三位数字（12）		(345)4562-3456	无效
11			有非数字字符（13）	(345)342-3A56	无效
12			少于四位数字（14）	345)342- 356	无效
13			多于四位数字（15）	(345)562-34567	无效

§ 3 边界值分析法



一、边界值分析方法的考虑:

长期的测试工作经验告诉我们,大量的错误是发生在输入或输出范围的边界上,而不是发生在输入输出范围的内部.因此针对各种边界情况设计测试用例,可以查出更多的错误。

■ 使用边界值分析方法设计测试用例,首先应确定边界情况.通常输入和输出等价类的边界,就是应着重测试的边界情况.应当选取正好等于,刚刚大于或刚刚小于边界的值作为测试数据,而不是选取等价类中的典型值或任意值作为测试数据。

■ 边界值分析方法是等价类划分方法的补充。



§ 3 边界值分析法

- 比如，在作三角形计算时，要输入三角形的三个边长：**A**、**B**和**C**。我们应该注意到这三个数值应当满足 $A+B>C$ 、 $A+C>B$ 及 $B+C>A$ 才能构成三角形。但如果把三个不等式的任何一个大于号）错写成大于等于号 $\geq$ ，那就无法构成三角形。问题恰恰出现在那些容易被疏忽的边界上。这里所说的边界是指，相对于输入等价类和输出等价类而言，稍高于其边界值及稍低于其边界值的一些特定情况。

§ 3 边界值分析法



■ 二、遵循的原则

- 针对边界值设计测试用例时，应遵循以下几条原则：

- (1) 如果输入条件规定了取值范围，或是规定了值的个数，应以该范围的边界内及刚刚超出范围的边界外的值，或是分别对最大、最小个数及稍小于最小、稍大于最大个数作为测试用例。例如，如果程序的规格说明中规定：“重量在10公斤至50公斤范围内的邮件，其邮费计算公式为……”。作为测试用例，我们应取10及50，还应取10.01，49.99，9.99及50.01等。如果另一问题规格说明规定：“某输入文件可包含1至255个记录，……”，则测试用例可取1和255，还应取0及256等。



§ 3 边界值分析法

- ②针对规格说明的每个输出条件使用前面的第（1）条原则。例如，某程序的规格说明要求计算出“每月保险金扣除额为0至1165.25元”，其测试用例可取0.00及1165.2、还可取一0.01及1165.26等。如果另一程序属于情报检索系统，要求每次”最多显示1条情报摘要”，这时我们应考虑测试用例包括1和4，还应包括0和5等。

§ 3 边界值分析法



- ③如果程序规格说明中提到的输入或输出域是个有序的集合（如顺序文件、表格等），就应注意选取有序集的第一个和最后一个元素作为测试用例。
- ④分析规格说明，找出其它的可能边界条件。



§ 3 边界值分析法

- 另一种看起来很明显的软件缺陷来源是当软件要求输入时(比如在文本框中),不是没有输入正确的信息,而是根本没有输入任何内容,单单按了Enter键。这种情况在产品说明书中常常忽视,程序员也可能经常遗忘,但是在实际使用中却时有发生。程序员总会习惯性的认为用户要么输入信息,不管是看起来合法的或非法的信息,要不就会选择Cancel键放弃输入,如果没有对空值进行好的处理的话,恐怕程序员自己都不知道程序会引向何方。
- 正确的软件通常应该将输入内容默认为合法边界内的最小值或者合法区间内某个合理值,否则返回错误提示信息。

§ 3 边界值分析法



■ 三、边界值设计测试用例的例子

■ 1、规格说明

■ 某一为学生考试试卷评分和成绩统计的程序，其规格说明有如下的要求：

■ 程序的输入文件（标准答卷、学生的答卷）由一些相关记录组成，每个纪录为80个字符。这些记录分成三组：

■ ① 标题纪录

■ 这组纪录只有一个记录，其内容为输出报告的名字。

■ ② 试卷各题的标准答案纪录

■ 该组纪录的每一个记录都在第80个字符处标以数字“2”；

■ 该组纪录的第一个记录的第1到3个字符为所出题目的个数（取值为1-999）；

■ 该组纪录的第一个记录的第10到59个字符为第1至第50题的标准答案（每个合法字符表示一个答案）；



- 如果题目数超过50个，则相应应有第二个、第三个记录.....，该组纪录的第二个记录、第三个记录.....相应的第10到50个字符分别为第51题到第100题，第101题到第150题.....的标准答案。

- ③ 每个学生的答案纪录

- 该组纪录的每一个记录都在第80个字符处标以数字“3”；

- 每个学生的答卷由若干个记录组成，每个学生答卷的第一个记录的第1到第9个字符给出学生的姓名及学号，第10至第59字符是学生所做的第1到第50题的答案；

- 若试题数超过50个，则每个学生的答卷纪录的第2、第3个.....纪录分别给出该学生的第51至第100，第101至第150题的解答；

- 一个学生的答卷记录全部给出以后，接着是另一个学生的答题纪录。

- ④ 学生数

- 学生数最多不超过200人。

§ 3 边界值分析法



- 2、输出要求

- 该程序给出四个输出报告：

- ① 按学生学号排序，每个学生的成绩和成绩等级报告；

- ② 按学生得分排序，每个学生的成绩；

- ③ 平均分数，最高分和最低分之差；

- ④ 按题号排序，每题学生答对的百分比。



§ 3 边界值分析法



3、输入文件（记录）的结构

第一组纪录

1	80
标 题	

第二组纪录

12345678910	59 60	79 80
	第 1 到第 50 题的标准答案	2
题目数		
12345678910	59 60	79 80
	第 51 到第 100 题的标准答案	2
12345678910	59 60	79 80
	第 101 到第 150 题的标准答案	2



§ 3 边界值分析法

第三组纪录

12345678910	59 60	79 80
第一个姓名、学号	第 1 到第 50 题的答案	3
12345678910	59 60	79 80
	第 51 到第 100 题的答案	3
12345678910	59 60	79 80
	第 101 到第 150 题的答案	3

12345678910	59 60	79 80
第二个姓名、学号	第 1 到第 50 题的答案	3
12345678910	59 60	79 80
	第 51 到第 100 题的答案	3
12345678910	59 60	79 80
	第 101 到第 150 题的答案	3

12345678910	59 60	79 80
第三个姓名、学号	第 1 到第 50 题的答案	3
12345678910	59 60	79 80
	第 51 到第 100 题的答案	3
12345678910	59 60	79 80
	第 101 到第 150 题的答案	3

§ 3 边界值分析法



- 输入条件分析
 - (1) 输入文件
 - (2) 标题
 - (3) 出题个数
 - (4) 答案记录
 - (5) 学生人数
 - (6) 学生答题



§ 3边界值分析法

按边界值设计测试用例说明：

如：标题	边界情况	{	无标题纪录	输入条件
			标题只有一个字符	
			标题有80个字符	

输入文件 边界情况 输入空文件

§ 3 边界值分析法

吉祥如意

答案纪录

标题纪录后没有标准答案纪录
标准答案纪录多一个
标准答案纪录少一个

出题个数

出了1题
出了50题
出了51题
出了100题
出了999题
没有出题
题目数是非数值量

§ 3 边界值分析法

四、测试用例设计

以下分别针对输入条件、输出条件，根据其边界值设计测试用例

（1）根据输入条件设计测试用例

序号	输入条件	测试用例设计
1	输入文件	空输入文件
2	标题	无标题纪录 只有 1 个字符的标题 具有 80 个字符的标题纪录
3	出题个数	出了 1 题 出了 50 题 出了 51 题 出了 100 题 出了 999 题 没有出题 题目数是非数值量
4	答案纪录	标题纪录后没有标准答案纪录 标准答案纪录多一个 标准答案纪录少一个
5	学生人数	学生人数为 0 学生人数为 1 学生人数为 200 学生人数为 201
6	学生答题	标准答案记录组有 2 个纪录，但某学生只有 1 个答案纪录 该学生是文件中的第 1 个学生 该学生是文件中的最后一个学生 标准答案记录组只有 1 个纪录，但某学生有 2 个答案纪录 该学生是文件中的第 1 个学生 该学生是文件中的最后一个学生

§ 3 边界值分析法



- 输出条件分析

(1) 学生的得分情况

(2) 排序时的学号情况

(3) 输出平均成绩时的平均分数的情况

(4) 答题情况



§ 3 边界值分析法



输出条件

学生得分

- 所有学生得分相同
- 所有学生得分都不相同
- 一些学生（不是全部）得分相同
- 一个学生的0分
- 一个学生的100分



§ 3 边界值分析法

(2) 根据输出条件设计测试用例

序号	输出条件	测试用例设计
1	学生得分	所有学生得分相同 所有学生得分都不相同 些学生 (不是全部) 得分相同 一个学生的 0 分 一个学生的 100 分
2	输出报告 ① ②	一个学生的学号最小 (检查排序) 一个学生的学号最大 学生数正好使输出报告打印满 1 页 学生数使输出报告在 1 页上打印不下, 尚多 1 名同学没有打印
3	输出报告 ③	平均值取最大值 (所有学生都满分) 平均值为 0 分 (所有学生都 0 分) 标准偏差取最大 (一个学生得 100 分, 一个学生的 0 分) 标准偏差取 0 (所有学生得分相同)
4	输出报告 ④	所有学生都答对第 1 题 所有学生都答错第 1 题 所有学生都答对最后 1 题 所有学生都答错最后 1 题 报告打印完 1 页后, 还剩 1 题没有打印 报告在 1 页上恰好打印完



作业1：有三个无符号整型数 a 、 b 、 c ，由键盘输入，输出其中最大数。试分别用等价类划分和边界值分析的方法设计测试用例。



§ 3 判定表法



- 一、什么是判定表

- 在一些数据处理问题中，某些操作是否实施依赖于多个逻辑条件的取值。也即在这些逻辑条件取值的组合所构成的多种情况下，分别执行不同的操作。处理这类问题的一个非常有力的分析和表达工具是判定表（Decision Table）。



§ 3 判定表法



- 让我们首先看一个简单的实例，用以说明什么是判定表。
- 在翻开一本技术书的几页目录之后，读者看到一张表，名为“本书阅读指南”。表的内容给读者指明了在读书过程中可能遇到的种种情况，以及作者针对各种情况给读者的建议。表中列举了读者读书时可能遇到的三个问题，若读者的回答是肯定的（判定取真值），标以字母“Y”；若回答是否定的（判定取假值），标以字母“N”。三个判定条件，其取值的组合共有8种情况。该表为读者提供了4条建议，但并不需要每种情况都施行。这里把要实施的建议在相应栏内标以“X”，其它建议相应的栏内什么也不标。例如，表中的第3种情况，当读者已经疲劳，对内容又不感兴趣，并且还没读懂，这时作者建议去休息。



§ 3 判定表法

		1	2	3	4	5	6	7	8
问题	你觉得疲倦吗?	Y	Y	Y	Y	N	N	N	N
	你对内容感兴趣吗?	Y	Y	N	N	Y	Y	N	N
	书中的内容使你糊涂吗?	Y	N	Y	N	Y	N	Y	N
建议	请回到本章开头重读	×				×			
	继续读下去		×				×		
	跳到下一章去读							×	×
	停止阅读,请休息			×	×				

■ 图3.1 读书指南判定表

早在程序设计发展的初期,判定表就已被当作编写程序的辅助工具使用了。由于它可以把复杂的逻辑关系和多种条件组合的情况表达得既明确又具体,因而给编写者、检查者和读者均带来很大方便。

§ 3 判定表法

判定表通常由4个部分组成，如下图（3.2）所示，用双线分割的四个部分是：

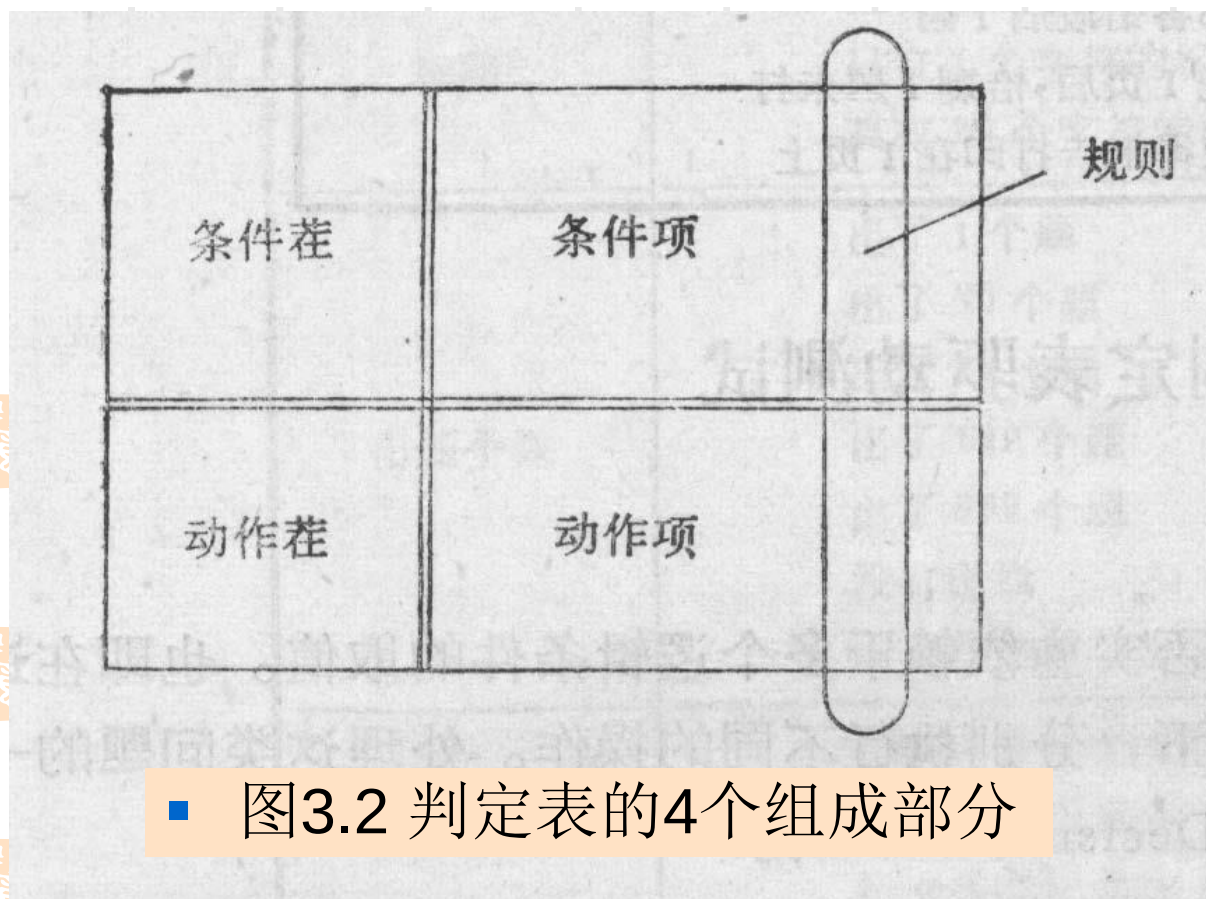
条件茬（Condition Stub）

动作茬（Action Stub）

条件项（Condition Entry）

动作项（Action Entry）

§ 3 判定表法



■ 图3.2 判定表的4个组成部分

§ 3 判定表法

■ 条件荏部分列出了问题的所有条件，除了某些问题对条件的先后次序有特定的要求外，通常在这里列出的条件，其先后次序无关紧要。条件项部分是针对它左列条件的取值，在所有可能情况下，给出真假值。

■ 动作荏则列出了问题规定可能采取的操作。这些操作的排列顺序一般并没有什么约束，但为了便于阅读也可令其按适当的顺序排列。

§ 3 判定表法

■ 动作项和条件项紧密相关。它指出了在条件项的各组取值情况下应采取的动作。我们把任何一个条件组合的特定取值及其相应要执行的操作称为规则。在判定表中贯穿条件项和动作项的一列就是一条规则。显然，判定表中列出多少组条件取值，也就有多少条规则，即条件项和动作项有多少列。

§ 3 判定表法



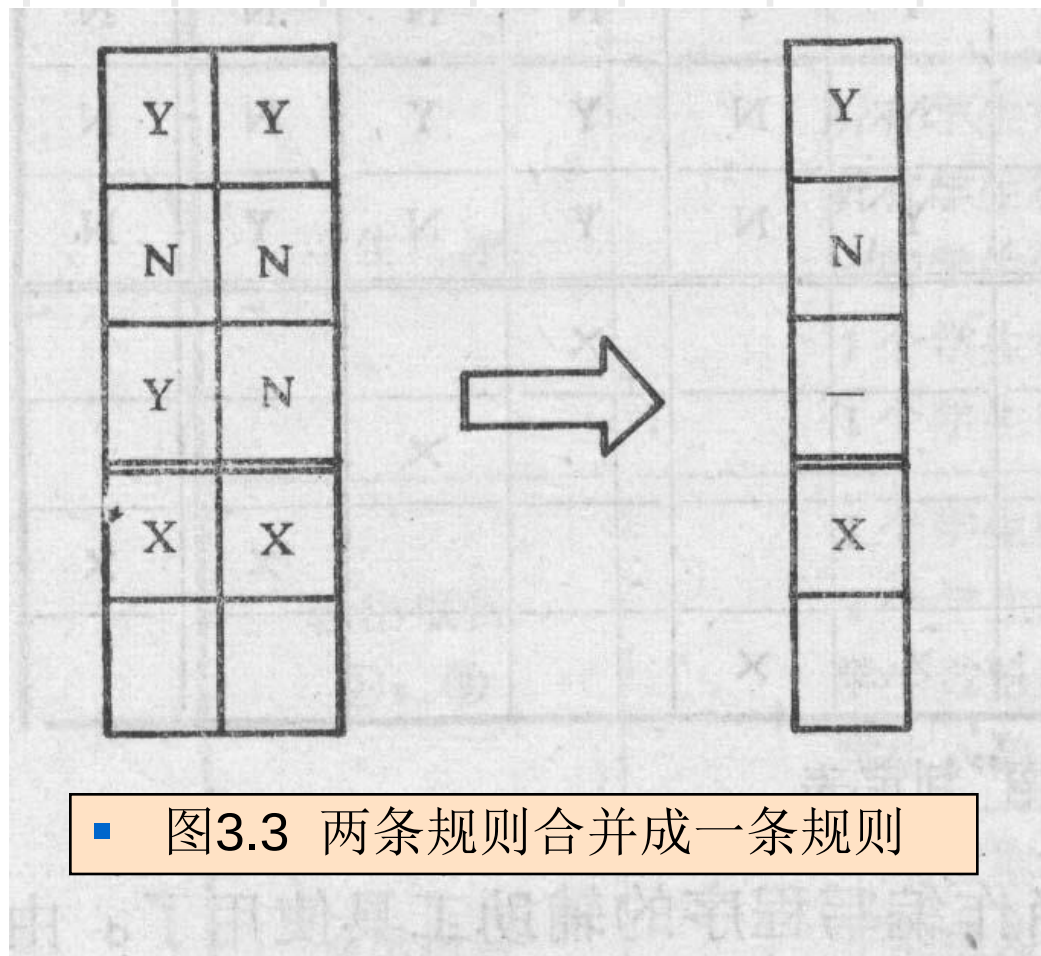
■ 二、判定表的化简

- 在实际使用判定表时，常常先把它化简。化简工作是以合并相似规则为目标的。若表中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，我们便可设法将其合并。例如，在图3.3中左端的两规则其动作项一致，条件项中前两条条件取值一致，只是第三条条件取值不同。这一情况表明，在第1、2条件分别取真值和假值时，无论第三条条件取任何值，都要执行同一操作。也即要执行的动作与第3条件的取值无关。于是，我们便将这两个规则合并。合并后的第3条件项用特定的符号“—”表示与取值无关。

与此类似，无关条件项“—”在逻辑上又可包含其它的条件项取值，具有相同动作的规则还可进一步合并，如图3.4所示。



§ 3 判定表法



§ 3 判定表法

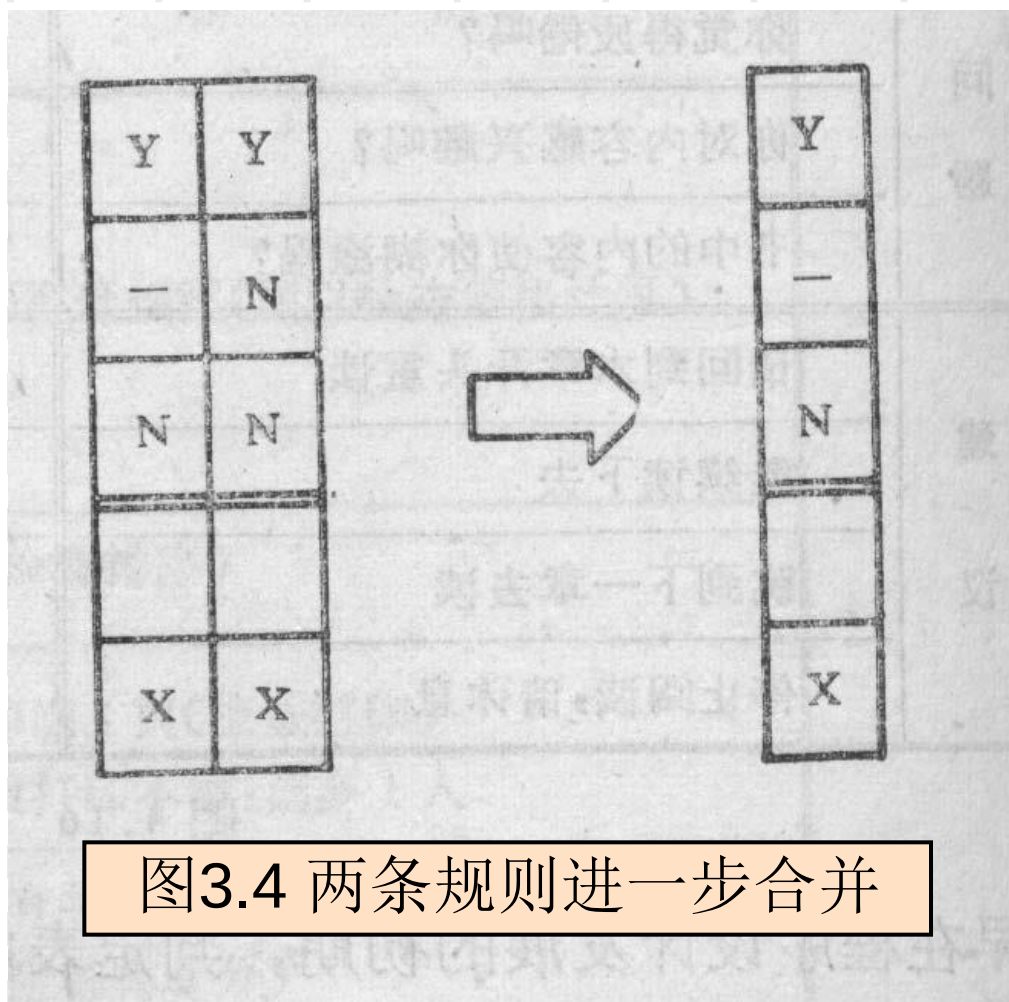


图3.4 两条规则进一步合并

§ 3 判定表法

- 判定表最突出的优点是，它能把复杂的问题按各种可能的情况一一列举出来，简明而易于理解，也可避免遗漏。

- 它的不足之处在于，不能表达重复执行的动作，例如循环结构。



§ 3 判定表法

按上述合并规则的办法，
可把“读书指南”判定表加以简化，
简化后的判定表见图3.5。

		1	2	3	4
问题	你觉得疲倦吗?	—	—	Y	N
	你对内容感兴趣吗?	Y	Y	N	N
	书中内容使你糊涂吗?	Y	N	—	—
建议	请回到本章开头重读	×			
	继续读下去		×		
	跳到下一章去读				×
	停止阅读,请休息			×	

图3.5 两条规则进一步合并

§ 3 判定表法



- 三、判定表在功能测试中的应用
- 一些软件的功能需求可用判定表表达得非常清楚，在检验程序的功能时判定表也就成为一个非常有用的工具。
- 如果一软件的规格说明指出：
 - ①当条件1和条件2满足，并且条件3和条件4不满足，或者当条件1、3和条件4满足时，要执行操作1。
 - (2) 在任一个条件都不满足时，要执行操作2。
 - (3) 在条件1不满足，而条件4被满足时，要执行操作3。



§ 3 判定表法

	规则 1	规则 2	规则 3	规则 4
条件 1	Y	Y	N	N
条件 2	Y	—	N	—
条件 3	N	Y	N	—
条件 4	N	Y	N	Y
操作 1	X	X		
操作 2			X	
操作 3				X

■ 图3.6 根据规格说明得到的判定表

§ 3 判定表法

■ 很显然，这个判定表中4个条件的**16**种规则并未一一给出。只是给出了**16**种规则中的**4**种，事实上，除这一条以外的一些规则是指当不能满足指定的条件，执行**3**种操作时，要执行**1**个默许的操作。在没有必要时，判定表通常可以略去这些规则。但如果我们打算用判定表来支持设计测试用例，那就有必要列出这些默许的规则（见图3.7）。

§ 3 判定表法

	规则2		规则1				规则8		规则4		规则5				规则3	
条件1	Y	Y	Y	Y	Y	Y	Y	Y	N	N	N	N	N	N	N	N
条件2	Y	Y	Y	Y	N	N	N	N	Y	Y	Y	Y	N	N	N	N
条件3	Y	Y	N	N	Y	Y	N	N	Y	Y	N	N	Y	Y	N	N
条件4	Y	N	Y	N	Y	N	Y	N	Y	N	Y		Y	N	Y	N
动作1	*			*	*											
动作2																*
动作3									*		*		*		*	
默认		*	*			*	*	*		*		*		*		

颜色相同者可合并

规则7

规则6

§ 3 判定表法

	规则 5	规则 6	规则 7	规则 8
条件 1	—	N	Y	Y
条件 2	—	Y	Y	N
条件 3	Y	N	N	N
条件 4	N	N	Y	—
默许操作	×	×	×	×

图3.7 默许的规则

§ 4 判定表法



■ B. Beizer指出了适合于使用判定表设计测试用例的条件：

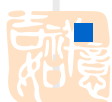
■ ①规格说明以判定表形式给出，或是很容易转换成判定表。

■ ②条件的排列顺序不会也不应影响执行哪些操作。

■ ③规则的排列顺序不会也不应影响执行哪些操作。

■ ④每当某一规则的条件已经满足，并确定要执行的操作后，不必检验别的规则。

■ ⑤如果某一规则得到满足要执行多个操作，这些操作的执行顺序无关紧要。





- **B.Beizer**提出这5个必要条件的目的是为了
为了使操作的执行完全依赖于条件的组合。
其实对于某些不满足这几条的判定表，同
样可以借以设计测试用例，只不过尚需增
加其它的测试用例罢了。



§ 3 判定表法



■ 四、保险的例子





- 作业：若要求“...对功率大于50马力的机器、维修纪录不全或已运行10年以上的机器，应给予优先维修处理...”，请给出判定表





		1	2	3	4	5	6	7	8
条件	功率大于50马力	Y	Y	Y	Y	N	N	N	N
	维修纪录不全	Y	Y	N	N	Y	Y	N	N
	运行超过过10年	Y	N	Y	N	Y	N	Y	N
动作	进行优先维修	X	X	X		X		X	
	作其他处理				X		X		X





		1	2	3	4	5
条件	功率大于50马力	Y	Y	Y	N	N
	维修纪录不全	Y	N	N	—	—
	运行超过过10年	—	Y	N	Y	N
动作	进行优先维修	X	X		X	
	作其他处理			X		X



■ 错误推测法

错误推测法: 基于经验和直觉推测程序中所有可能存在的各种错误, 从而有针对性的设计测试用例的方法.

错误推测方法的基本思想: 列举出程序中所有可能有的错误和容易发生错误的特殊情况, 根据他们选择测试用例. 例如, 在单元测试时曾列出的许多在模块中常见的错误. 以前产品测试中曾经发现的错误等, 这些就是经验的总结. 还有, 输入数据和输出数据为0的情况. 输入表格为空格或输入表格只有一行. 这些都是容易发生错误的情况. 可选择这些情况下的例子作为测试用例.