

## 黑盒测试

- 黑盒测试的概念
- 事务流测试
- 等价类划分
- 边界值分析

### 黑盒测试：

黑盒测试也称功能测试或数据驱动测试，它是在已知产品所应具有的功能，通过测试来检测每个功能是否都能正常使用，在测试时，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，测试者在程序接口进行测试，它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，并且保持外部信息（如数据库或文件）的完整性。

黑盒测试方法主要有等价类划分、边界值分析、因果图、错误推测等，主要用于软件确认测试。

“黑盒”法是穷举输入测试，只有把所有可能的输入都作为测试情况使用，才能以这种方法查出程序中所有的错误。实际上测试情况有无穷多个，人们不仅要测试所有合法的输入，而且还要对那些不合法但是可能的输入进行测试。



# 黑盒测试

- 黑盒测试的概念
- 事务流测试
- 等价类划分
- 边界值分析

黑盒测试试图发现以下错误类型：

- ( 1 ) 功能不对或遗漏；
- ( 2 ) 界面错误；
- ( 3 ) 数据结构或外部数据库访问的错误；
- ( 4 ) 性能错误；
- ( 5 ) 初始化和终止错误。

问题：黑盒测试依据什么？

依据软件行为的描述的测试



北京大学

# 黑盒测试

- 黑盒测试的概念
- 事务流测试
- 等价类划分
- 边界值分析

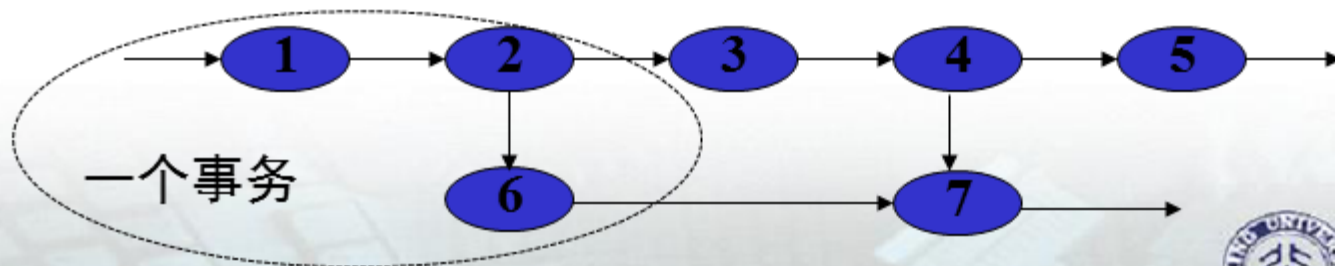
## 8.5.2.1 事务流测试技术

### (1) 基本概念:

① **事务**: 以用户的角度所见的一个工作单元。

一个事务由一系列操作组成。其中某些操作可含有系统执行成分, 或含有设备执行成分, 它们共同协作, 完成用户的一项工作。

② **事务处理流程 (图)**: 系统行为的一种表示方法, 为功能测试建立了软件动作模式。其中使用了白盒测试中的一些概念, 例如: 操作 (如下图 1、3、6、5)、分支 (下图 2), 节点 (下图 7), 链 (下图中 →) 等。



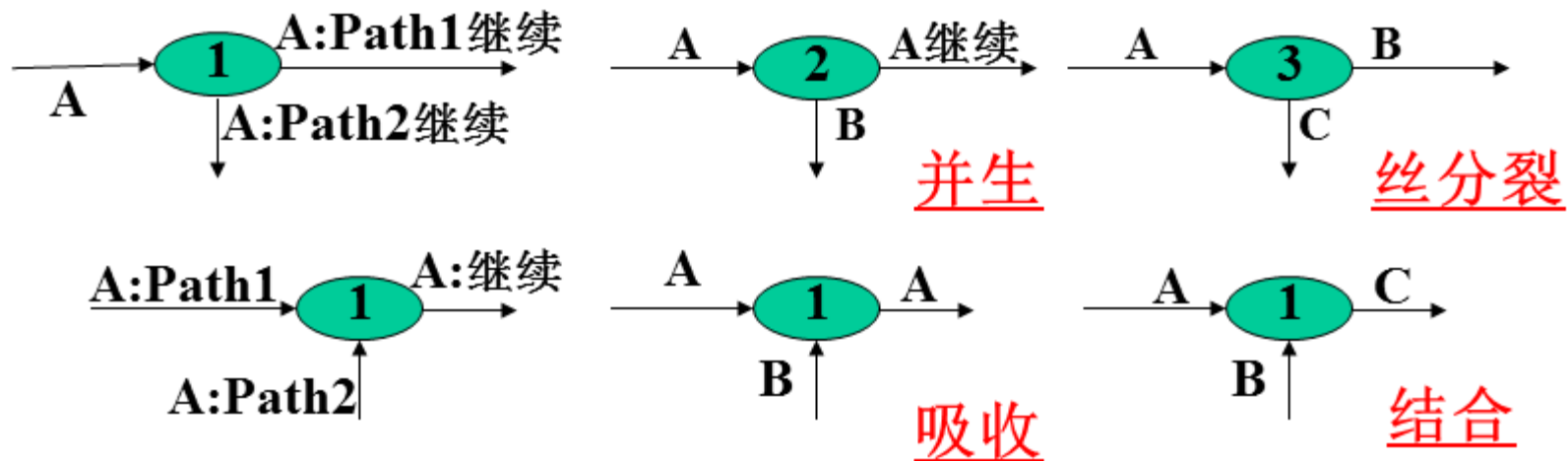
# 黑盒测试

- 黑盒测试的概念
- 事务流测试
- 等价类划分
- 边界值分析

(2) 与程序控制流程图的比较:

① 事务流图是一种数据流图，即从操作应用的历史，观察数据对象。

② 事务流图中的分支：“抽象”了一个复杂的过程。



③ 事务流图存在“中断”，把一个过程等价地变换为具有繁多出口的链支。

测试设备:路径分析器,测试用例数据库,  
测试执行调度器



北京大学



# 黑盒测试

- 黑盒测试的概念
- 事务流测试
- 等价类划分
- 边界值分析

## (3) 测试步骤

第一步：获取事务流程图，即建立被测对象模型；

第二步：浏览与复审

主要对事务进行分类，为设计用例奠定基础；

第三步：用例设计

设计足够测试用例，实现基本事务覆盖。

涉及：覆盖策略，事务选取等；

第四步：测试设备开发：

路径分析器，测试用例数据库，

测试执行调度器，...

第五步：测试执行；

第六步：测试结果比较。



北京大学

# 黑盒测试

- 黑盒测试的概念
- 事务流测试
- 等价类划分
- 边界值分析

## 8.5.2.2 等价类划分技术

### (1) 基本概念

① **等价类**：输入域的一个子集，在该子集中，各个输入数据对于揭示程序中的错误都是等效的。即：以等价类中的某代表值进行的测试，等价于对该类中其他取值的测试。

② **有效等价类**：指那些对于软件的规格说明书而言，是合理的、有意义的输入数据所构成的集合。

—用于实现功能和性能的测试。

③ **无效等价类**：指那些对于软件的规格说明书而言，是不合理的、无意义的输入数据所构成的集合。

—用于测试那些所实现的功能和性能不符合规格说明书的要求。



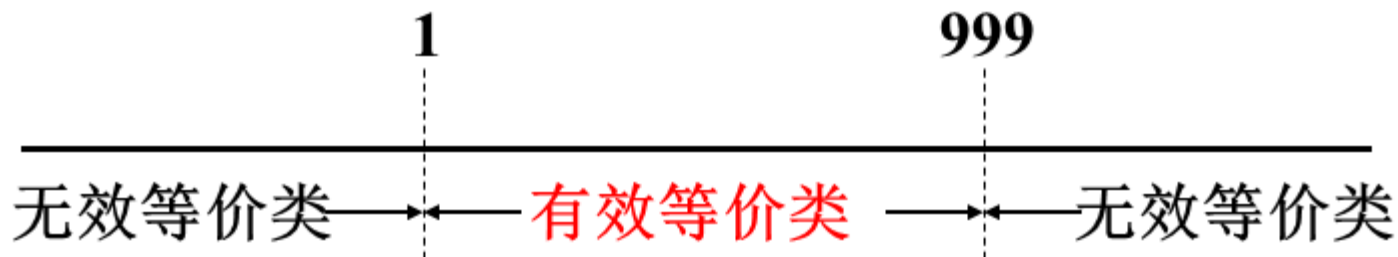
# 黑盒测试

- 黑盒测试的概念
- 事务流测试
- 等价类划分
- 边界值分析

## (2) 等价类划分原则（指南）

① 如果输入条件规定了输入数据的取值范围或值的个数，则可以确定一个有效等价类和二个无效等价类。

例如：输入条件：“...项数可以是1到999...”



② 如果输入条件规定了输入数据的一组值，而且软件要对每个输入值进行处理，则可以为每一个输入值确定一个有效等价类，为所有不允许的输入值确定一个无效等价类。

例如：对大一、大二、大三、大四的学生进行管理，可确定4个有效等价类，为大一、大二、大三、大四的学生，一个无效等价类为不符合上述身份的人。



北京大学

# 黑盒测试

- 黑盒测试的概念
- 事务流测试
- 等价类划分
- 边界值分析

③ 如果输入条件是一个布尔量，则可以确定一个有效等价类和一个无效等价类。

④ 如果输入条件规定了输入值必须符合的条件，则可以确定一个有效等价类（符合条件）和一个无效等价类（不符合条件）。

例如：“标识符是一字母打头的...串。” 则  
字母打头的--为一个有效等价类，而  
其余的--为一个无效等价类

注意：如果在已确定的等价类中各元素在软件中的处理方式不同，则应根据需要对等价类进一步进行划分。





# 黑盒测试

- 黑盒测试的概念
- 事务流测试
- 等价类划分
- 边界值分析

## (3) 测试用例设计

在确定了等价类之后，建立等价类表：

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


再根据等价类来设计测试用例，过程如下：

- (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，使其尽可能多的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；



# 黑盒测试

- 黑盒测试的概念
- 事务流测试
- 等价类划分
- 边界值分析




(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。

问题：为什么设计无效等价类的测试用例时仅包括一个未被覆盖的无效等价类？



# 黑盒测试

- 黑盒测试的概念
- 事务流测试
- 等价类划分
- 边界值分析



(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。

问题：为什么设计无效等价类的测试用例时仅包括一个未被覆盖的无效等价类？

因为某些程序中对某一输入错误的检查往往会屏蔽对其它输入错误的检查。因此设计无效等价类的测试用例时应该仅包括一个未被覆盖的无效等价类。



# 黑盒测试

- 黑盒测试的概念
- 事务流测试
- 等价类划分
- 边界值分析

## 8.5.2.3 边界值分析

**边界值分析是一种最常用的黑盒测试技术。**因为测试工作经验表明，大量的错误经常发生在输入或输出范围的边界上。因此设计一些测试用例，使程序运行在边界情况附近，这样揭露错误的可能性比较大。

使用边界值分析设计测试用例可遵循以下原则：

- (1) 如果某个输入条件规定了输入值的范围，则应选择正好等于边界值的数据，以及刚刚超过边界值的数据作为测试数据。
- (2) 如果某个输入条件规定了值的个数，则可用最大个数、最小个数、比最大个数多1、比最小个数少1的数作为测试数据。
- (3) 根据规格说明的每个输出条件，使用前面的原则（1）。
- (4) 根据规则说明的每个输出条件，使用前面的原则（2）。





# 黑盒测试

- 黑盒测试的概念
- 事务流测试
- 等价类划分
- 边界值分析

(5) 如果程序的规格说明中，输入域或输出域是一个有序集合，在实践中，则经常选取集合的第一个元素、最后一个元素以及典型元素作为测试用例。

(6) 如果程序中使用了内部数据结构，则应选择这个内部数据结构的边界上的值作为测试用例。

(7) 分析规则说明，找出其他可能的边界条件。

