

黑盒测试技术

black-box-testing

【书上第3.4章】

Dynamic black-box testing

- Dynamic black-box testing is testing without having an insight into the details of the underlying code.
 - Dynamic, because the program is running
 - Black-box, because testing is done without knowledge of how the program is implemented.
- Sometimes referred to as *behavioral testing*.
- Requires an executable program and a specification (or at least a user manual).
- Test cases are formulated as a set of pairs
 - E.g., (input, expected output)

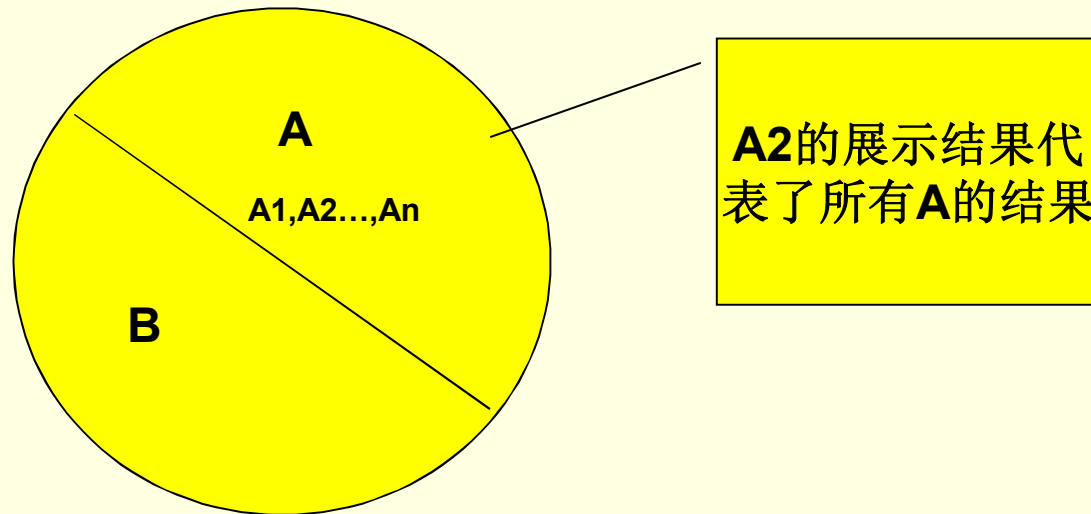
Test Data and Test Cases

- **Test data:** Inputs which have been devised to test the system.
- **Test cases:** Inputs to test the system and the predicted outputs from these inputs if the system operates according to its specification.

黑盒测试

- 测试用例设计技术
 - 等价类划分方法
 - 边界值分析方法
 - 错误推测方法
 - 判定表驱动分析方法
 - 因果图方法
 - 场景法

Equivalence Partitioning



- 这样，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。可以减少测试用例的量

划分等价类

■ 等价类划分有两种不同的情况：

❖ **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非功能。

❖ **无效等价类**：与有效等价类的定义恰巧相反。

■ 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。

数学知识 - 集合论

集合的划分

A_1, A_2, \dots, A_n 是集合 A 的子集

A_1, A_2, \dots, A_n 是集合 A 的一个划分

$A_1 \cup A_2 \cup \dots \cup A_n = A$ 且

$A_i \cap A_j = \emptyset \ (i \neq j)$

测试 (1) 完备性

(2) 无冗余性

确定等价类的原则

- 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。

确定等价类的原则

- 2. 在输入条件规定了输入值的集合或者规定了“必须如何”的条件的前提下，可以确立一个有效等价类和一个无效等价类。

确定等价类的原则

- 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。

等价类划分

4. 在规定了输入数据的一组值，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。

确定等价类的原则

- 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，)。

确定等价类的原则

(6) 如果确知，已划分的等价类中各个元素在程序中的处理方式不同，则应将此等价类进一步划分成更小的等价类。

有效等价类：

无效等价类：

```
例如： if(i>=0&& i<=100)
switch(i) {
    case 1:...;break;
    case 3:...;break;
    case 5:...;break;
    ...
}
```

根据等价类创建测试用例的步骤

- 建立等价类表，列出所有划分出的等价类：

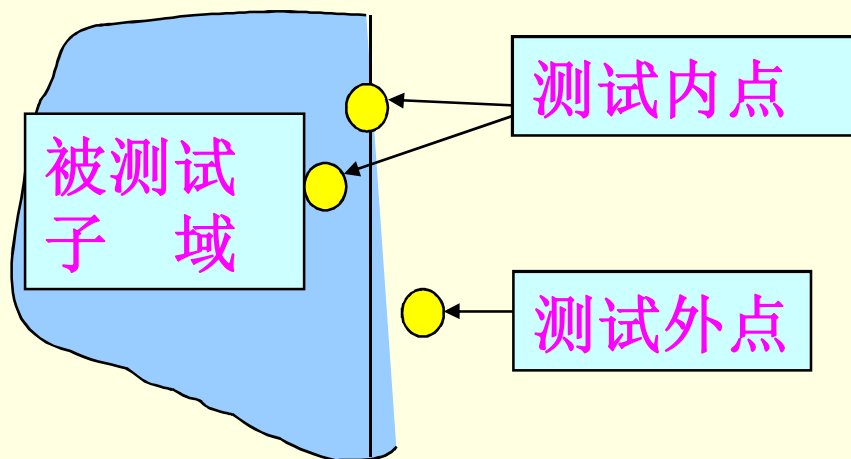
分析	有效等价类	无效等价类
...
...

- 为每个等价类规定一个唯一的编号；
- 设计一个新的测试用例，使其覆盖一个有效等价类。重复这一步，最后使得所有有效等价类均被测试用例所覆盖；
- 设计一个新的测试用例，使其只覆盖一个无效等价类。重复这一步使所有无效等价类均被覆盖。

14.3.2 边界值分析法

(BVA, Boundary Value Analysis)

- 边界值分析也是一种黑盒测试方法，是对等价类划分方法的补充。
- 边界值分析不是从某等价类中随便挑一个作为代表，而是使这个等价类的每个边界都要作为测试条件。
- 人们从长期的测试工作经验得知，大量的错误是发生在输入或输出范围的边界上，而不是在输入范围的内部。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。



边界值测试用例设计方法

- 边界值分析法：
 - 是等价类划分法的一个补充
 - 程序的很多错误发生在输入或输出范围的边界上，因此针对各种边界情况设置测试用例，可以发现不少程序缺陷。
 - 设计方法：
 - 确定边界情况（输入或输出等价类的边界）
 - 选取正好等于、刚刚大于或刚刚小于边界值作为测试数据
- 一般情况，一个边界可以给出3个边界测试用例，自身+左右值