软件质量保证与测试

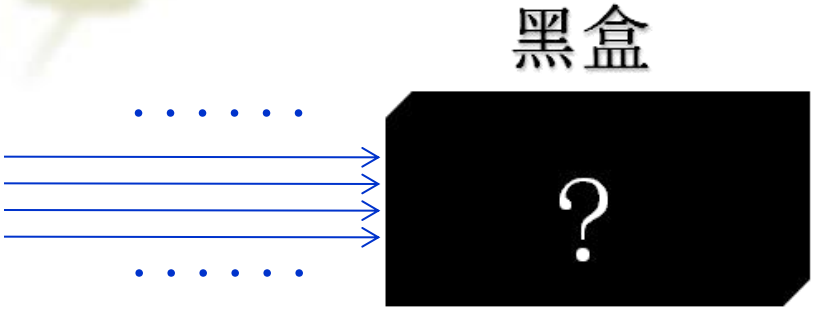
Software Quality Assurance and Testing

第 3 章 黑盒测试

3.2 等价类划分

金陵科技學院

黑盒测试

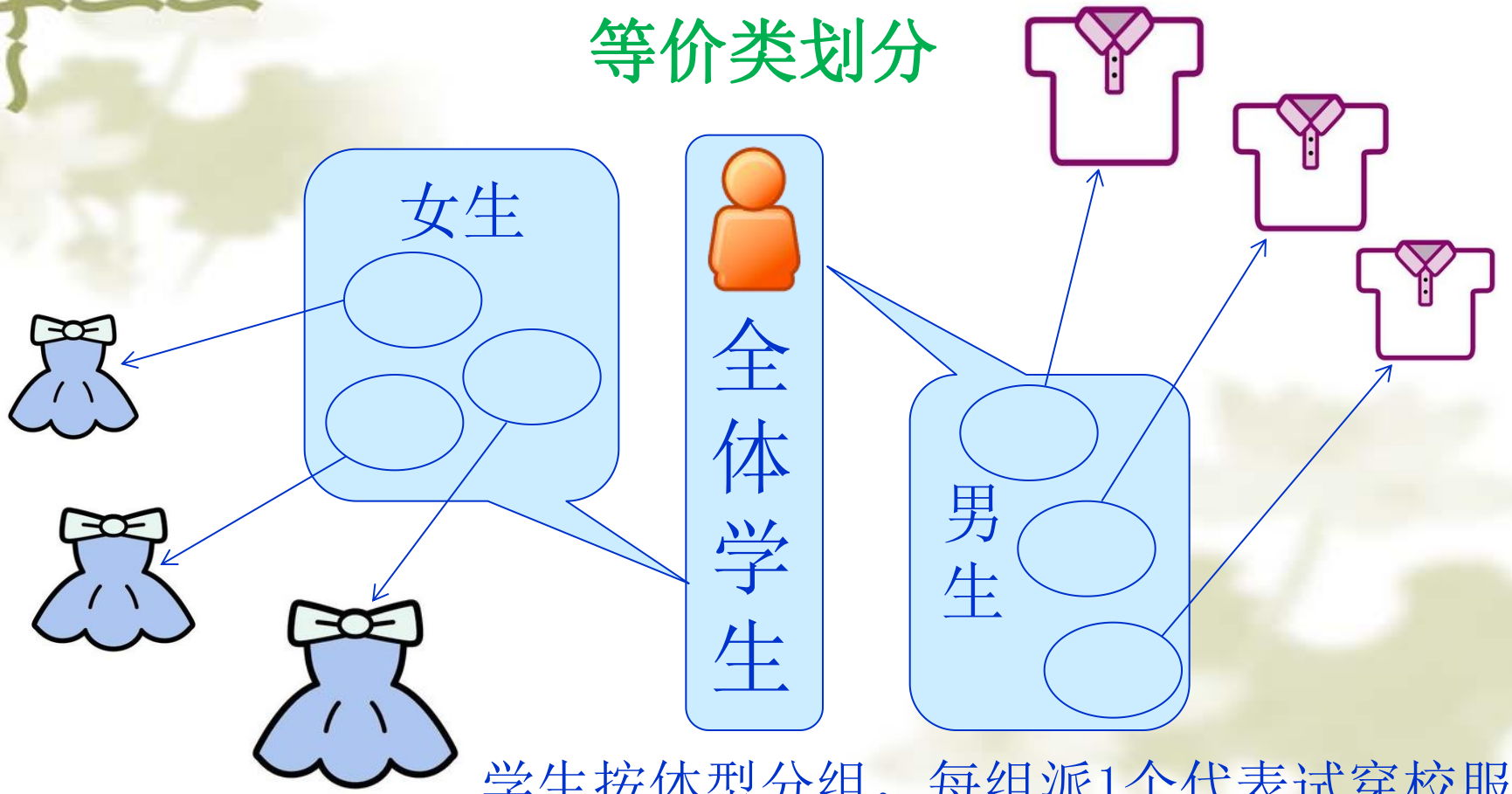


穷举所有可能的输入？

不可能！！！

必须要提高测试的针对性，既要测试各种可能的情况，提高测试的完备性，又要避免重复，降低冗余，节约测试成本！

等价类划分



学生按体型分组，每组派1个代表试穿校服

等价类

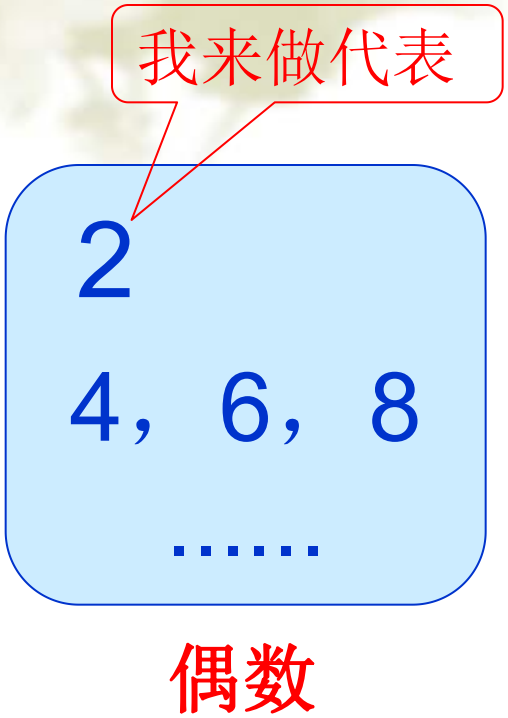
某个元素相应的等价类是指，对某一个等价关系而言，与其等价的所有元素的集合。

简单地说，等价类是数据集的某个子集，等价类中的各个元素具有某种相同的特性。

例如按照奇偶性，整数可以分为奇数和偶数两个等价类。

奇偶性
- 0,2,4,.....都等价，都是偶数，它们构成偶数等价类
- 1,3,5,.....都等价，都是奇数，它们构成奇数等价类

等价类

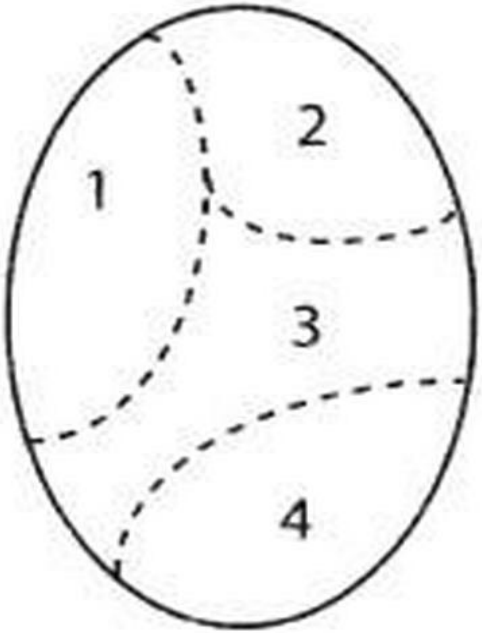


从软件测试的角度来说，由于等价类中的各个元素具有相同的特性，所以对于发现或者揭露程序中的缺陷，它们的作用是等价的，或者说效果是相同的。

于是等价类划分法就合理地假定：对于某个等价类而言，只需要测试其中的某个代表数据，就等于对这一等价类中所有数据的测试。

等价类划分

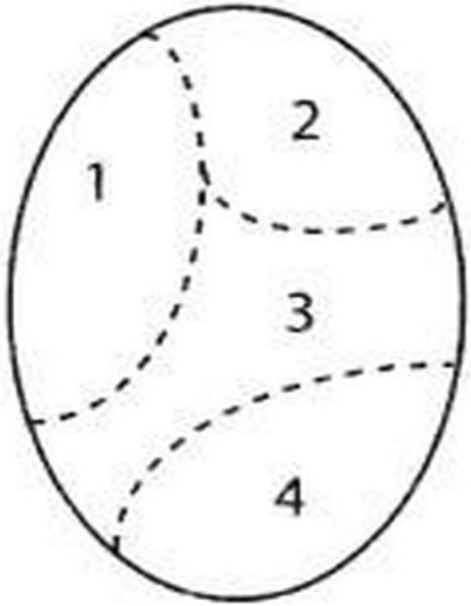
把所有可能的输入数据，划分成若干个等价类，然后从每一个等价类中选取1个或者少量数据，作为测试数据去测试程序。



各个等价类之间不应存在相同的元素。

所有等价类的并集应当是被划分集合的全集。

等价类划分



把可能无限的输入，变成有限的等价类，然后从中选出代表作为测试用例，以期达到在测试工作尽可能完备的同时又尽可能避免测试冗余，降低测试成本，提高测试的有效性。

等价类划分是最基本和最常用的黑盒测试方法。

等价类划分

等价类可以分为有效等价类和无效等价类。

有效等价类：是指对于程序规格说明来说，合理的、有意义的输入数据构成的集合。利用它，可以检验程序是否实现了规格说明预先规定的功能和性能等特性。

无效等价类：是指对于程序规格说明来说，不合理的、无意义的输入数据构成的集合。利用它，可以检验程序能否正确应对异常的输入，而不至于产生不希望出现的后果。

等价类划分

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并处理合理的数据，也要能经受意外的考验。

在遇到不合理的、无意义的数据输入时，程序应能妥善处理，而不至于无法应对，出现意外的结果，只有通过这样的测试，才能确保软件具有更高的可靠性。

等价类划分简单实例

符号函数

$$\begin{cases} x>0 \rightarrow y=1 \\ x=0 \rightarrow y=0 \\ x<0 \rightarrow y=-1 \end{cases}$$

有效等价类：三类，分别是x>0, x=0和x<0

无效等价类：一类，所有不能和0进行大小比较的数据

常见的划分等价类的方式

① 按区间划分

② 按数值划分

③ 按集合划分

④ 按限制条件或者限制规则划分

⑤ 按处理方式划分等

常见的划分等价类的方式

例如：

对某一个人所得税计算软件进行测试，可以按照个人所得税分等级计算标准，把输入数据“应纳税所得额”按区间进行等价类划分。

有效等价类	全月应纳税所得额
1	不超过1500元
2	超过1500元至4500元
3	超过4500元至9000元
4	超过9000元至35000元
5	超过35000元至55000元
6	超过55000元至80000元
7	超过80000元

常见的划分等价类的方式

例如：

对某五级计分制转换百分制程序进行测试，可以把输入数据“五级计分制成绩”按照处理方式进行等价类划分。

有效等价类
- 优秀 → 90
- 良好 → 80
- 中等 → 70
- 及格 → 60
- 不及格 → 50

划分等价类的建议

到目前为止，还没有高质量划分等价类的标准方法，针对软件不同的规格说明可能使用不同的等价类划分方法。不同的等价类划分得到的测试用例的质量不同。

在划分等价类时，可以参考下面的建议：

①如果输入条件规定了取值的范围，那么可以确定一个有效等价类和两个无效等价类。

例如：程序输入条件为小于等于100大于等于0的整数x，则有效等价类为0＜=x＜=100，两个无效等价类为x<0和x>100。

划分等价类的建议

②如果输入条件规定了一个输入值的集合，那么可以确定一个有效等价类和一个无效等价类。

例如:某程序规定了输入数据ZC的有效取值需来自集合R={助教、讲师、副教授、教授、其它、无}，则有效等价类为ZC属于R，无效等价类为ZC不属于R

③如果输入条件规定了输入值必须满足某种要求，那么可以确定一个有效等价类和一个无效等价类。

例如：某程序规定输入数据x的取值条件为数字符号，则有效等价类为x是数字符号，无效等价类为x含有非数字符号。

划分等价类的建议

④在输入条件是一个布尔量的情况下，那么可以确定一个有效等价类和一个无效等价类。

例如：某程序规定其有效输入为布尔真值，则有效等价类为布尔真值true，无效等价类为布尔假值false。

有效等价类：布尔真值 true

无效等价类：布尔假值 false

划分等价类的建议

⑤如果规定了输入数据为一组值（n个），并且程序要对每一组输入值分别进行处理，那么可以确定n个有效等价类和一个无效等价类。

例如：某程序输入x取值于一组值{优秀, 良好, 中等, 及格，不及格}，且程序中会对这5个值分别进行处理，则有效等价类有5个，分别为x=“优秀”、x=“良好”、x=“中等”、 x=“及格” 、x=“不及格”，无效等价类为x不属于集合{优秀, 良好, 中等, 及格，不及格}。

划分等价类的建议

⑥如果规定输入数据必须符合某些规则，那么可以确定一个有效等价类（符合规则）和若干个分别从不同角度违反规则的无效等价类。

例如：程序输入条件为以英文字母开头、长度为8、只包含英文字母和数字符号的字符串，则有效等价类为满足了上述所有条件的字符串，无效等价类为不以英文字母开头的字符串、长度不为8的字符串和包含了英文字母和数字符号之外其它字符的字符串。

划分等价类的建议

⑦在初步划分等价之后，如果发现某一等价类中的各元素在程序中的处理有区别，则应再将该等价类进一步划分为更小的等价类。

例如：

- 输入的成绩
  - 有效等价类
    - 五级计分制文本数据
    - [0，100]范围内数字数据
  - 无效等价类：
    - 非五级计分制的文本数据
    - 超出[0，100]范围的数字数据

等价类划分方法设计测试用例步骤

1. 划分等价类，包括有效等价类和无效等价类，建立等价类表，并为每一个等价类规定一个惟一的编号；

以符号函数为例：

条件	有效等价类	编号	无效等价类	编号
x	x<0	Y1	不能和0比较大小的输入	N1
x	x=0	Y2		
x	x>0	Y3		

等价类划分方法设计测试用例步骤

2. 设计一个新的测试用例，使其尽可能多地覆盖尚未覆盖的有效等价类；重复这一步骤，直到所有的有效等价类都被覆盖为止。

编号	测试用例	覆盖的等价类
T1	x= -4	Y1
T2	x=0	Y2
T3	x= 4	Y3

等价类划分方法设计测试用例步骤

3. 设计一个新的测试用例，使其仅覆盖一个无效等价类，重复这一步骤，直到所有的无效等价类都被覆盖为止。

编号	测试用例	覆盖的等价类
T1	x= -4	Y1
T2	x=0	Y2
T3	x= 4	Y3
T4	x= “good”	N1

等价类划分

为什么设计一个测试用例可覆盖多个有效等价类，而一般只能覆盖一个无效等价类？

假设有一个成绩输入软件，输入的成绩由平时成绩cj1和期末成绩cj2两部分组成，cj1、cj2的无效等价类都是两个，小于0和大于100。

等价类划分

程序员在编写程序时，用两条语句来应对可能的无效输入，代码本应当如下：

If （cj1<0 or cj1>100） Return “平时成绩超出范围”
If （cj2<0 or cj2>100） Return “期末成绩超出范围”

但程序员在敲代码时出现了疏忽，把cj2>100，写成了cj2>1000，代码变成了：

If （cj1<0 or cj1>100） Return “平时成绩超出范围”
If （cj2<0 or cj2>1000） Return “期末成绩超出范围”

等价类划分

现在我们设计一个测试用例 cj1 = -10，cj2 = 800，覆盖了两个无效等价类，分别是cj1的小于0无效等价类，和cj2的大于100无效等价类。

我们输入这一测试数据，程序执行在执行完第一行对cj1进行有效性检验之后，就提示“平时成绩超出范围”并退出执行返回了，根本就没有继续执行第二行代码。这样第二行的错误就发现不了，而我们很可能还错误的认为程序已经顺利通过了针对cj2的大于100无效等价类的测试。

而如果一次只覆盖一个无效等价类，如cj1 = 70，cj2 = 800，就可以发现第二行代码中的错误。

等价类划分

所以说一个测试用例可覆盖多个有效等价类，而一般只能覆盖一个无效等价类。

除非是：一次覆盖一个无效等价类已经做完了，专门再来对多个变量做无效等价类的组合覆盖。

等价类的组合

如果有多个输入条件，并且各个条件之间存在关联，那么仅仅只是覆盖所有的等价类还不够，还需要考虑等价类之间的组合。

组合可分为完全组合和部分组合两种，如果输入条件比较多，并且每个输入条件的等价类也比较多，那么总的完全组合数将非常大，此时可以采用部分组合。

等价类的组合

实例：

函数y = f (x1, x2) 输入变量的取值范围分别为:x1 ∈[a, d], x2 ∈ [e, g]。

根据函数的规格说明划分得到相应的等价类

X1：
有效等价类 [a, b) [b, c) [c, d];
无效等价类 (-∞, a), (d, +∞)

X2：
有效等价类 [e, f) [f, g];
无效等价类 (-∞, e), (g, +∞)

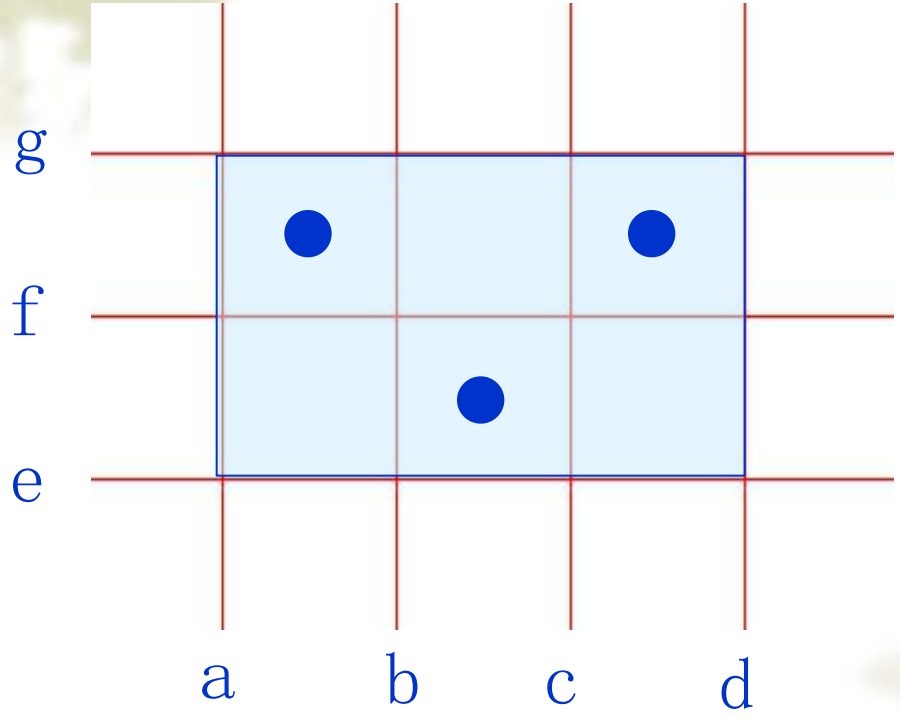
等价类的组合

弱一般等价类：

设计若干测试用例，每个测试用例应尽可能多地覆盖尚未覆盖的被测变量的有效等价类并且每个被测变量的有效等价类应至少出现一次。

测试用例个数为：各个被测变量中的最大有效等价类个数。

等价类的组合



弱一般等价类测试用例

有效区域

无效区域

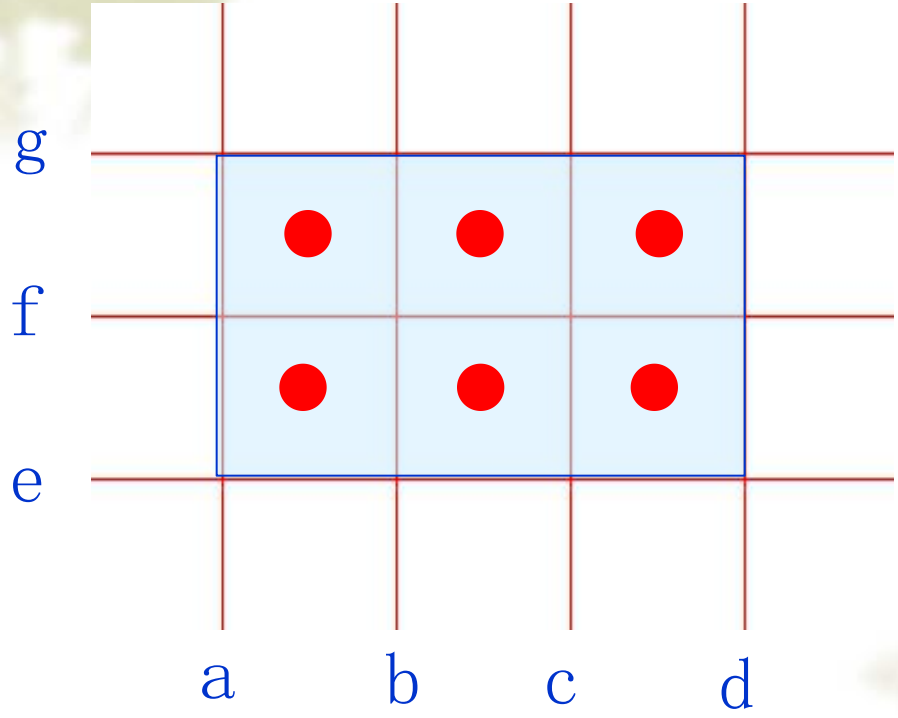
等价类的组合

强一般等价类：

设计若干测试用例，使其覆盖所有被测变量有效等价类的组合。

测试用例个数为：各个被测变量有效等价类数的乘积。

等价类的组合



强一般等价类测试用例

有效区域

无效区域

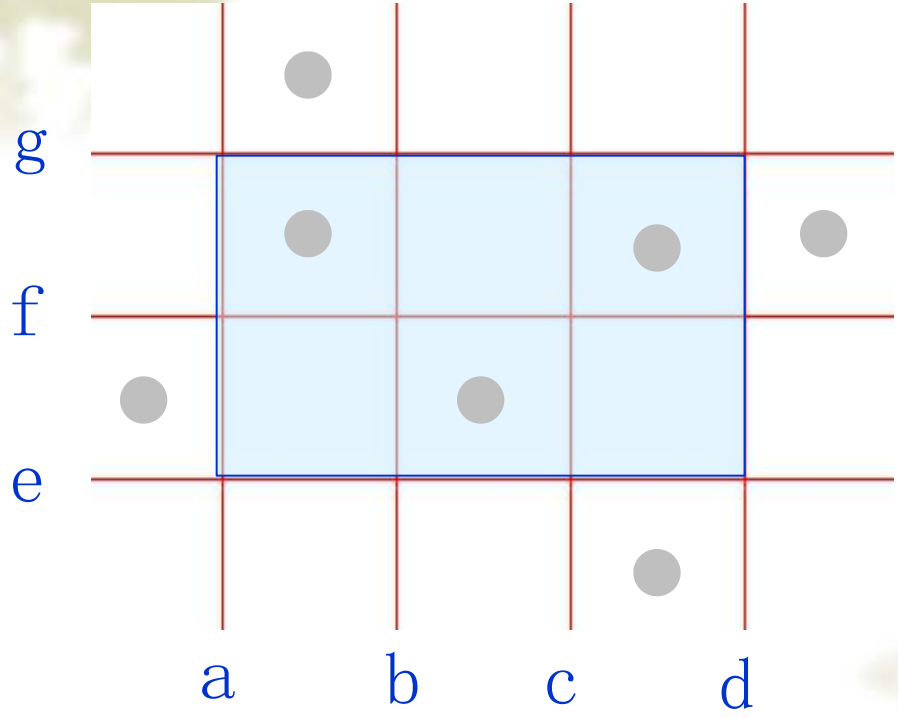
等价类的组合

弱健壮等价类：

设计若干测试用例，每个测试用例应尽可能多地覆盖尚未覆盖的有效等价类，对于无效等价类，每个测试用例只考虑一个被测变量的无效等价类。

测试用例个数为，各个被测变量中的最大有效等价类个数+Σ各个被测变量的无效等价类数。

等价类的组合



弱健壮等价类测试用例

有效区域

无效区域

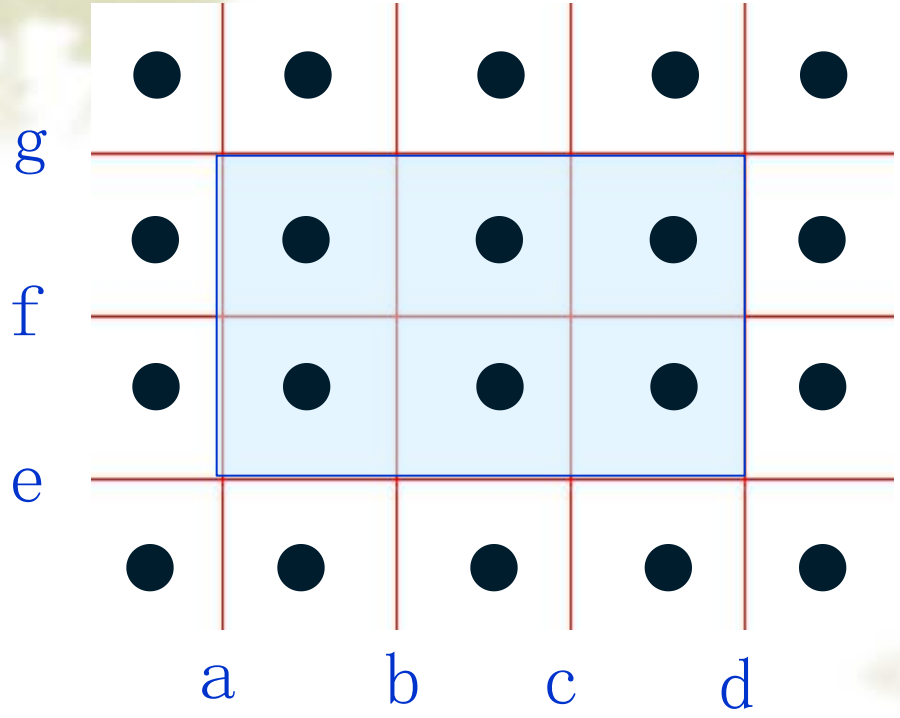
等价类的组合

强健壮等价类：

设计若干测试用例，使其覆盖所有被测变量的有效等价类和无效等价类的组合。

测试用例个数为，各个被测变量的等价类总数的乘积，各个被测变量的等价类总数等于其有效等价类数+无效等价类数。

等价类的组合



强健壮等价类测试用例

有效区域

无效区域

等价类的组合

- 弱：至少覆盖一次即可
- 强：覆盖组合

- 一般：只覆盖有效等价类
- 健壮：覆盖有效和无效等价类

等价类划分

等价类划分方法的思想本质是，依据软件规格说明，将数据按照处理方式的不同，划分为不同的等价类，从等价类中选出代表作为测试用例，以期达到软件测试工作尽可能完备同时又避免冗余。

等价类划分测试方法除了可以用于输入数据之外，还可以对输出数据应用实施。