

测试用例设计 - 黑盒测试

测试用例设计 - 黑盒测试

- 测试用例设计概述
- 黑盒测试的概念
- 黑盒测试的数学知识
- 黑盒测试用例设计技术

测试用例设计概述

- 测试用例的定义
- 测试用例的重要性
- 好测试用例的特点
- 设计测试用例的基本准则
- 设计测试用例的着眼点
- 测试用例的编写标准
- 测试用例设计的误区

测试用例设计概述

□ 什么是软件测试用例

- (1) 测试用例是为特定的目的而设计的一组测试输入、 执行条件和预期的结果。
- (2) 测试用例是执行的最小实体。

测试用例设计概述

□ 好测试用例的特点

1. 完整的

完整性是对测试用例最基本的要求，尤其是一些基本功能项上，如果有遗漏，那将是不可原谅的。完整性还体现在临界测试、压力测试、性能测试等方面，这方面测试用例也要能够涉及到。

2. 准确

测试者按照测试用例的输入一步步测试完成后，要能够根据测试用例描述的输出得出正确的结论，不能出现模糊不清的语言。

3. 清晰、简洁

好的测试用例描述清晰，每一步都应该有响应的作用，有很强的针对性，不应该出现一些冗繁无用的操作步骤。测试用例不应该太简单，也不能够太过复杂，最大操作步骤最好控制在15步之内。

测试用例设计概述

5. 可维护性

由于软件开发过程中需求变更等原因的影响，常常需要对测试用例进行修改、增加、删除等，以便测试用例符合相应测试要求。测试用例应具备这方面的功能。

6. 适当性

测试例应该适合特定的测试环境以及符合整个团队的测试水平，如纯英语环境下的测试用例最好使用英文编写。

7. 可复用性

要求不同测试者在同样测试环境下使用同样测试用例都能得出相同结论。

测试用例设计概述

8. 其他

如可追溯性、可移植性也是对编写测试用例的一个要求。另外，好的测试用例也是最有可能抓住错误的；不重复、多余的；是一组相似测试用例中最有效的；

测试用例设计概述

□ 设计测试用例的基本准则

■ 测试用例的代表性

能够代表并覆盖各种合理的和不合理的、合法的和非合法的、边界的和越界的以及极限的输入数据、操作和环境设置等。

■ 测试结果的可判定性

即测试执行结果的正确性是可判定的，每一个测试用例都应有相应的期望结果。

■ 测试结果的可再现性

即对同样的测试用例，系统的执行结果应当是相同的。

测试用例设计概述

□ 设计测试用例的着眼点

- 根据产品规格，测试基本功能；
- 考虑设计一般用户（非专业人员）的使用方案；
- 考虑设计稀有或特殊的使用方案；
- 与系统其他组成部分的配合（如FAX和上网可能要用到MODEM，测试中考虑对设备的共享）；
- 考虑特殊情况（如内存和硬件的冲突等）；
- 设计极端情况（如内存泄漏、破坏性测试等）；
- 好的测试用例集能花费最小的代价（人力、物力、财力、时间）做最好的测试。

测试用例设计概述

□ 测试用例的编写标准

在ANSI/IEEE829-1983标准中列出了和测试设计相关的测试用例编写规范和模板。标准模板中主要元素如下：

- 标识符——惟一标识每一个测试用例
- 测试项——准确的描述所需要测试的项及其特征
- 测试环境要求——表征执行该测试用例需要的测试环境
- 输入标准——执行测试用例的输入需求（这些输入可能包括数据、文件或者操作）
- 输出标准——按照指定的环境和输入标准得到的期望输出结果
- 测试用例之间的关联——标识该测试用例与其它测试（或其它测试用例）之间的依赖关系

测试用例设计概述

- 测试用例示例：对Windows记事本程序进行测试，选取其中的一个测试项——文件菜单栏的测试：测试对象——记事本程序文件菜单栏（测试用例标识10000，下同）

所包含的子测试用例描述如下：

- |-----文件/新建（1001）
- |-----文件/打开（1002）
- |-----文件/保存（1003）
- |-----文件/另存为（1004）
- |-----文件/页面设置（1005）
- |-----文件/打印（1006）
- |-----文件/退出（1007）

选取其中的一个子测试用例——文件/退出（1007）作为例子，测试用例如下表所示。

字段名称	描 述
标识符	1007
测试项	记事本程序，“文件”菜单栏中的“文件” “退出”命令的功能测试
测试环境要求	Windows 2000 Professional 中文版
输入标准	<p>(1) 打开记事本程序，不输入任何字符，单击“文件” “退出”命令</p> <p>(2) 打开记事本程序，输入一些字符，不保存文件，单击“文件” “退出”命令</p> <p>(3) 打开记事本程序，输入一些字符，保存文件，单击“文件” “退出”命令</p> <p>(4) 打开一个记事本文件 (*.txt)，不做任何修改，单击“文件” “退出”命令</p> <p>(5) 打开一个记事本文件，修改后不保存，单击“文件” “退出”命令</p>
输出标准	<p>(1) 记事本未做修改，单击“文件” “退出”命令，能正确地退出应用程序，无提示信息</p> <p>(2) 记事本做修改未保存或者另存，单击“文件” “退出”命令，会提示“未定标题文件的文字已经改变，想保存文件吗？”单击“是”，Windows 将打开“保存”/“另存为”对话框，单击“否”，文件将不被保存并退出记事本程序，单击“取消”将返回记事本窗口</p>
测试用例间的关联	12

测试用例设计概述

□ 下表是另一个测试用例示例(登录功能测试)

该测试用例并未涵盖所有的非法输入，如非法输入中可能会有“user=*,pw=*”的组合，对回车的默认操作，空格输入，对输入上溢的处理的处理（可能会跳过身份验证）等等。

如果你有兴趣，至少可以再补充5-10条左右的输入组合（当然，如果步骤超过15步，用例的易操作性就降低，可以再创建一个测试用例）。

项目/软件	技术出口合同网络申 领系统（企业端）	程序版本	1.0.25			
功能模块 名	Login	编制人	xxx			
用例编号-	TC-TEP_Login_1	编制时间	2002.10.12			
相关的用 例	无					
功能特性	用户身份验证					
测试目的	验证是否输入合法 的信息，允许合法 登陆，阻止非法 登陆					
预置条件	无	特殊规程说 明	如数据库访问权限			
参考信息	需求说明中关于“登陆” 的说明					
测试数据	用户名=yiyh 密码=1					14

操作步骤	操作描述	数据	期望结果	实际结果	测试状态
1	输入用户名称，按“登陆”按钮。	用户名=yiyh，密码为空	显示警告信息“请输入用户名和密码！”		
2	输入密码，按“登陆”按钮。	用户名为空，密码=1	显示警告信息“请输入用户名和密码！”		
3	输入用户名和密码，按“登陆”按钮。	用户名=yiyh，密码=2	显示警告信息“请输入用户名和密码！”		
4	输入用户名和密码，按“登陆”按钮。	用户名=xxx，密码=1	显示警告信息“请输入用户名和密码！”		
5	输入用户名和密码，按“登陆”按钮。	用户名=xxx，密码=2	显示警告信息“请输入用户名和密码！”		
6	输入用户名和密码，按“登陆”按钮。	用户名=空，密码=空	显示警告信息“请输入用户名和密码！”		

7	输入用户名和密码，按“登陆”按钮。	用户名=yiyh，密码=1	进入系统页面。		
8	输入用户名和密码，按“登陆”按钮。	用户名=Admin，密码=admin	进入系统维护页面。		
9	输入用户名和密码，按“登陆”按钮。	用户名=yiyh',密码=1	显示警告信息“请输入用户名和密码！”		
10	输入用户名和密码，按“登陆”按钮。	用户名=yiyh，密码=1'	显示警告信息“请输入用户名和密码！”		
11	输入用户名和密码，按“重置”按钮。	用户名=yiyh，密码=1	清空输入信息		
测试人员		开发人员			

测试用例设计概述

□ 测试用例设计的误区

- 能发现到目前为止没有发现的缺陷的用例是好的用例
作为测试实施依据的测试用例，应当作为一个集合来认识，
必须要能完整覆盖测试需求，而不应该针对单个的测试用例去评判好坏。
- 测试用例应该详细记录所有的操作信息，使一个没有接触过系统的人员也能进行测试
测试用例维护费用太高，测试资源难保证
- 测试用例设计是一劳永逸的事情
测试用例是动态的，一旦测试环境、需求、设计、实现发生了变化，测试用例都需要相应发生变化

测试用例设计概述

- 测试用例不应该包含实际的数据
测试用例是“一组输入、执行条件、预期结果”、毫无疑问地应该包括清晰的输入数据和预期输出，没有测试数据的用例最多只具有指导性的意义，不具有可执行性。当然，测试用例中包含输入数据会带来维护、与测试环境同步之类的问题。
- 测试用例中不需要明显的验证手段
“预期结果”的含义应不只是程序的可见行为。例如，对一个订货系统，输入订货数据，点击“确定”按钮后，系统提示“订货成功”，这样是不是一个完整的用例呢？是不是系统输出的“订货成功”就应该作为我们唯一的验证手段呢？显然不是。订货是否成功还需要查看相应的数据记录是否更新，因此，在这样的一个用例中，还应该包含对测试结果的显式的验证手段：在数据库中执行查询语句进行查询，看查询结果是否与预期的一致。

黑盒测试

□ 黑盒测试的概念

■ 什么是黑盒测试？

黑盒测试又称功能测试、数据驱动测试或基于规格说明书的测试，是一种从用户观点出发的测试。

黑盒测试示意图



测试人员把被测程序当作一个黑盒子。

黑盒测试

□ 黑盒测试特点

- 测试人员不需了解程序内部的代码和实现
- 测试用例设计不依赖于系统内部的设计与实现
- 测试用例设计基于功能的定义和需求说明书
- 从用户角度出发进行测试，能较容易知道用户的需求和用户会遇到的问题
- 关注于测试数据的选择和测试结果的分析
- 对测试人员的编程技术要求不高
- 在自动化测试时较为方便
- 不易发现代码部分的缺陷

黑盒测试

□ 采用黑盒测试的目的

主要是在已知软件产品所应具有的功能的基础上，进行：

- 1) 检查程序功能能否按需求规格说明书的规定正常使用，测试各个功能是否有遗漏，检测性能等特性要求是否满足。
- 2) 检测人机交互是否错误，检测数据结构或外部数据库访问是否错误，程序是否能适当地接收输入数据而产生正确的输出结果，并保持外部信息（如数据库或文件）的完整性。
- 3) 检测程序初始化和终止方面的错误。

黑盒测试

□ 黑盒测试主要测试的错误类型有：

- ①不正确或遗漏的功能；
- ②接口、界面错误；
- ③性能错误；
- ④数据结构或外部数据访问错误；
- ⑤初始化或终止条件错误等等。

黑盒测试

□ 黑盒测试要考虑以下问题：

- 1) 如何测试功能的有效性？
- 2) 何种类型的输入会产生好的测试用例？
- 3) 系统是否对特定的输入值尤其敏感？
- 4) 如何分隔数据类的边界？
- 5) 系统能够承受何种数据率和数据量？
- 6) 特定类型的数据组合会对系统产生何种影响？

黑盒测试

□ 黑盒测试的应用

■ 运用黑盒测试方法，可以导出满足以下标准的测试用例集：

- 1) 所设计的测试用例能够减少达到合理测试所需的附加测试用例数；
- 2) 所设计的测试用例能够告知某些类型错误的存在或不存在，而不是仅仅与特定测试相关的错误。

■ 黑盒测试又可分为功能测试和非功能测试

功能测试主要有：等价类划分、边界值分析、因果图法、判定表法、场景法、正交实验法，随机测试法、错误推测法等。

非功能测试主要有：配置/安装测试、兼容性测试、互操作性测试、文档和帮助测试、性能测试、可靠性测试、易用性测试和界面测试等。

黑盒测试

□ 穷举测试的不可能性

理论上，用黑盒测试发现程序中的错误，必须把所有可能的输入和输出作为测试数据考虑，来检查程序是否都能产生正确结果。

但这是不可能的，因为穷举测试数量太大，无法完成。如

黑盒测试

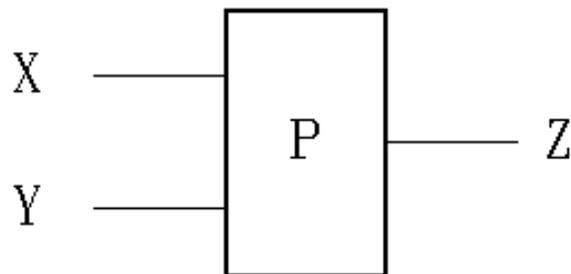
假设一个程序P有输入量X和Y及输出量Z。在字长为32位的计算机上运行。若X、Y取整数，按黑盒方法进行穷举测试：

可能采用的测试数据组：

$$2^{32} \times 2^{32} = 2^{64}$$

如果测试一组数据需要1毫秒，一年工作365×24小时，完成所有测试需5亿年。

因此，我们只能在大量可能的数据中，选取其中一部分作为测试用例。



黑盒测试的数学知识

□ 集合论

1 集合与成员

$$M_{30} = \{ 4月, 6月, 9月, 11月 \}$$

2 集合的表示

列出集合元素 如 M_{30}

给出集合元素判定规则

$$\text{如 } Y = \{ \text{年} \mid 1900 < \text{年} < 2199 \}$$

3 空集的表示

Φ

$$\Phi = \{ \text{年} : 2012 \leq \text{年} \leq 1812 \}$$

注: Φ , $\{ \phi \}$ 是不同的。

数学知识 - 集合论

4 维恩图(Venn Diagram)

The Venn Diagram is made up of two or more overlapping circles. It is often used in mathematics to show relationships between sets

5 集合的关系

A 是 B 的子集： $A \subseteq B$

A 是 B 的真子集： $A \subset B$

A 和 B 是相等集合： $A = B$

数学知识 - 集合论

6 集合的划分

若 A_1, A_2, \dots, A_n 是集合 A 的子集, 则

A_1, A_2, \dots, A_n 是集合 A 的一个划分, 当且仅当

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

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

测试 (1) 完备性

(2) 无冗余性

数学知识 - 函数

□ 函数映射关系

任何程序都可以看成将其输出与输入关联起来的函数，因此函数是开发测试的核心概念。

1-1函数

多对1函数

程序实现的功能大多数是多对一的函数。这对测试很重要（多对一测试可选代表等价类1对1，功能相似也可分等价类）

数学知识 - 命题逻辑

□ 命题：有真假值，且无歧义

P	q	$p \wedge q$	$p \vee q$	$\neg p$	$p \oplus q$
T	T	T	T	F	F
T	F	F	T	F	T
F	T	F	T	T	T
F	F	F	F	T	F

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
- 因果图方法
- 判定表驱动分析方法
- 场景设计方法
- 其他几种方法
- 功能性测试小结

等价类划分

- 概述
- 等价类的划分原则
- 等价类划分法的测试用例设计
- 常见等价类划分测试形式
- 使用等价类划分法测试的实例

等价类划分（概述）

- 等价类划分法是一种重要的、常用的黑盒测试方法，它将不能穷举的测试过程进行合理分类，从而保证设计出来的测试用例具有完整性和代表性。
- 等价类划分法
是把所有可能的输入数据，即程序的输入域划分为若干个等价类（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。
- 等价类
所谓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，它们具有等价特性。因此，可以合理的假定：测试某等价类的代表值就是等效于对于这一类其它值的测试。

等价类划分法（续）

例：设计这样的测试用例，来实现一个对所有实数进行开平方运算（ $y = \text{sqrt}(x)$ ）的程序的测试。

➤思考方向：

由于开平方运算只对非负实数有效，这时需要将所有的实数（输入域 x ）进行划分，可以分成：正实数、0 和 负实数。假设我们选定+1.4444代表正实数，-2.345代表负实数，则为该程序设计的测试用例的输入为+1.4444、0 和 -2.345。

3.4.1 等价类的划分原则

□ 等价类特性

等价类是输入域的某个子集合，而所有等价类的并集就是整个输入域。因此，等价类对于测试有两个重要的意义：

- 完备性——整个输入域提供一种形式的完备性
- 无冗余性——若互不相交则可保证一种形式的无冗余性

□ 如何划分？

划分等价类重要的是：集合的划分，划分为互不相交的一组子集，而子集的并是整个集合。

先从程序的规格说明书中找出各个输入条件，再为每个输入条件划分多个等价类，形成若干互不相交的子集。再在同一类中标识（选择）一个测试用例。（同一等价类中，往往处理相同，相同处理映射到“相同的执行路径”）

等价类划分

□ 采用等价类划分法设计测试用例通常分两步进行：

- 1) 确定等价类，列出等价类表。
- 2) 确定测试用例。

□ 划分等价类可分为两种情况：

(1) 有效等价类

是指对软件规格说明而言，是有意义的、合理的输入数据所组成的集合。

利用有效等价类，能够检验程序是否实现了规格说明中预先规定的功能和性能。

等价类的划分原则（续）

（2）无效等价类

是指对软件规格说明而言，是无意义的、不合理的输入数据所构成的集合。

利用无效等价类，可以鉴别程序异常处理的情况，检查被测对象的功能和性能的实现是否有不符合规格说明要求的地方。

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.

等价类的划分原则（续）

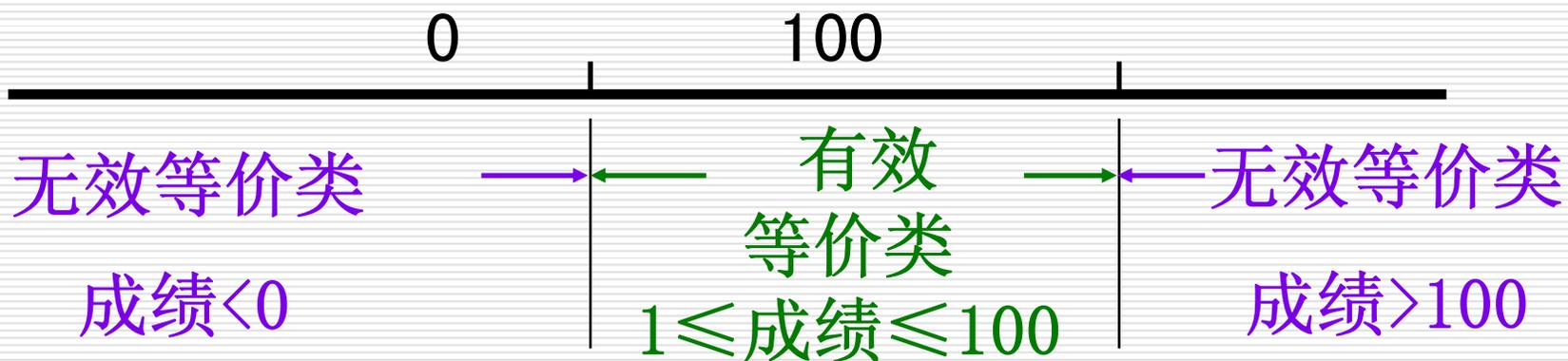
□ 划分等价类的方法

下面给出六条确定等价类的原则

①按照区间划分

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。

如：输入值是学生成绩，范围是0~100



等价类的划分原则（续）

②按照输入限制划分

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
的情况下,可确立一个有效等价类和一个无效等价类。

如规定输入必须为数字,可确定一个有效等价类: 数字和一个
无效等价类: 字母。

③按照输入布尔量划分

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。

等价类的划分原则（续）

④按照数值划分

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（每个值确定一个有效等价类）和一个无效等价类（所有不允许的输入值的集合）。

例：输入条件说明学历可为：专科、本科、硕士、博士四种之一，则分别取这四种这四个值作为四个有效等价类，另外把四种学历之外的任何学历作为无效等价类。

等价类的划分原则（续）

⑤按照限制条件或规则划分

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）。

如：程序中，成绩输入为0~100的整数，可确立一个有效等价类：89，然后再确定若干个无效等价类：89.1，-1，101，x等

等价类的划分原则（续）

⑥ 细分等价类

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。

如：程序中，成绩输入为0~100的整数，程序对不同的分数进行优、良、中、及格等的转换。

除按第一条原则划分，还可进一步划分为更小的等价类：
85~100（优），75~80（良）等。

等价类划分法的测试用例设计

□ 设计测试用例

- 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
...
...

- 在设计测试用例时,应同时考虑有效等价类和无效等价类测试用例的设计。

等价类划分法的测试用例设计

- 然后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。

常见等价类划分测试形式

- 针对缺陷相关性假设，可将等价类测试分为弱等价类测试(单缺陷假设)和强等价类测试(多缺陷假设)。
- 针对是否对无效数据进行测试，可以将等价类测试分为：
 - 标准等价类测试——不考虑无效数据值，测试用例使用每个等价类中的一个值。
 - 健壮等价类测试——主要的出发点是考虑了无效等价类。对有效输入，测试用例从每个有效等价类中取一个值；对无效输入，一个测试用例有一个无效值，其他值均取有效值。

等价类划分举例

□ 举例

例1 给定一个两变量 x_1 和 x_2 的函数 f ，如果 f 实现为一程序，则输入变量 x_1 和 x_2 将拥有以下边界及边界内的区间：

$a \leq x_1 \leq d$ ，区间为 $[a,b]$ ， $[b,c]$ ， $[c,d]$

$e \leq x_2 \leq g$ ，区间为 $[e,f]$ ， $[f,g]$

给出其强、弱、健壮等价类测试。

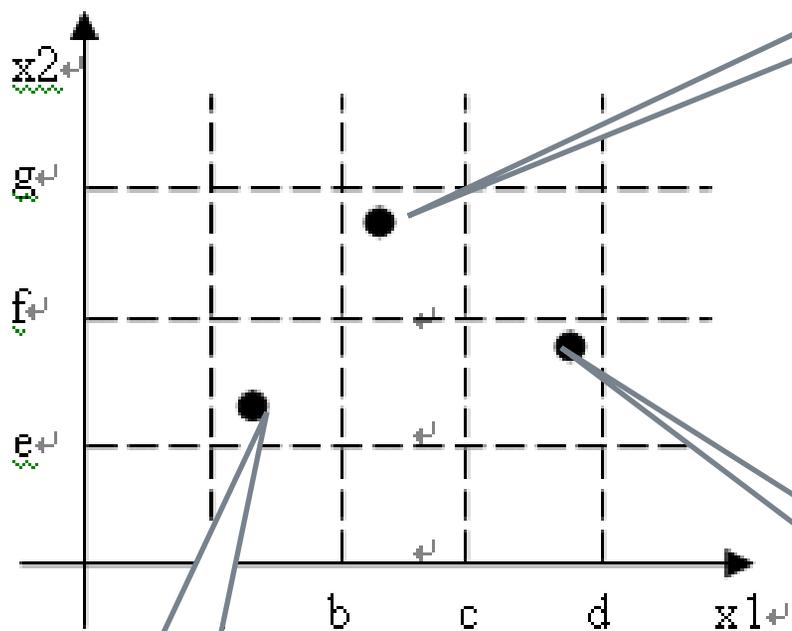
等价类划分举例

- 确立等价类，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
X1	1) $a \leq x1 < b$ 2) $B \leq x1 \leq c$ 3) $C \leq x1 \leq d$	6) $x1 < a$ 7) $x1 > d$
x2	4) $e \leq x2 < f$ 5) $f \leq x2 < g$	8) $X2 < e$ 9) $x2 > g$

等价类划分举例

弱一般等价类测试



输入条件	有效等价类	无效等价类
x1	1) $a \leq x1 < b$ 2) $B \leq x1 \leq c$ 3) $C \leq x1 \leq d$	6) $x1 < a$ 7) $x1 > d$
x2	4) $e \leq x2 < f$ 5) $f \leq x2 < g$	8) $X2 < e$ 9) $x2 > g$

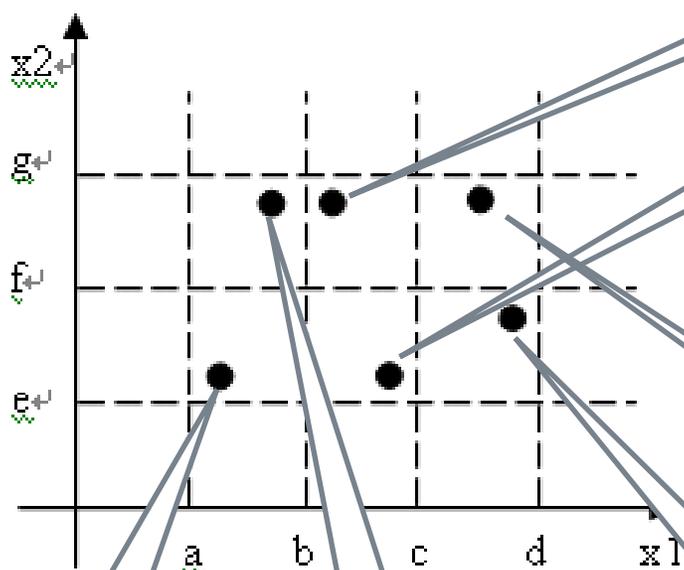
覆盖1, 4

覆盖2, 5

覆盖3, 4

等价类划分举例

强一般等价类测试



输入条件	有效等价类	无效等价类
X1	1) $a \leq x1 < b$ 2) $B \leq x1 \leq c$ 3) $C \leq x1 \leq d$	6) $x1 < a$ 7) $x1 > d$
X2	4) $e \leq x2 < f$ 5) $f \leq x2 < g$	8) $X2 < e$ 9) $x2 > g$

覆盖1, 4

覆盖1, 5

覆盖3, 4

覆盖3, 5

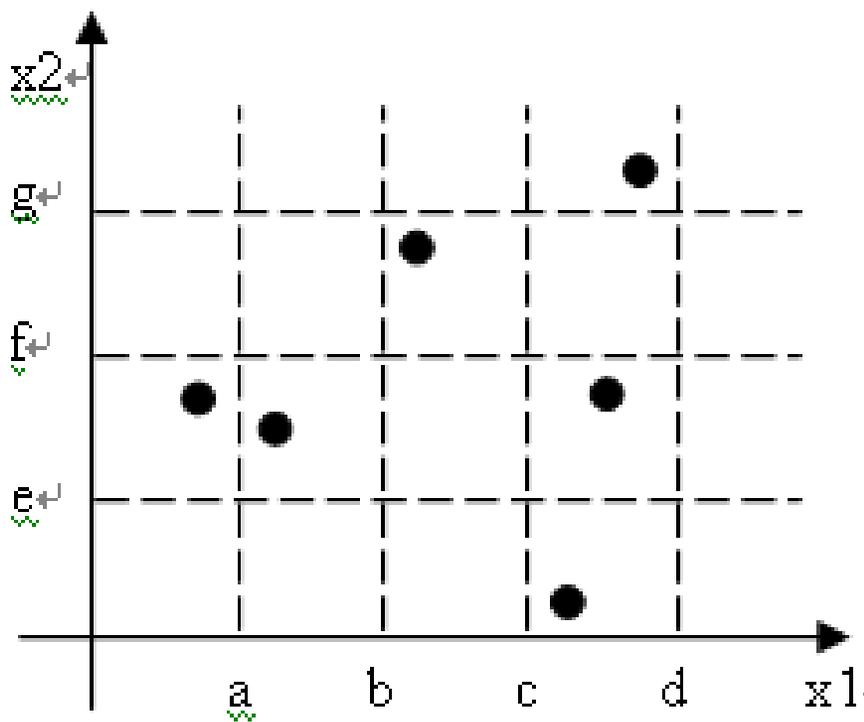
覆盖2, 5

覆盖2, 4

等价类划分举例

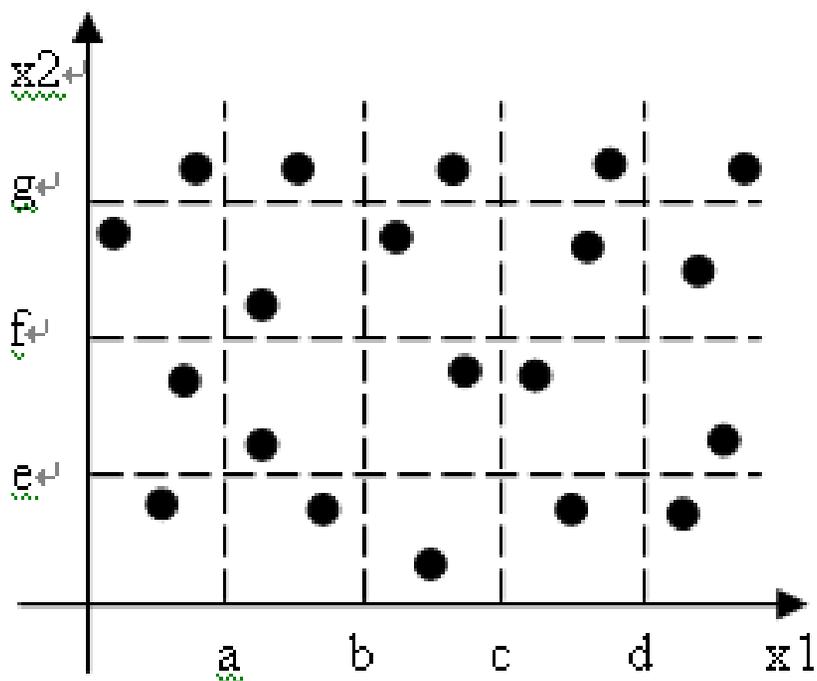
□ 弱健壮等价类测试

右图有遗漏，
遗漏在哪？



等价类划分举例

□ 强健壮等价类测试



输入条件	有效等价类	无效等价类
x1	1) $a \leq x1 < b$ 2) $B \leq x1 \leq c$ 3) $C \leq x1 \leq d$	6) $x1 < a$ 7) $x1 > d$
x2	4) $e \leq x2 < f$ 5) $f \leq x2 < g$	8) $X2 < e$ 9) $x2 > g$

等价类划分举例

- 例2：设有一个档案管理系统，要求用户输入以年月表示的日期。假设日期限定在1990年1月~2049年12月，并规定日期由6位数字字符组成，前4位表示年，后2位表示月。现用等价类划分法设计测试用例，来测试程序的“日期检查功能”。

等价类划分举例

1) 划分等价类并编号

下表等价类划分的结果

输入等价类	有效等价类	无效等价类
日期的类型及长度	①6位数字字符	②有非数字字符 ③少于6位数字字符 ④多于6位数字字符
年份范围	⑤在1990~2049之间	⑥小于1990 ⑦大于2049
月份范围	⑧在01~12之间	⑨等于00 ⑩大于12

等价类划分举例

2) 设计测试用例，以便覆盖所有有效等价类

在表中列出了3个有效等价类，编号分别为①、⑤、⑧，设计的测试用例如下：

测试数据	期望结果	覆盖的有效等价类
200211	输入有效	①、⑤、⑧

输入等价类	有效等价类	无效等价类
日期的类型及长度	①6位数字字符	②有非数字字符③…④…
年份范围	⑤在1990~2049之间	⑥小于1990 ⑦大于2049
月份范围	⑧在01~12之间	⑨等于00 ⑩大于12

等价类划分举例

3) 为每一个无效等价类设计一个测试用例，设计结果如下：

测试数据 期望结果 覆盖的无效等价类

95June 无效输入 ②

20036 无效输入 ③

2001006 无效输入 ④

198912 无效输入 ⑥

200401 无效输入 ⑦

200100 无效输入 ⑨

200113 无效输入 ⑩

输入等价类	无效等价类
日期的类型及长度	②有非数字字符 ③少于6位数字字符 ④多于6位数字字符
年份范围	⑥小于1990 ⑦大于2049
月份范围	⑨等于00 ⑩大于12

等价类划分举例

□ 例3 某城市的电话号码由三部分组成，这三部分的名称和内容分别是：

地区码：空白或3位数字；

前缀：非“0”和非“1”开头的3位数字；

后缀：4位数字

假定被测程序接受一切符合上述规定的电话号码，拒绝所有不符合的号码，用等价分类法来设计它的测试用例。

等价类划分举例

第一步 划分等价类。包括4个有效等价类，11个无效等价类。

输入条件	有效等价类	无效等价类
地区码	空白①, 3位数字②	有非数字字符⑤, 少于三位数字⑥, 多于三位数字⑦
前缀	200-999之间的三位数字③	有非数字字符⑧, 起始位为0⑨, 起始位为1⑩, 少于三位数字11, 多于三位数字12
后缀	4位数字④	有非数字字符13, 少于4位数字14, 多于4位数字15

等价类划分举例

□ 第二步 设计测试用例

测试数据	测试范围	期望结果
() 276-2345	等价类①, ③, ④	有效
(635) 805-9321	等价类②, ③, ④	有效

输入条件	有效等价类
地区码	空白①, 3位数字②
前缀	200-999之间的 三位数字 ③
后缀	4位数字④

对11个无效等价类，选择11个测试用例如下：

等价类划分举例

测试数据	测试范围	期望结果
(20A) 423-4567	无效等价类⑤	无效
(33) 234-5678	无效等价类⑥	无效
(7777) 345-6789	无效等价类⑦	无效
(111) P12-9876	无效等价类⑧	无效
(111) 021-5432	无效等价类⑨	无效
(222) 121-1378	无效等价类⑩	无效
(333) 52-1378	无效等价类11	无效
(333) 6751-1378	无效等价类12	无效
(027) 345-6B01	无效等价类13	无效
(020) 345-601	无效等价类14	无效
(031) 478-23452	无效等价类15	无效

等价类划分举例

□ 例4 保险公司计算保费费率的程序

某保险公司的人寿保险的保费计算方式为：

投保额 × 保险费率

其中，保险费率依点数不同而有别，10点及10点以上保险费率为0.6%，10点以下保险费率为0.1%；而点数又是由投保人的年龄、性别、婚姻状况和抚养人数来决定，具体规则如下：

年龄			性别		婚姻		抚养人数
20~39	40~59	其它	M	F	已婚	未婚	1人扣0.5点 最多扣3点 (四舍五入取整)
6点	4点	2点	5点	3点	3点	5点	

对其保险费率的计算进行测试

计算保费费率的程序

1) 分析程序规格说明中给出和隐含的对输入条件的要求，列出等价类表（包括有效等价类和无效等价类）。

- 年龄：一位或两位非零整数，值的有效范围为1~99
- 性别：一位英文字符，只能取值'M'或'F'
- 婚姻：字符，只能取值'已婚'或'未婚'
- 抚养人数：空白或一位非零整数（1~9）
- 点数：一位或两位非零整数，值的范围为1~99

等价类表见下页

2) 根据1) 中的等价类表，设计能覆盖所有等价类的测试用例。

输入条件	有效等价类	编号	无效等价类	编号
年龄	20~39岁	1		
	40~59岁	2		
	1~19岁 60~99岁	3	小于1	12
			大于99	13
性别	单个英文字符	4	非英文字符	14
			非单个英文字符	15
	'M'	5	除'M'和'F'之外的 其它单个字符	16
	'F'	6		
婚姻	已婚	7	除'已婚'和'未婚'之外的 其它字符	17
	未婚	8		
抚养人数	空白	9	除空白和数字之外的 其它字符	18
	1~6人	10	小于1	19
	6~9人	11	大于9	20

测试用例 编号	输入数据				预期输出
	年龄	性别	婚姻	抚养人数	保险费率
1	27	F	未婚	空白	0.6%
2	50	M	已婚	2	0.6%
3	70	F	已婚	7	0.1%
4	0	M	未婚	空白	无法推算
5	100	F	已婚	3	无法推算
6	99	男	已婚	4	无法推算
7	1	Child	未婚	空白	无法推算
8	45	N	已婚	5	无法推算
9	38	F	离婚	1	无法推算
10	62	M	已婚	没有	无法推算
11	18	F	未婚	0	无法推算
12	40	M	未婚	10	无法推算

使用等价类划分法测试的实例

例5：输入三个整数a、b、c(假定在1~100之间)，分别作为三角形的三条边，现通过程序判断由三条边构成的三角形的类型为等边三角形、等腰三角形、一般三角形（特殊的还有直角三角形），以及构不成三角形。

用等价类划分方法为该程序进行测试用例设计。

（三角形问题的复杂之处在于输入与输出之间的关系比较复杂。）

使用等价类划分法测试的实例

□ 解法一：分析题目中给出和隐含的对输入条件的要求：

- (1) 整数 (2) 三个数 (3) 非零数
- (4) 正数 (5) 两边之和大于第三边
- (6) 等腰 (7) 等边

如果 a 、 b 、 c 满足条件 (1) ~ (4)，则输出下列四种情况之一：

1. 如果不满足条件 (5)，则程序输出为“非三角形”。
2. 如果三条边相等即满足条件 (7)，则程序输出为“等边三角形”。
3. 如果只有两条边相等、即满足条件 (6)，则程序输出为“等腰三角形”。
4. 如果三条边都不相等，则程序输出为“一般三角形”。

列出等价类表并编号

有效等价类型	号码	无效等价类		号码	
		描述	号码		
整数	1	一边为非整数	a 为非整数	12	
			b 为非整数	13	
			c 为非整数	14	
		两边为非整数	a, b 为非整数	15	
			b, c 为非整数	16	
			a, c 为非整数	17	
	三边 a, b, c 均为非整数	18			
	三个数	2	只给一边	只给 a	19
				只给 b	20
				只给 c	21
		只给两边	只给 ab	22	
			只给 b, c	23	
			只给 ac	24	
		给出三个以上	25		
	非零数	3	一边为零	a 为 0	26
				b 为 0	27
				c 为 0	28
		二边为零	a, b 为 0	29	
b, c 为 0			30		
a, c 为 0			31		
三边 a, b, c 均为 0		32			
正数	4	一边 < 0	a < 0	33	
			b < 0	34	
			c < 0	35	
	二边 < 0	a < 0 且 b < 0	36		
		a < 0 且 c < 0	37		
		b < 0 且 c < 0	38		
	三边均 < 0: a < 0 且 b < 0 且 c < 0	39			

使用等价类划分法测试的实例

等价类表并编号（续上表）

输出条件	构成一般三角形	$a+b>c$	5	$\left\{ \begin{array}{l} a+b<0 \\ a+b=0 \\ b+c<a \\ b+c=a \\ a+c<b \\ a+c=b \end{array} \right.$	40
		$b+c>a$	6		41
		$a+c>b$	7		42
					43
					44
					45
输出条件	构成等腰三角形	$a=b$	$\left. \begin{array}{l} a=b \\ b=c \\ a=c \end{array} \right\} \begin{array}{l} \text{且两边} \\ \text{之和} \\ \text{大于第} \\ \text{三边} \end{array}$	8	
		$b=c$		9	
		$a=c$		10	
输出条件	构成等腰三角形	$a=b=c$	11		

边

使用等价类划分法测试的实例

覆盖有效等价类的测试用例：

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)

使用等价类划分法测试的实例

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
3.5	4.5	5	15	0	0	0	32
3	4.5	5.5	16	-3	4	5	33
3.5	4	5.5	17	3	-4	5	34
4.5	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		22	-3	-4	-5	39
	4	5	23	3	1	5	40
3		5	24	3	2	5	41
3	4	5	25	3	1	1	42
0	4	5	26	3	2	1	43
3	0	5	27	1	4	2	44
3	4	0	28	3	4	1	45

使用等价类划分法测试的实例

- 解法2：在多数情况下，是从输入域划分等价类的，但并非不能从被测程序的输出域反过来定义等价类，事实上，这对于三角形问题却是最简单的划分方法。

在三角形问题中，有四种可能的输出：等边三角形、等腰三角形、一般三角形和非三角形。利用这些信息能够确定下列输出（值域）等价类。

$R1 = \{ \langle a,b,c \rangle: \text{边为} a,b,c \text{的等边三角形} \}$

$R2 = \{ \langle a,b,c \rangle: \text{边为} a,b,c \text{的等腰三角形} \}$

$R3 = \{ \langle a,b,c \rangle: \text{边为} a,b,c \text{的一般三角形} \}$

$R4 = \{ \langle a,b,c \rangle: \text{边为} a,b,c \text{不能组成三角形} \}$

使用等价类划分法测试的实例

测试用例	a	b	c	预期输出
Test1	10	10	10	等边三角形
Test2	10	10	5	等腰三角形
Test3	3	4	5	一般三角形
Test4	4	1	2	非三角形

三角形问题的标准等价类测试用例

使用等价类划分法测试的实例

测试用例	a	b	c	预期输出
Test11	-1	5	5	a值超出输入值定义域
Test12	5	-1	5	b值超出输入值定义域
Test13	5	5	-1	c值超出输入值定义域
Test14	101	5	5	a值超出输入值定义域
Test15	5	101	5	b值超出输入值定义域
Test16	5	5	101	c值超出输入值定义域

三角形问题的弱健壮等价类测试用例

NextDate 函数等价类测试用例

- 例6: NextDate 函数包含三个变量: month、day 和 year, 函数的输出为输入日期后一天的日期。例如, 输入为 2006年3月7日, 则函数的输出为 2006年3月8日。

要求输入变量 month、day 和 year 均为整数值, 并且满足下列条件:

1) $1 \leq \text{month} \leq 12$

2) $1 \leq \text{day} \leq 31$

3) $1812 \leq \text{year} \leq 2012$

- 有效等价类为:

$$M_1 = \{ \text{月份}: 1 \leq \text{月份} \leq 12 \}$$

$$D_1 = \{ \text{日期}: 1 \leq \text{日期} \leq 31 \}$$

$$Y_1 = \{ \text{年}: 1812 \leq \text{年} \leq 2012 \}$$

NextDate 函数等价类测试用例

- 若条件（1）~（3）中任何一个条件失效，则 NextDate 函数都会产生一个输出，指明相应的变量超出取值范围，比如“month 的值不在 1-12 范围当中”。显然还存在着大量的 year、month、day 的无效组合，NextDate 函数将这些组合作统一的输出：“无效日期”。其无效等价类为：

$$M_2 = \{\text{月份: 月份} < 1\}$$

$$M_3 = \{\text{月份: 月份} > 12\}$$

$$D_2 = \{\text{日期: 日期} < 1\}$$

$$D_3 = \{\text{日期: 日期} > 31\}$$

$$Y_2 = \{\text{年: 年} < 1812\}$$

$$Y_3 = \{\text{年: 年} > 2012\}$$

NextDate 函数等价类测试用例

输入条件	有效等价类	无效等价类
月份	$M_1) 1 \leq \text{月份} \leq 12$	$M_2) \text{月份} < 1$ $M_3) \text{月份} > 12$
日期	$D_1) 1 \leq \text{日期} \leq 31$	$D_2) \text{日期} < 1$ $D_3) \text{日期} > 31$
年	$Y_1) 1812 \leq \text{年} \leq 2012$	$Y_2) \text{年} < 1812$ $Y_3) \text{年} > 2012$

NextDate 函数等价类测试用例

□ 弱一般等价类测试用例

月份	日期	年	预期输出
6	15	1912	1912年6月16日

□ 强一般等价类测试用例同弱一般等价类测试用例

注：弱——有单缺陷假设

健壮——考虑了无效值

输入条件	有效等价类	无效等价类
月份	$M_1) 1 \leq \text{月份} \leq 12$	$M_2) \text{月份} < 1$ $M_3) \text{月份} > 12$
日期	$D_1) 1 \leq \text{日期} \leq 31$	$D_2) \text{日期} < 1$ $D_3) \text{日期} > 31$
年	$Y_1) 1812 \leq \text{年} \leq 2012$	$Y_2) \text{年} < 1812$ $Y_3) \text{年} > 2012$

NextDate 函数等价类测试用例

□ 弱健壮等价类测试

用例ID	月份	日期	年	预期输出
WR ₁	6	15	1912	1912年6月16日
WR ₂	-1	15	1912	无效日期
WR ₃	13	15	1912	无效日期
WR ₄	6	-1	1912	无效日期
WR ₅	6	32	1912	无效日期
WR ₆	6	15	1811	无效日期
WR ₇	6	15	2013	无效日期

输入条件	无效等价类
月份	M ₂) 月份 < 1 M ₃) 月份 > 12
日期	D ₂) 日期 < 1 D ₃) 日期 > 31
年	Y ₂) 年 < 1812 Y ₁) 年 > 2012

NextDate 函数等价类测试用例

□ 强健壮等价类测试（27个中的一个“角”）

用例ID	月份	日期	年	预期输出
SR ₁	-1	15	1912	月份不在1~12中
SR ₂	6	-1	1912	日期不在1~31中
SR ₃	6	15	1811	年份不在1812~2012中
SR ₄	-1	-1	1912	两个无效一个有效
SR ₅	6	-1	1811	两个无效一个有效
SR ₆	-1	15	1811	两个无效一个有效
SR ₇	-1	-1	1811	三个无效

习题

问题：给出下面的有效和无效等价类

输入条件：“...统计全国各省、市、自治区的人口...”

输入条件：“标识符应以字母开头...”

输入条件：长度为1-20的字符串

输入条件：数据库中的值域, CHAR(20), NOT NULL

边界值分析法

□ 什么是边界值分析法

边界值分析法就是对输入或输出的边界值进行测试的一种黑盒测试方法。通常边界值分析法是作为对等价类划分法的补充，这种情况下，其测试用例来自等价类的边界。

□ 为什么使用边界值分析法？

无数的测试实践表明，大量的故障往往发生在输入定义域或输出值域的边界上，而不是在其内部。因此，针对各种边界情况设计测试用例，通常会取得很好的测试效果。

边界值分析法

□ 举例 —— 常见的边界值

- 对16-bit 的整数而言 32767 和 -32768 是边界
- 屏幕上光标在最左上、最右下位置
- 报表的第一行和最后一行
- 数组元素的第一个和最后一个
- 循环的第 0 次、第 1 次和倒数第 2 次、最后一次

边界值分析法

□ 怎样用边界值分析法设计测试用例？

- (1) 首先确定边界情况。通常输入或输出等价类的边界就是应该着重测试的边界情况。
- (2) 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。

边界值分析法

□ 与等价划分的区别

- 边界值分析使用与等价类划分法相同的划分，只是边界值分析假定错误更多地存在于划分的边界上，因此在等价类的边界上以及两侧的情况设计测试用例。
- 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。
- 边界值分析不仅考虑输入条件，还要考虑输出空间产生的测试情况。

边界值分析法

□ 例7：测试计算平方根的函数

输入/输出：实数；

规格说明：当输入一个0或比0大的数的时候，返回其正平方根；
当输入一个小于0的数时，显示错误信息并返回0。

■ 等价类划分

可以考虑作出如下划分：

输入：(i) ≥ 0 和 (ii) < 0

输出：(a) ≥ 0 和 (b) Error

测试用例有两个：

输入4，输出2。 对应于 (i) 和 (a) 。

输入-10，输出0和错误提示。对应于 (ii) 和 (b) 。

边界值分析法

- 边界值分析：

划分(i)的边界为0和最大正实数；划分(ii)的边界为最小负实数和0。由此得到以下测试用例：

输入 { 最小负实数 }

输入 { 绝对值很小的负数 }

输入 0

输入 { 绝对值很小的正数 }

输入 { 最大正实数 }

边界值分析法

□ 基于边界值分析方法选择测试用例的原则

1) 如果输入条件规定了值的范围,则应取刚达到这个范围的边界的值,以及刚刚超越这个范围边界的值作为测试输入数据。

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。

边界值分析法

2) 如果输入条件规定了值的个数,则用最大个数,最小个数,比最小个数少一,比最大个数多一的数作为测试数据。

比如,一个输入文件应包括1~255个记录,则测试用例可取1和255,还应取0及256等。

边界值分析法

3) 将规则1) 和2) 应用于输出条件, 即设计测试用例使输出值达到边界值及其左右的值。

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1165.24、还可取-0.01及1165.26等。

再如一程序属于情报检索系统, 要求每次“最少显示1条、最多显示4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。

边界值分析法

- 5) 如果程序的规格说明给出的输入域或输出域是有序集合（如有序表、顺序文件等），则应选取集合的第一个元素和最后一个元素作为测试用例。
- 6) 如果程序中使用了一个内部数据结构，则应当选择这个内部数据结构的边界上的值作为测试用例。
- 7) 分析规格说明，找出其它可能的边界条件。

边界值分析

□ 上述原则的一些说明

- 1) 在通常情况下，软件测试所包含的边界检验有几种类型：数字、字符、位置、质量、大小、速度、方位、尺寸、空间等。相应地，以上类型的边界值应该在：最大/最小、首位/末位、上/下、最快/最慢、最高/最低、最短/最长、空/满等情况下。如：

项	边界值	测试用例的设计思路
数值	最小值-1/最大值+1	假设某软件的数据输入域要求输入5位的数据值，可以使用10000作为最小值、99999作为最大值；然后使用刚好小于5位和大于5位的数值来作为边界条件。
空间	小于空余空间一点/大于满空间一点	例如在用U盘存储数据时，使用比剩余磁盘空间大一点（几KB）的文件作为边界条件。

边界值分析

2)内部边界值条件

多数情况下，边界值条件可从软件的规格说明或常识中得到，最终用户也很容易发现问题。然而，某些边界值条件是不需要呈现给用户的，用户很难注意到，但同时确实属于检验范畴内的边界条件，称为内部边界值条件或子边界值条件。内部边界值条件主要有下面几种：

- 数值的边界值检验
- 字符的边界值检验
- 其它边界值检验

如：

数值的边界值检验

- 计算机是基于二进制进行工作的，因此，软件的任何数值运算都有一定的范围限制。

计算机数值运算的范围

项	范围或值
位 (bit)	0 或 1
字节 (byte)	0 ~ 255
字 (word)	0~65535 (单字) 或 0~4294967295 (双字)
千 (K)	1024
兆 (M)	1048576
吉 (G)	1073741824

字符的边界值检验

- 在计算机软件中，字符也是很重要的表示元素，其中ASCII和Unicode是常见的编码方式。下表中列出了一些常用字符对应的ASCII码值。

字符	ASCII码值	字符	ASCII码值
空 (null)	0	A	65
空格 (space)	32	a	97
斜杠 (/)	47	Z	90
0	48	z	122
冒号 (:)	58	单引号 (')	96
@	64		

边界值分析

3) “单故障”假设

边界值分析法是基于可靠性理论中称为“单故障”的假设，即有两个或两个以上故障同时出现而导致软件失效的情况很少，也就是说，软件失效基本上是由单故障引起的。

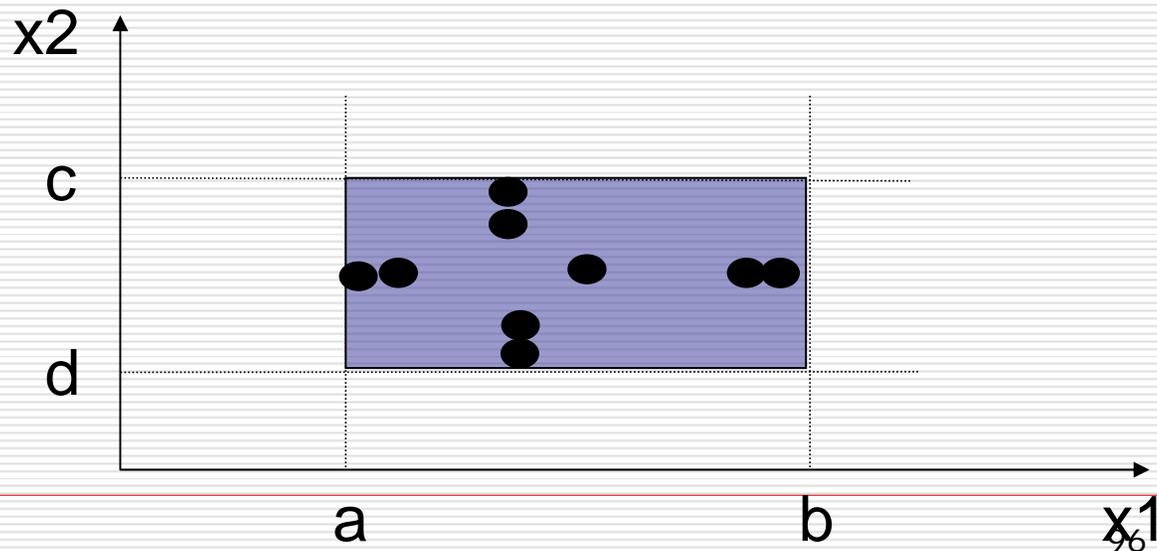
因此，在多变量的边界值分析法中获取测试用例的方法是：

- (1) 每次保留程序中一个变量，让其余的变量取正常值，被保留的变量依次取min、min+、nom、max-和max。
- (2) 对程序中的每个变量重复 (1) 。

边界值分析法 - 举例

□ 例8：有两个输入变量 x_1 ($a \leq x_1 \leq b$)和 x_2 ($c \leq x_2 \leq d$)的程序F的边界值分析测试用例如下：

{ $\langle x_1 \text{nom}, x_2 \text{min} \rangle$, $\langle x_1 \text{nom}, x_2 \text{min}+ \rangle$, $\langle x_1 \text{nom}, x_2 \text{nom} \rangle$,
 $\langle x_1 \text{nom}, x_2 \text{max} \rangle$, $\langle x_1 \text{nom}, x_2 \text{max}- \rangle$, $\langle x_1 \text{min}, x_2 \text{nom} \rangle$,
 $\langle x_1 \text{min}+, x_2 \text{nom} \rangle$, $\langle x_1 \text{max}, x_2 \text{nom} \rangle$, $\langle x_1 \text{max}-, x_2 \text{nom} \rangle$ }



边界值分析法 - 举例

- 例9：有二元函数 $f(x,y)$ ，其中 $x \in [1,12]$ ， $y \in [1,31]$ 。
则采用边界值分析法设计的测试用例是：

{ $\langle 1,15 \rangle$, $\langle 2,15 \rangle$, $\langle 11,15 \rangle$, $\langle 12,15 \rangle$, $\langle 6,15 \rangle$, $\langle 6,1 \rangle$,
 $\langle 6,2 \rangle$, $\langle 6,30 \rangle$, $\langle 6,31 \rangle$ }

- 推论：对于一个含有 n 个变量的程序，采用边界值分析法测试程序会产生 $4n+1$ 个测试用例（一个变量取最小值，略高于最小值，正常值，略低于最大值，最大值外，其余变量取正常值。对每个变量都重复进行）。

边界值分析法 - 举例

- 再如：有函数 $f(x,y,x)$ ，其中 $x \in [1900, 2100]$ ， $y \in [1, 12]$ ， $z \in [1, 31]$ 的。请写出该函数采用边界值分析法设计的测试用例。

共13个，分别是：{ $\langle 2000, 6, 1 \rangle$, $\langle 2000, 6, 2 \rangle$,
 $\langle 2000, 6, 30 \rangle$, $\langle 2000, 6, 31 \rangle$, $\langle 2000, 1, 15 \rangle$,
 $\langle 2000, 2, 15 \rangle$, $\langle 2000, 11, 15 \rangle$, $\langle 2000, 12, 15 \rangle$,
 $\langle 1900, 6, 15 \rangle$, $\langle 1901, 6, 15 \rangle$, $\langle 2099, 6, 15 \rangle$,
 $\langle 2100, 6, 15 \rangle$, $\langle 2000, 6, 15 \rangle$ }

边界值分析法 - 举例

例10：现有一个学生标准化考试批阅试卷,产生成绩报告的程序。其规格说明如下:程序的输入文件由一些有80个字符的记录组成,如右图所示,所有记录分为3组:

(试题部分)

标 题				
1				80
试题数		标准答案 (1~50 题)		2
1	3 4	9 10	59 60	79 80
试题数		标准答案 (51~100 题)		2
1	3 4	9 10	59 60	79 80

.....

(学生答卷部分)

学号 1		学生答案 (1~50 题)		3
1		9 10	59 60	79 80
学号 1		学生答案 (51~100 题)		3
1		9 10	59 60	79 80

.....

边界值分析法 - 举例

① 标题:

这一组只有一个记录，其内容为输出成绩报告的名字。

(试题部分)

标 题				
1				80
试题数		标准答案 (1~50 题)		2
1	3 4	9 10	59 60	79 80
试题数		标准答案 (51~100 题)		2
1	3 4	9 10	59 60	79 80

.....

(学生答卷部分)

学号 1		学生答案 (1~50 题)		3
1		9 10	59 60	79 80
学号 1		学生答案 (51~100 题)		3
1		9 10	59 60	79 80

.....

边界值分析法 - 举例

②试卷各题标准答案记录：
每个记录均在第80个字符处标以数字“2”。

该组的第一个记录的第1至第3个字符为题目编号（取值为1—999）。第10至第59个字符给出第1至第50题的答案（每个合法字符表示一个答案）。

该组的第2，第3……个记录相应为第51至第100，第101至第150，…题的答案。

(试题部分)

标 题			
1			80
试题数		标准答案 (1~50 题)	2
1	3 4	9 10	59 60 79 80
试题数		标准答案 (51~100 题)	2
1	3 4	9 10	59 60 79 80

(学生答卷部分)

学号 1		学生答案 (1~50 题)	3
1		9 10	59 60 79 80
学号 1		学生答案 (51~100 题)	3
1		9 10	59 60 79 80

边界值分析法 - 举例

③每个学生的答卷描述:

该组中每个记录的第80个字符均为数字“3”。

每个学生的答卷在若干个记录中给出。如甲的首记录第1至第9字符给出学生姓名及学号，第10至第59字符列出的是甲所做的第1至第50题的答案。

若试题数超过50，则第2，第3.....纪录分别给出他的第51至第100，第101至第150.....题的解答。

然后是学生乙的答卷记录。

(试题部分)

标 题				
1				80
试题数	标准答案 (1~50 题)			2
1	3 4	9 10	59 60	79 80
试题数	标准答案 (51~100 题)			2
1	3 4	9 10	59 60	79 80
.....				

(学生答卷部分)

学号 1	学生答案 (1~50 题)			3
1		9 10	59 60	79 80
学号 1	学生答案 (51~100 题)			3
1		9 10	59 60	79 80
.....				

学生人数不超过200，试题数不超过999。

边界值分析法 - 举例

程序的输出有4个报告：

- a)按学号排列的成绩单，列出每个学生的成绩、名次。
- b)按学生成绩排序的成绩单。
- c)平均分数及标准偏差的报告。
- d)试题分析报告。按试题号排序，列出各题学生答对的百分比。

解答：分别考虑输入条件和输出条件，以及边界条件。给出下表所示的输入条件及相应的测试用例。

边界值分析法 - 举例

输入条件	测试用例																				
输入文件	空输入文件																				
标题	(试题部分) 没有标题 标题只有一个字符 标题有 80 个字符 <table border="1" style="margin-left: 20px;"> <tr> <td colspan="2" style="text-align: center;">标 题</td> </tr> <tr> <td>1</td> <td></td> </tr> </table>	标 题		1																	
标 题																					
1																					
试题数	试题数为 1 试题数为 50 试题数为 51 试题数为 100 试题数为 0 试题数含有非数字字符 <table border="1" style="margin-left: 20px;"> <tr> <td>试题数</td> <td></td> <td>标准答案 (1~50 题)</td> <td></td> </tr> <tr> <td>1</td> <td>3 4</td> <td>9 10</td> <td>59 60</td> </tr> <tr> <td>试题数</td> <td></td> <td>标准答案 (51~100 题)</td> <td></td> </tr> <tr> <td>1</td> <td>3 4</td> <td>9 10</td> <td>59 60</td> </tr> <tr> <td colspan="4" style="text-align: right;">.....</td> </tr> </table>	试题数		标准答案 (1~50 题)		1	3 4	9 10	59 60	试题数		标准答案 (51~100 题)		1	3 4	9 10	59 60			
试题数		标准答案 (1~50 题)																			
1	3 4	9 10	59 60																		
试题数		标准答案 (51~100 题)																			
1	3 4	9 10	59 60																		
.....																					
标准答案记录	(学生答卷部分) 没有标准答案记录, 有标题 标准答案记录多于一个 标准答案记录少一个 <table border="1" style="margin-left: 20px;"> <tr> <td>学号 1</td> <td>学生答案 (1~50 题)</td> <td></td> </tr> <tr> <td>1</td> <td>9 10</td> <td>59 60</td> </tr> <tr> <td>学号 1</td> <td>学生答案 (51~100 题)</td> <td></td> </tr> <tr> <td>1</td> <td>9 10</td> <td>59 60</td> </tr> <tr> <td colspan="3" style="text-align: right;">.....</td> </tr> </table>	学号 1	学生答案 (1~50 题)		1	9 10	59 60	学号 1	学生答案 (51~100 题)		1	9 10	59 60							
学号 1	学生答案 (1~50 题)																				
1	9 10	59 60																			
学号 1	学生答案 (51~100 题)																				
1	9 10	59 60																			
.....																					
学生人数	0 个学生 1 个学生 200 个学生 201 个学生 <table border="1" style="margin-left: 20px;"> <tr> <td>学号 1</td> <td>学生答案 (51~100 题)</td> <td></td> </tr> <tr> <td>1</td> <td>9 10</td> <td>59 60</td> </tr> <tr> <td colspan="3" style="text-align: right;">.....</td> </tr> </table>	学号 1	学生答案 (51~100 题)		1	9 10	59 60													
学号 1	学生答案 (51~100 题)																				
1	9 10	59 60																			
.....																					

边界值分析法 - 举例

学生答题	<p>某学生只有一个回答记录，但有两个标准答案记录</p> <p>该学生是文件中的第一个学生</p> <p>该学生是文件中的最后一个学生（记录数出错的学生）</p>
学生答题	<p>某学生有两个回答记录，但只有一个标准答案记录</p> <p>该学生是文件中的第一个学生（记录数出错的学生）</p> <p>该学生是文件中的最后一个学生</p>
学生成绩	<p>所有学生的成绩都相等</p> <p>每个学生的成绩都不相等</p> <p>部分学生的成绩相同</p> <p>（检查是否能按成绩正确排名次）</p> <p>有个学生 0 分</p> <p>有个学生 100 分</p>

(试题部分)

标 题		
1		
试题数	标准答案 (1~50 题)	
1	3 4	9 10
试题数	标准答案 (51~100 题)	
1	3 4	9 10

(学生答卷部分)

学号 1	学生答案 (1~50 题)	
1		9 10
学号 1	学生答案 (51~100 题)	
1		9 10

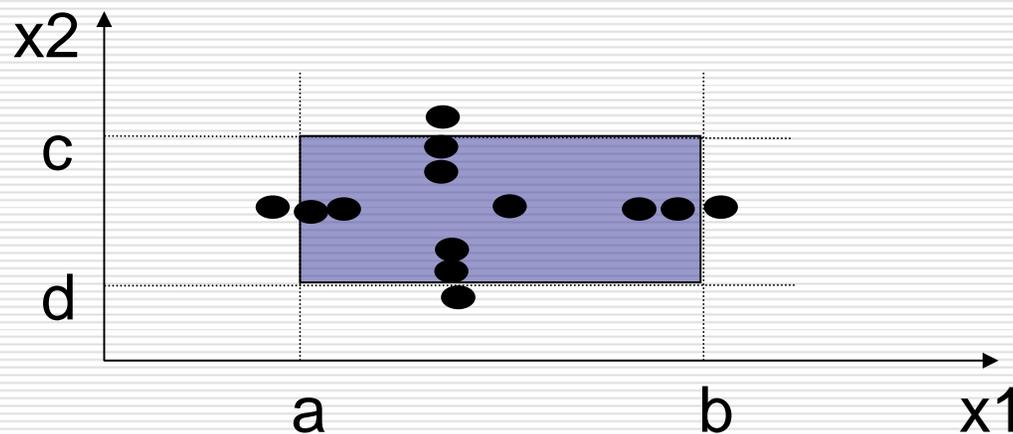
边界值分析法 - 举例

右表为输出条件及相应的测试用例表。

输出条件	测试用例
输出报告 a、b	有个学生的学号最小（检查按序号排序是否正确） 有个学生的学号最大（检查按序号排序是否正确） 适当的学生人数，使产生的报告刚好满一页（检查打印页数） 学生人数比刚才多出 1 人（检查打印换页）
输出报告 c	平均成绩 100 平均成绩 0 标准偏差为最大值（有一半的 0 分，其他 100 分） 标准偏差为 0（所有成绩相等）
输出报告 d	所有学生都答对了第一题 所有学生都答错了第一题 所有学生都答对了最后一题 所有学生都答错了最后一题 选择适当的试题数，是第四个报告刚好打满一页 试题数比刚才多 1，使报告打满一页后，刚好剩下一题未打

健壮性测试

- 健壮性测试是作为边界值分析的一个简单的扩充，它除了对变量的5个边界值分析取值外，还需要增加一个略大于最大值(max+)以及略小于最小值(min-)的取值，检查超过极限值时系统的情况。因此，对于有n个变量的函数采用健壮性测试需要 $6n+1$ 个测试用例。
- 前面例8中的程序F的健壮性测试如下图所示：



- 练习：请为例9中的函数 $f(x,y)$ 写出相应的健壮性测试用例。

边界值分析法 - 举例

□ 例11 三角形问题的边界值分析测试用例

在三角形问题描述中，三角形每边边长的取范围值设值为[1, 100]。

测试用例	a	b	c	预期输出
Test 1	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

边界值分析法 - 举例

□ 例12 NextDate函数的健壮性边界值分析测试用例

在NextDate函数中，隐含规定了变量month和变量day的取值范围为 $1 \leq \text{month} \leq 12$ 和 $1 \leq \text{day} \leq 31$ ，并设定变量year的取值范围为 $1912 \leq \text{year} \leq 2050$ 。

测试用例	mouth	day	year	预期输出
Test 1	6	15	1911	1911.6.16
Test2	6	15	1912	1912.6.16
Test3	6	15	1913	1913.6.16
Test4	6	15	1975	1975.6.16
Test5	6	15	2049	2049.6.16
Test6	6	15	2050	2050.6.16
Test7	6	15	2051	2051.6.16
Test8	6	-1	2001	day超出[1...31]
Test9	6	1	2001	2001.6.2
Test10	6	2	2001	2001.6.3
Test11	6	30	2001	2001.7.1
Test12	6	31	2001	输入日期超界
Test13	6	32	2001	day超出[1...31]
Test14	-1	15	2001	Mouth超出[1...12]
Test15	1	15	2001	2001.1.16
Test16	2	15	2001	2001.2.16
Test17	11	15	2001	2001.11.16
Test18	12	15	2001	2001.12.16
Test19	13	15	2001	Mouth超出[1...12]

边界值分析法

□ 最坏情况测试

软件失效是由多故障引起的。

对每个变量首先进行包含最小值、略高于最小值、正常值、略低于最大值和最大值五元素集合的测试，然后对这些集合进行笛卡儿乘积计算，以生成测试用例。

如：三角形问题和NextDate函数问题的最坏情况测试有：
 $5 \times 5 \times 5 = 125$ 种测试用例。

N个变量的最坏情况测试会产生 5^n 个测试用例

N个变量的健壮最坏情况测试会产生 7^n 个测试用例

习题

□ 找零钱最佳组合

假设商店货品价格(R) 都不大于100元（且为整数），若顾客付款(P)在100元内，现有一个程序能在每位顾客付款后给出找零钱的最佳组合（找给顾客货币张数最少）。假定此商店的货币面值只包括：50元(N50)、10元(N10)、5元(N5)、1元(N1) 四种。

请结合等价类划分法和边界值分析法为上述程序设计 出相应的测试用例。

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。

- 例如,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。

错误推测法

- 例如，针对例10，采用错误推测法还可补充设计一些测试用例：
 - 1、 程序是否把空格作为回答
 - 2、 在回答记录中混有标准答案记录
 - 3、 除了标题记录外，还有一些的记录最后一个字符即不是2也不是3
 - 4、 有两个学生的学号相同
 - 5、 试题数是负数。

错误推测法

- 再如，测试一个对线性表（比如数组）进行排序的程序，可推测列出以下几项需要特别测试的情况：
 - 1) 输入的线性表为空表；
 - 2) 表中只含有一个元素；
 - 3) 输入表中所有元素已排好序；
 - 4) 输入表已按逆序排好；
 - 5) 输入表中部分或全部元素相同。

因果图方法

- [因果图法简介](#)
- [因果图介绍](#)
- [因果图法测试举例](#)

因果图法的简介

□ 因果图法简介

- 等价类划分方法和边界值分析方法，都未过多考虑输入条件之间的联系，相互组合、相互制约等。
- 考虑输入条件间的相互组合不是一件易事，即使把所有输入条件划分成等价类，他们之间的组合情况也相当多。因此必须考虑采用一种适合于描述对于多种条件的组合，相应产生多个动作的形式来考虑设计测试用例。这就需要利用因果图（Cause—Effect Graphics）方法。
- 采用因果图方法能够帮助我们按一定步骤，高效率地选择测试用例，同时还能为我们指出，程序规格说明描述中存在着什么问题。

因果图法的简介（续）

□ 因果图法简介

- 因果图法是基于这样的一种思想：一些程序的功能可以用判定表（或称决策表）的形式来表示，并根据输入条件的组合情况规定相应的操作。
- 因果图法的定义：是一种利用图解法分析输入的各种组合情况，从而设计测试用例的方法，它适合于检查程序输入条件的各种组合情况。
- 采用因果图法设计测试用例的步骤：
 - 1) 根据程序规格说明书描述，分析并确定因（输入条件）和果（输出结果或程序状态的改变），画出因果图。
 - 2) 将得到的因果图转换为判定表。
 - 3) 为判定表中每一列所表示的情况设计一个测试用例。

因果图法的简介（续）

□ 使用因果图法的优点：

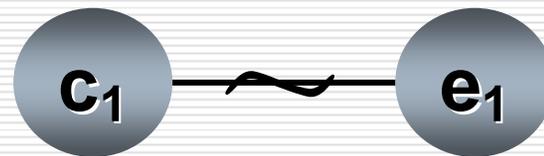
- 1) 考虑到了输入情况的各种组合以及各个输入情况之间的相互制约关系。
- 2) 能够帮助测试人员按照一定的步骤，高效率的开发测试用例。
- 3) 因果图法是将自然语言规格说明转化成形式语言规格说明的一种严格的方法，可以指出规格说明存在的不完整性和二义性。

因果图介绍

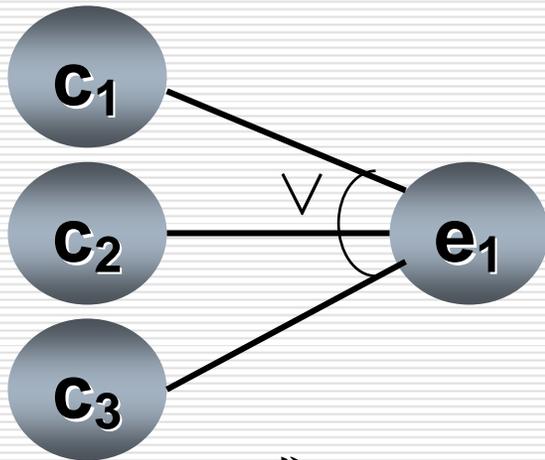
□ 4种符号分别表示了规格说明中的4种因果关系。



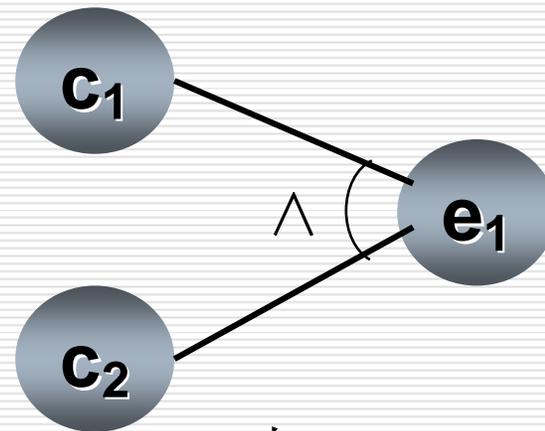
恒等



非



或



与

因果图介绍

- 因果图中使用了简单的逻辑符号，以直线联接左右结点。左结点表示输入状态（或称原因），右结点表示输出状态（或称结果）。
- C_i 表示原因，通常置于图的左部； e_i 表示结果，通常在图的右部。 c_i 和 e_i 均可取值0或1，0表示某状态不出现，1表示某状态出现。
- 关系
 - ①恒等：若 c_i 是1，则 e_i 也是1；否则 e_i 为0。
 - ②非：若 c_i 是1，则 e_i 是0；否则 e_i 是1。
 - ③或：若 c_1 或 c_2 或 c_3 是1，则 e_i 是1；否则 e_i 为0。“或”可有任意个输入。
 - ④与：若 c_1 和 c_2 都是1，则 e_i 为1；否则 e_i 为0。“与”也可有任意个输入。

因果图介绍

□ 约束

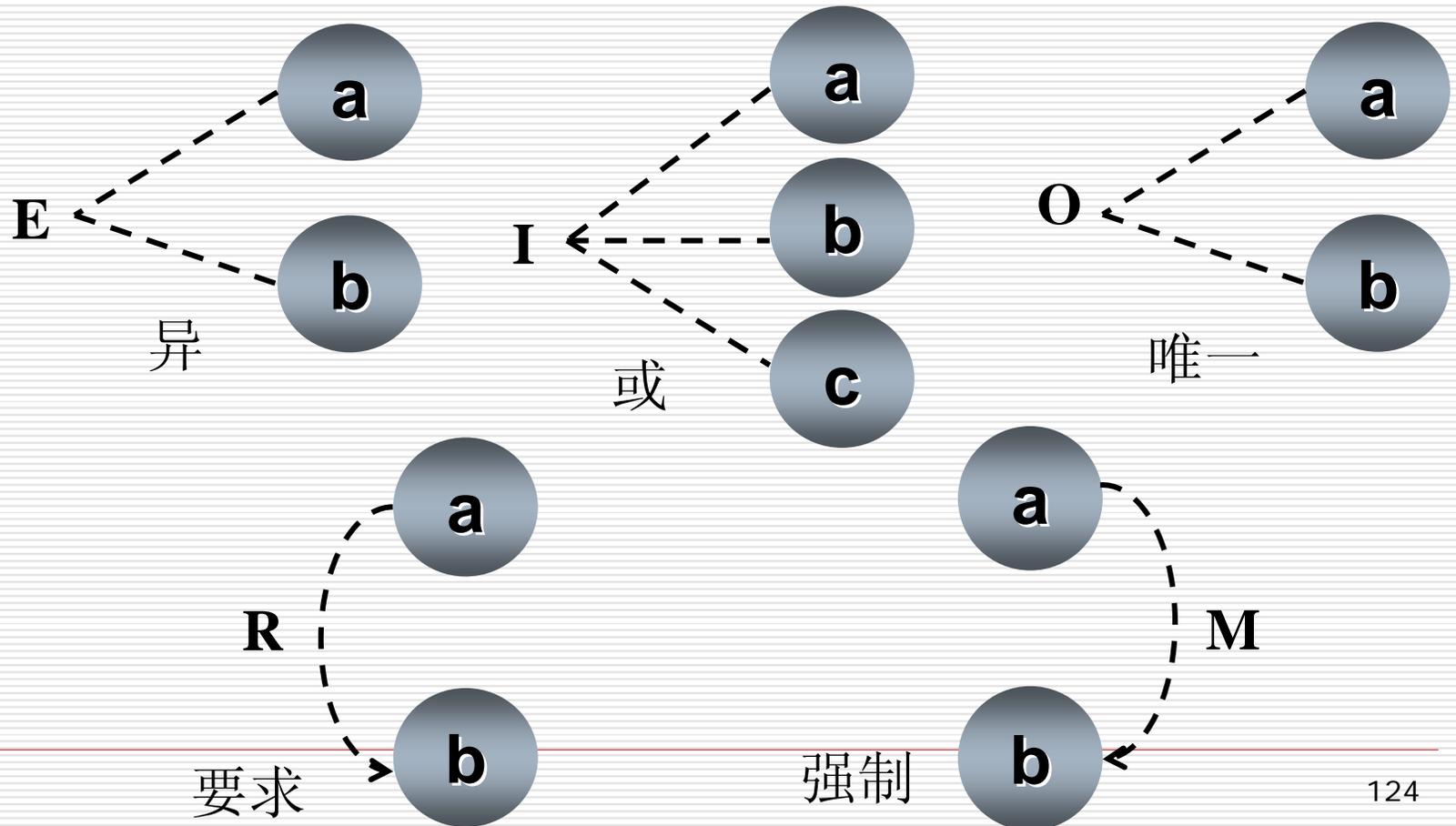
输入状态相互之间还可能存在某些依赖关系，称为约束。

如，某些输入条件本身不可能同时出现。

输出状态之间也往往存在约束。在因果图中,用特定的符号标明这些约束。

因果图介绍

□ 因果图中用来表示约束关系的约束符号：



因果图介绍

□ 输入条件的约束有以下4类：

- ① E约束（异）：a和b中至多有一个可能为1，即a和b不能同时为1。
- ② I约束（或）：a、b和c中至少有一个必须是1，即 a、b 和c不能同时为0。
- ③ O约束（唯一）；a和b必须有一个，且仅有1个为1。
- ④ R约束（要求）：a是1时，b必须是1，即不可能a是1时b是0。

□ 输出条件约束类型

输出条件的约束只有M约束（强制）：若结果a是1，则结果b强制为0。

因果图介绍

- 因果图方法最终生成的是判定表。它适合于检查程序输入条件的各种组合情况。利用因果图生成测试用例的基本步骤：
 - (1) 分析软件规格说明描述中, 那些是原因(即输入条件或输入条件的等价类), 那些是结果(即输出条件), 并给每个原因和结果赋予一个标识符。
 - (2) 分析软件规格说明描述中的内容, 找出原因与结果之间, 原因与原因之间对应的关系. 根据这些关系, 画出因果图。
 - (3) 由于语法或环境限制, 有些原因与原因之间, 原因与结果之间的组合情况不不可能出现. 为表明这些特殊情况, 在因果图上用一些约束符号表明约束或限制条件。
 - (4) 把因果图转换为判定表。
 - (5) 把判定表的每一列拿出来作为依据, 设计测试用例。

因果图法测试举例

□ 例13 用因果图法测试以下程序。

程序的规格说明要求：输入的第一个字符必须是#或*，第二个字符必须是一个数字，此情况下进行文件的修改；如果第一个字符不是#或*，则给出信息N，如果第二个字符不是数字，则给出信息M。

➤ 解题步骤：

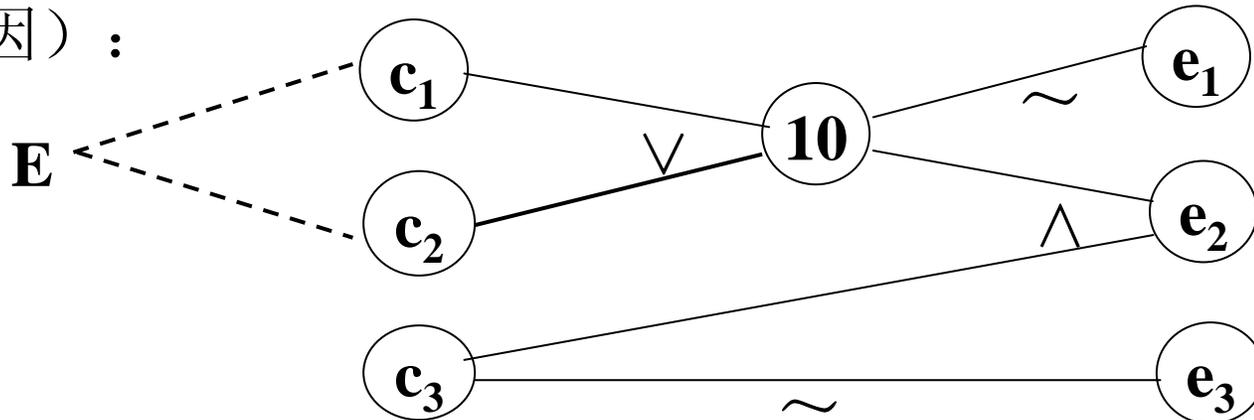
- (1) 分析程序的规格说明，列出原因和结果。
- (2) 找出原因与结果之间的因果关系、原因与原因之间的约束关系，画出因果图。
- (3) 将因果图转换成决策表。
- (4) 根据（3）中的决策表，设计测试用例的输入数据和预期输出。

因果图法测试举例（续）

(1) 分析程序规格说明中的原因和结果：

原因	结果
c1: 第一个字符是#	e1: 给出信息N
c2: 第一个字符是*	e2: 修改文件
c3: 第二个字符是一个数字	e3: 给出信息M

(2) 画出因果图（编号为10的中间结点是导出结果的进一步原因）：



因果图法测试举例（续）

(3) 将因果图转换成如下所示的决策表：

选项 \ 规则	1	2	3	4	5	6	7	8
条件(原因):								
C1	1	1	1	1	0	0	0	0
C2	1	1	0	0	1	1	0	0
C3	1	0	1	0	1	0	1	0
10			1	1	1	1	0	0
动作(结果):								
e1							√	√
e2			√		√			
e3				√		√		√
不可能	√	√						
测试用例			#3	#A	*6	*B	A1	GT

表中8种情况的左面两列情况中，原因C1和原因C2不可能同时为1，故应排除这两种情况。表的最下一栏给出了6种情况的测试用例，这是我們所需要的数据。

因果图法测试举例（续）

（4）根据决策表中的每一列设计测试用例：

测试用例编号	输入数据	预期输出
1	#3	修改文件
2	#A	给出信息M
3	*6	修改文件
4	*B	给出信息M
5	A1	给出信息N
6	GT	给出信息N和信息M

因果图方法举例

例14：有一个处理单价为5角钱的饮料的自动售货机软件测试用例的设计。其规格说明如下：若投入5角钱或1元钱的硬币，押下〔橙汁〕或〔啤酒〕的按钮，则相应的饮料就送出来。若售货机没有零钱找，则一个显示〔零钱找完〕的红灯亮，这时在投入1元硬币并押下按钮后，饮料不送出来而且1元硬币也退出来；若有零钱找，则显示〔零钱找完〕的红灯灭，在送出饮料的同时退还5角硬币。

因果图方法举例

分析这一段说明，列出原因和结果

原因：

1. 售货机有零钱找
2. 投入1元硬币
3. 投入5角硬币
4. 押下橙汁按钮
5. 押下啤酒按钮

结果：

21. 售货机〔零钱找完〕灯亮
22. 退还1元硬币
23. 退还5角硬币
24. 送出橙汁饮料
25. 送出啤酒饮料

画出因果图，如图所示。

所有原因结点列在左边，所有结果结点列在右边。

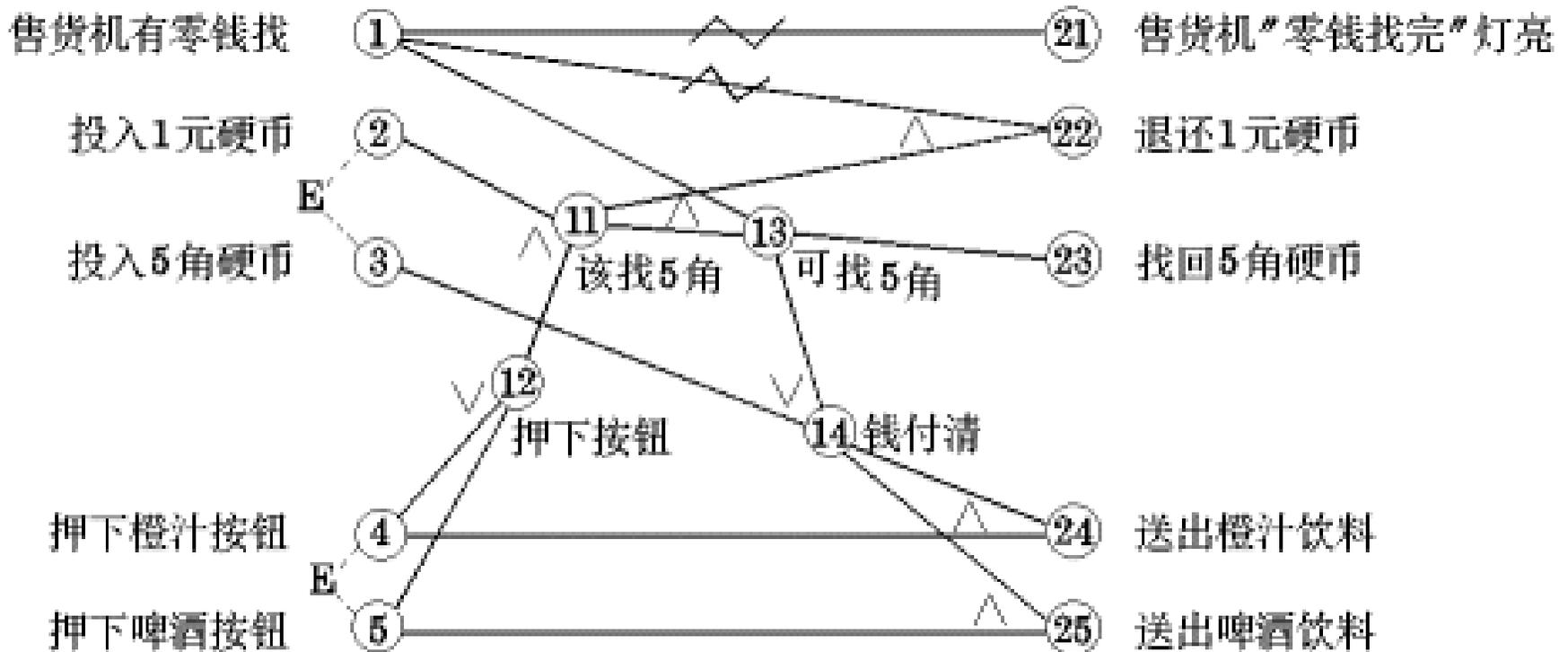
建立中间结点，表示处理的中间状态。

中间结点： 11. 投入1元硬币且押下饮料按钮

12. 押下〔橙汁〕或〔啤酒〕的按钮

13. 应当找5角零钱并且售货机有零钱找

14. 钱已付清



转换成判定表:

序号	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	1	2		
条件	①	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
	②	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	
	③	1	1	1	1	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1	0	0	0	1	1	1	1	0	0	0	0	0	
	④	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0	0	1	1	0	0	1	1	0	0	1	1	0	0
	⑤	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1
中间结果	⑪						1	1	0		0	0	0		0	0	0						1	1	0		0	0	0		0	0	0	
	⑫						1	1	0		1	1	0		1	1	0						1	1	0		1	1	0		1	1	0	
	⑬						1	1	0		0	0	0		0	0	0						0	0	0		0	0	0		0	0	0	
	⑭						1	1	0		1	1	1		0	0	0						0	0	0		1	1	1		0	0	0	
结果	⑳						0	0	0		0	0	0		0	0	0						1	1	1		1	1	1		1	1	1	
	㉑						0	0	0		0	0	0		0	0	0						1	1	0		0	0	0		0	0	0	
	㉒						1	1	0		0	0	0		0	0	0						0	0	0		0	0	0		0	0	0	
	㉓						1	0	0		1	0	0		0	0	0						0	0	0		1	0	0		0	0	0	
	㉔						0	1	0		0	1	0		0	0	0						0	0	0		0	1	0		0	0	0	
测试用例						Y	Y	Y		Y	Y	Y		Y	Y							Y	Y	Y		Y	Y	Y		Y	Y			

因果图方法举例

- 在判定表中，阴影部分表示因违反约束条件的不可能出现的情况，删去。第16列与第32列因什么动作也没做，也删去。最后可根据剩下的16列作为确定测试用例的依据。

因果图方法小结

- 因果图方法是一个非常有效的黑盒测试方法，它能够生成没有重复性的且发现错误能力强的测试用例，而且对输入、输出同时进行了分析。
- 从因果图生成的测试用例（局部,组合关系下的）包括了所有输入数据的取TRUE与取FALSE的情况,构成的测试用例数目达到最少,且测试用例数目随输入数据数目的增加而线性地增加。
- 如果哪个开发项目在设计阶段就采用了判定表，也就不必再画因果图，而是可以直接利用判定表设计测试用例了。

习题

- 1、使用因果图法为三角形问题设计测试用例。
- 2、某软件的一个模块的需求规格说明书中描述：
 - (1) 年薪制员工：严重过失，扣年终风险金的4%；过失，扣年终风险金的2%。
 - (2) 非年薪制员工：严重过失，扣当月薪资的8%；过失，扣当月薪资的4%。

请绘制出因果图和判定表，并给出相应的测试用例。

判定表驱动测试方法

- 前面因果图方法中已经用到了判定表（ Decision Table，也称决策表）。在所有的黑盒测试方法中，基于判定表的测试是最为严格、最具有逻辑性的测试方法。
- 决策表的优点：能够将复杂的问题按照各种可能的情况全部列举出来，简明并避免遗漏。因此，利用决策表能够设计出完整的测试用例集合。
- 在一些数据处理问题当中，某些操作的实施依赖于多个逻辑条件的组合，即：针对不同逻辑条件的组合值，分别执行不同的操作。决策表很适合于处理这类问题。

判定表驱动测试方法

- 决策表的概念：决策表是分析和表达多逻辑条件下执行不同操作的工具。

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了,因为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具体又明确。

判定表驱动测试方法

下表是一张关于科技书阅读指南的判定驱动表：3个问题8种情况

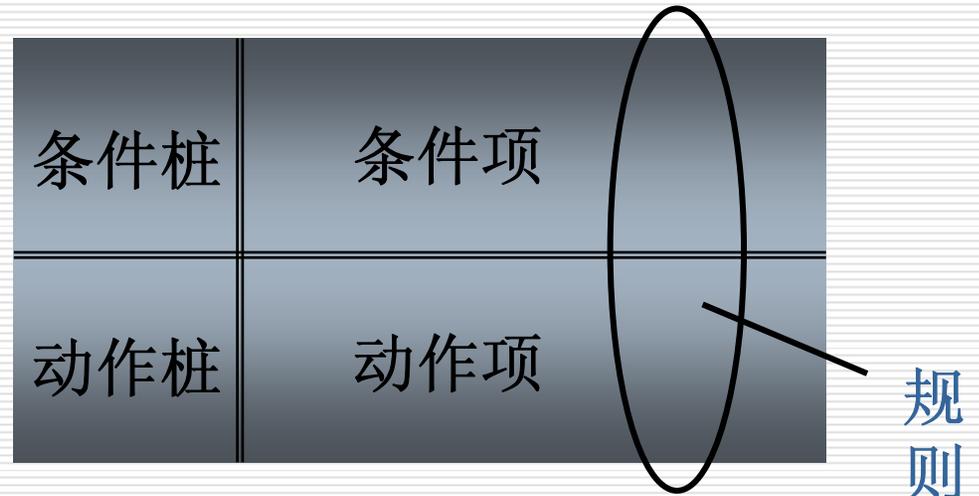
		1	2	3	4	5	6	7	8
问题	你觉得疲倦吗?	Y	Y	Y	Y	N	N	N	N
	你对内容感兴趣吗?	Y	Y	N	N	Y	Y	N	N
	书中内容使你糊涂吗?	Y	N	Y	N	Y	N	Y	N
建议	请回到本章开头重读	x				x			
	继续读下去		x				x		
	跳到下一章去读							x	x
	停止阅读, 请休息			x	x				

”读书指南”判定表

判定表组成

□ 判定表通常由四个部分组成：

- 条件桩
- 动作桩
- 条件项
- 动作项



判定表组成

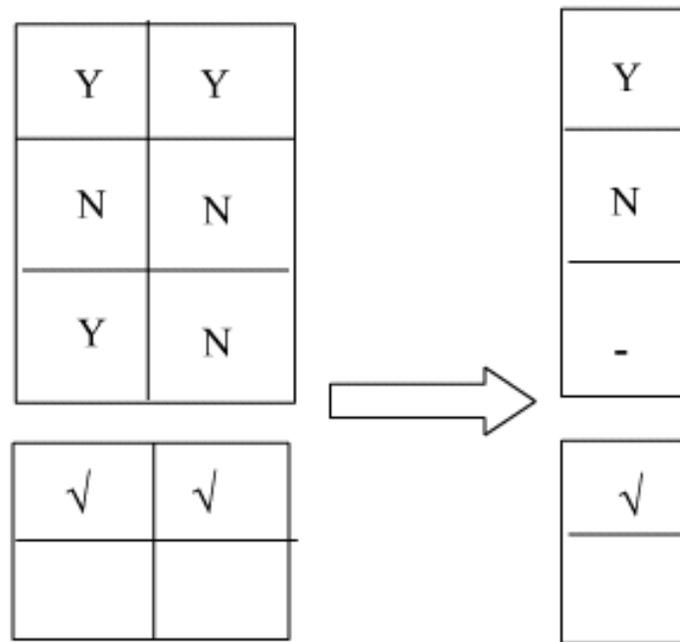
- 判定表通常由四个部分组成：
 - 条件桩（Condition Stub）：列出了问题的所有条件，通常认为列出得条件的次序无关紧要。
 - 动作桩（Action Stub）：列出了问题规定可能采取的操作，这些操作的排列顺序没有约束。
 - 条件项（Condition Entry）：列出针对它左列条件的取值，在所有可能情况下的真假值。
 - 动作项（Action Entry）：列出在条件项的各种取值情况下应该采取的动作。

规则及规则合并

- 规则: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,既条件项和动作项有多少列。
- 化简:就是规则合并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。

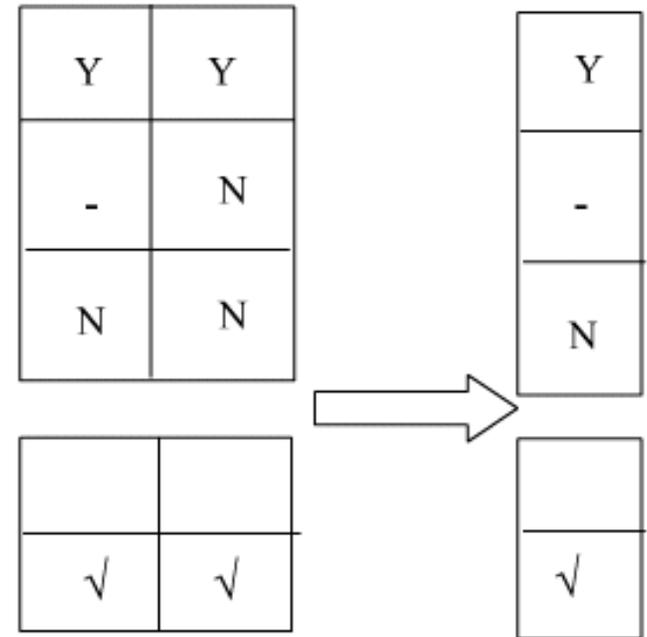
规则及规则合并举例

如右图左端，两规则动作项一样，条件项类似，在1、2条件项分别取Y、N时，无论条件3取何值，都执行同一操作。即要执行的动作与条件3无关。于是可合并。“-”表示与取值无关。



规则及规则合并举例

与上类似，右图中，无关条件项“-”可包含其他条件项取值，具有相同动作的规则可合并



		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	x				x			
	继续读下去		x				x		
	跳到下一章去读							x	x
	停止阅读, 请休息			x	x				

”读书指南”判定表

		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	x			
	继续读下去		X		
	跳到下一章去读				x
	停止阅读, 请休息			x	

化减后的”读书指南”判定表

判定表驱动测试方法

□ 判定表的建立步骤：（根据软件规格说明）

- ①确定规则的个数.假如有 n 个条件。每个条件有两个取值（0,1）,故有 2^n 种规则。
- ②列出所有的条件桩和动作桩。
- ③填入条件项。
- ④填入动作项。得到初始判定表。
- ⑤简化.合并相似规则（相同动作）。

建立判定表举例

例15：问题要求：“.....对功率大于50马力的机器、维修记录不全或已运行10年以上的机器，应给予优先的维修处理.....”。这里假定，“维修记录不全”和“优先维修处理”均已在别处有更严格的定义。请建立判定表。

建立判定表举例

□ 解答：

- ①确定规则的个数：这里有3个条件，每个条件有两个取值，故应有 $2*2*2=8$ 种规则。
- ②列出所有的条件茬和动作茬：

条 件	功率大于 50 马力吗？
	维修记录不全吗？
	运行超过 10 年吗？
动 作	进行优先处理
	作其他处理

建立判定表举例

③填入条件项。

④填入动作桩和动作项。这样便得到形如图的初始判定表。

		1	2	3	4	5	6	7	8
条 件	功率大于 50 马力吗?	Y	Y	Y	Y	N	N	N	N
	维修记录不全吗?	Y	Y	N	N	Y	Y	N	N
	运行超过 10 年吗?	Y	N	Y	N	Y	N	Y	N
动 作	进行优先处理	x	x	X		X		X	
	作其他处理				X		x		x

初始判定表

⑤化简。合并相似规则后得到图。

		1	2	3	4	5	6	7	8
条件	功率大于 50 马力吗?	Y	Y	Y	Y	N	N	N	N
	维修记录不全吗?	Y	Y	N	N	Y	Y	N	N
	运行超过 10 年吗?	Y	N	Y	N	Y	N	Y	N
动作	进行优先处理	x	x	X		X		X	
	作其他处理				X		x		x

初始判定表

		1	2	3	4	5
条件	功率大于 50 马力吗?	Y	Y	Y	N	N
	维修记录不全吗?	Y	N	N	-	-
	运行超过 10 年吗?	-	Y	N	Y	N
动作	进行优先处理	x	x		X	
	作其他处理			x		x

化减后的判定表

三角形问题的决策表

- 例16：以三角形问题给出构造决策表的5个步骤。
- (1) 确定规则个数。例如，三角形问题的决策表有 4 个条件：
 - c1: a、b、c构成三角形？
 - c2: a=b?
 - c3: a=c?
 - c4: b=c?
- 每个条件可以取两个值，故有 $2^4 = 16$ 种规则。
- (2) 列出所有的条件桩和动作桩。
 - (3) 填入输入项。
 - (4) 填入动作项，得到初始决策表。
 - (5) 化简。合并相似规则后得到三角形问题的决策表

三角形问题的决策表

选项 \ 规则	规则 1-8	规则 9	规则 10	规则 11	规则 12	规则 13	规则 14	规则 15	规则 16
条件:									
c1: a,b,c构成三角形?	N	Y	Y	Y	Y	Y	Y	Y	Y
c2: a=b?	-	Y	Y	Y	Y	N	N	N	N
c3: a=c?	-	Y	Y	N	N	Y	Y	N	N
c4: b=c?	-	Y	N	Y	N	Y	N	Y	N
动作:									
a1: 非三角形	√								
a2: 一般三角形									√
a3: 等腰三角形					√		√	√	
a4: 等边三角形		√							
a5: 不可能			√	√		√			

三角形问题的决策表

□ 如将上页c1: a, b, c 构成三角形? 扩展为三个不等式表示

$$C1: a < b + c$$

$$C2: b < a + c$$

$$C3: c < a + b$$

决策表变成如下:

三角形问题的决策表

条件 —— 规则	1	2	3	4	5	6	7	8	9	10	11
C1: $a < b + c$	F	T	T	T	T	T	T	T	T	T	T
C2: $b < a + c$	—	F	T	T	T	T	T	T	T	T	T
C3: $c < a + b$	—	—	F	T	T	T	T	T	T	T	T
C4: $a = b ?$	—	—	—	T	T	T	T	F	F	F	F
C5: $a = c ?$	—	—	—	T	T	F	F	T	T	F	F
C6: $b = c ?$	—	—	—	T	F	T	F	T	F	T	F
规则条数统计	32	16	8	1	1	1	1	1	1	1	1
a1: 非三角形	x	x	x								
a2: 不等边三角形											X
a3: 等腰三角形							x		x	x	
a4: 等边三角形				x							
a5: 不可能					x	x		x			

三角形问题的决策表

用例	a	B	c	预期输出
DT1	4	1	2	非三角形
DT2	1	4	2	非三角形
DT3	1	2	4	非三角形
DT4	5	5	5	等边三角形
DT5	?	?	?	不可能
DT6	?	?	?	不可能
DT7	2	2	3	等腰三角形
DT8	?	?	?	不可能
DT9	2	3	2	等腰三角形
DT10	3	2	2	等腰三角形
DT11	3	4	5	不等边三角形

决策表应用-NextDate函数

□ NextDate函数的决策表测试用例设计

- 问题分析：NextDate 函数中有两种复杂性的输入来源：一是所讨论的输入域的复杂性，二是确定闰年的规则，并要增加“额外天（2月29日）”。
- NextDate函数的三个变量之间在输入定义域中存在一定的逻辑依赖关系，由于等价类划分和边界值分析测试都假设了变量是独立的，如果采用上述两种方法设计测试用例，那么这些依赖关系在机械的选取输入值时可能会丢失。而采用决策表法则可以通过使用“不可能动作”的概念表示条件的不可能组合，来强调这种依赖关系。
- 说明：当决策表规模（指规则的数目， n 个条件的决策表有 2^n 个规则）较大时，可以通过扩展条目决策表（条件使用等价类）、代数简化表、将大表“分解”为小表等方法。

NextDate函数

- 为了获得下一个日期，NextDate函数执行如下操作：
 - 如果输入日期不是当月最后一天，则把day变量的值加1；
 - 如果输入日期是1~11月份中某月的最后一天，则把day变量的值复位为1，month变量的值加1；
 - 如果输入日期是12月的最后一天，则day变量和month变量的值都复位为1，year变量的值加1。
 - 关于最后一天的判断：
 - 如果是有31天的月份(1,3,5,7,8,10,12)，day变量值为31；
 - 如果是有30天的月份(4,6,9,11)，day变量值为30；
 - 如果是有29天的月份(闰年的2月)，day变量值为29；
 - 如果是有28天的月份(非闰年的2月)，day变量值为28。

NextDate函数的动作桩和条件桩

- 根据所执行的操作，可列出NextDate函数的动作桩：
 - a1: 不可能；
 - a2: day加1；
 - a3: day复位；
 - a4: month加1；
 - a5: month复位；
 - a6: year加1

NextDate函数的动作桩和条件桩

- 考虑到决策表的规模，条件使用month、day、year变量的等价类，在以下等价类集合上建立决策表：
 - 对于month变量的取值
 - M1: {month: month有30天};
 - M2: {month: month有31天, 12月除外};
 - M3: {month: month是12月};
 - M4: {month: month是2月};
 - 对于day变量的取值
 - D1: {day: $1 \leq \text{day} \leq 27$ };
 - D2: {day: day=28};
 - D3: {day: day=29};
 - D4: {day: day=30};
 - D5: {day: day=31};
 - 对于year变量的取值
 - Y1: {year: year是闰年};
 - Y2: {year: year不是闰年}

NextDate函数的动作桩和条件桩

条件 — 规则	1	2	3	4	5	6	7	8	9	10
C1:月份在 C2:日期在 C3:年在	M1 D1 —	M1 D2 —	M1 D3 —	M1 D4 —	M1 D5 —	M2 D1 —	M2 D2 —	M2 D3 —	M2 D4 —	M2 D5 —
a1:不可能 a2:日期增1 a3:日期复位 a4:月份增1 a5:月份复位 a6:年增1	X	X	X	X X	X	X	X	X	X	X X

NextDate函数的动作桩和条件桩

条件 — 规则	11	12	13	14	15	16	17	18	19	20	21	22
C1:月份在 C2:日期在 C3:年在	M3 D1	M3 D2	M3 D3	M3 D4	M3 D5 —	M4 D1 —	M4 D2 Y1	M4 D2 Y2	M4 D3 Y1	M4 D3 Y2	M4 D4 —	M4 D5 —
a1:不可能 a2:日期增1 a3:日期复位 a4:月份增1 a5:月份复位 a6:年增1	X	X	X	X	X X X	X	X	X X	X X	X	X	X

输入变量间存在大量逻辑关系的NextData决策表

条件 — 规则	1-3	4	5	6-9	10	11-14	15	16	17	18	19	20	21-22
C1:月份在	M3	M1	M1	M2	M2	M3	M3	M4	M4	M4	M4	M4	M4
C2:日期在	D1 D2 D3	D4	D5	D1 D2 D3 D4	D5	D1 D2 D3 D4	D5	D1	D2	D2	D3	D3	D4 D5
C3:年在	-	-	-	-	—	—	-	-	Y1	Y2	Y1	Y2	—
a1:不可能			X									X	X
a2:日期增1	X			X		X		X	X				
a3:日期复位		X			X		X			X	X		
a4:月份增1		X			X					X	X		
a5:月份复位							X						
a6:年增1							X						

NextDate函数的测试用例

用例	月份	日期	年	预期输出
1 — 3	4	15	2001	2001年4月16日
4	4	30	2001	2001年5月1日
5	4	31	2001	不可能
6 — 9	1	15	2001	2001年1月16日
10	1	31	2001	2001年2月1日
11 — 14	12	15	2001	2001年12月16日
15	12	31	2001	2002年1月1日
16	2	15	2001	2001年2月16日
17	2	28	2004	2004年2月29日
18	2	28	2001	2001年3月1日
19	2	29	2004	2004年3月1日
20	2	29	2001	不可能
21、22	2	30	2001	不可能

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不错的工具。

如果一个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。

判定表在功能测试中的应用

根据规格说明得到如下判定表

	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

根据规则说明得到的判定表

判定表在功能测试中的应用

这里，判定表只给出了16种规则中的8种。事实上，除这8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。在没必要时，判定表通常可略去这些规则。但如果用判定表来设计测试用例，就必须列出这些默许规则（如下表）。

	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	x	x	x	x

默许的规则

判定表在功能测试中的应用

□ 判定表的优点和缺点

■ 优点：

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。

■ 缺点：

不能表达重复执行的动作，例如循环结构。

判定表在功能测试中的应用

- B. Beizer 指出了适合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,或很容易转换成判定表。
 - ②条件的排列顺序不会也不影响执行哪些操作。
 - ③规则的排列顺序不会也不影响执行哪些操作。
 - 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。

习题

- 用决策表测试法测试以下程序：

该程序有三个输入变量month、day、year（month、day和year均为整数值，并且满足： $1 \leq \text{month} \leq 12$ 和 $1 \leq \text{day} \leq 31$ ），分别作为输入日期的月份、日、年份，通过程序可以输出该输入日期在日历上隔一天的日期。

例如，输入为2004年11月29日，则该程序的输出为2000年12月1日。

- (1) 分析各种输入情况，列出为输入变量month、day、year划分的有效等价类。
- (2) 分析程序规格说明，结合以上等价类划分的情况给出问题规定的可能采取的操作（即列出所有的动作桩）。
- (3) 根据（1）和（2），画出简化后的决策表。

习题

- month变量的有效等价类:

M1: {month=4,6,9,11}

M2: {month=1,3,5,7,8,10}

M3: {month=12}

M4: {month=2}

- day变量的有效等价类:

D1: {1 ≤ day ≤ 26}

D2: {day=27}

D3: {day=28}

D4: {day=29}

D5: {day=30}

D6: {day=31}

- year变量的有效等价类:

Y1: {year是闰年}

Y2: {year不是闰年}

- 考虑各种有效的输入情况，程序中可能采取的操作有以下六种:

a1: day+2

a2: day=2

a3: day=1

a4: month+1

a5: month=1

a6: year+1

场景设计方法

□ 场景与事件流

现在的软件几乎都是用事件触发来控制流程的，事件触发时的情景便形成了场景，而同一事件不同的触发顺序和处理结果就形成事件流。

□ 软件测试中的场景与事件流

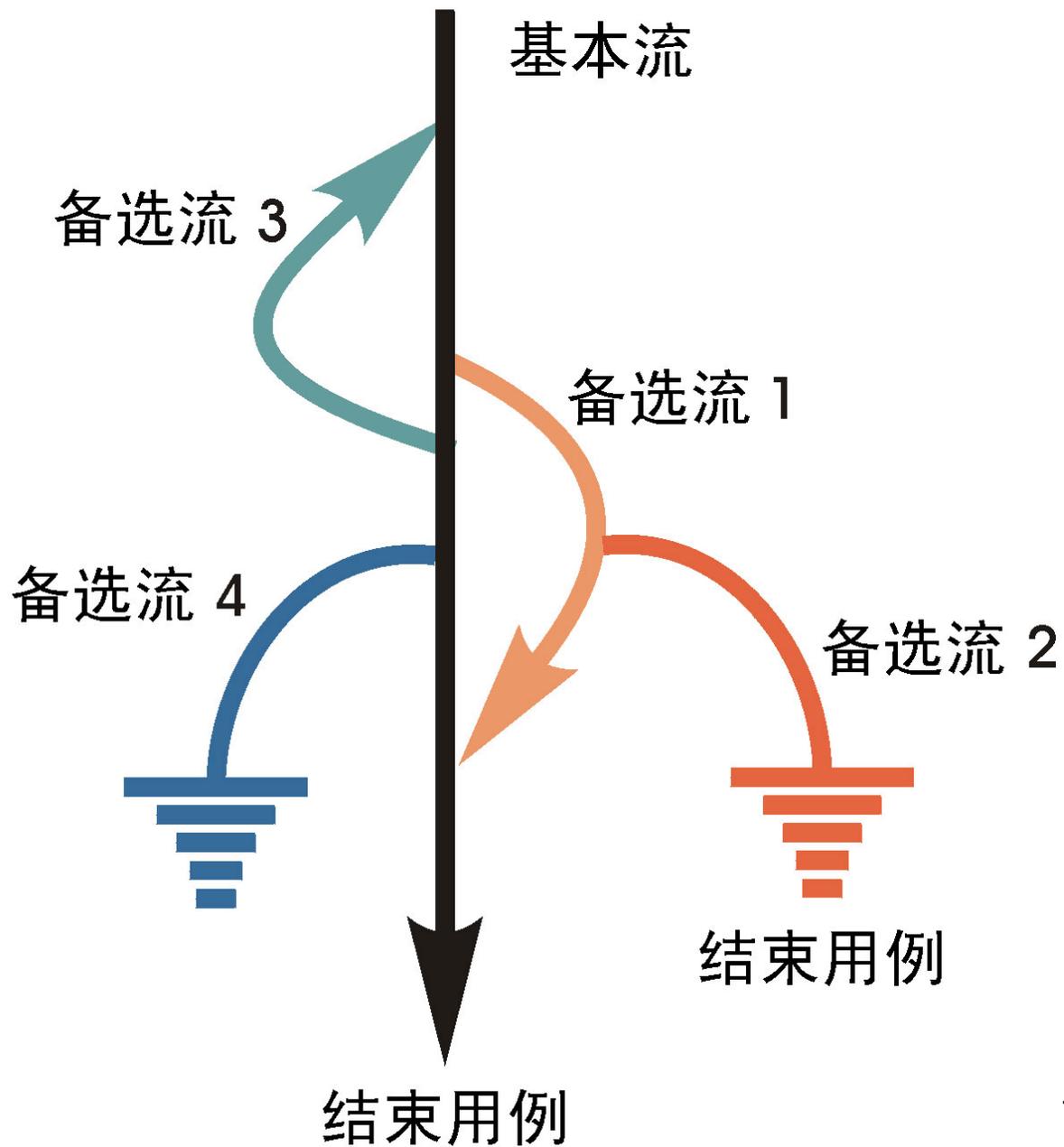
将在软件设计方面的场景与事件流思想引入到软件测试中，可以比较生动地描绘出事件触发时的情景，有利于测试设计者设计测试用例，同时使测试用例更容易理解和执行。

场景设计方法

□ 基本流和备选流

如下页图所示，图中经过用例的每条路径都用基本流和备选流来表示，直黑线表示基本流，是经过用例的最简单的路径。备选流用不同的色彩表示，一个备选流可能从基本流开始，在某个特定条件下执行，然后重新加入基本流中（如备选流1和3）；也可能起源于另一个备选流（如备选流2），或者终止用例而不再重新加入到某个流（如备选流2和4）。

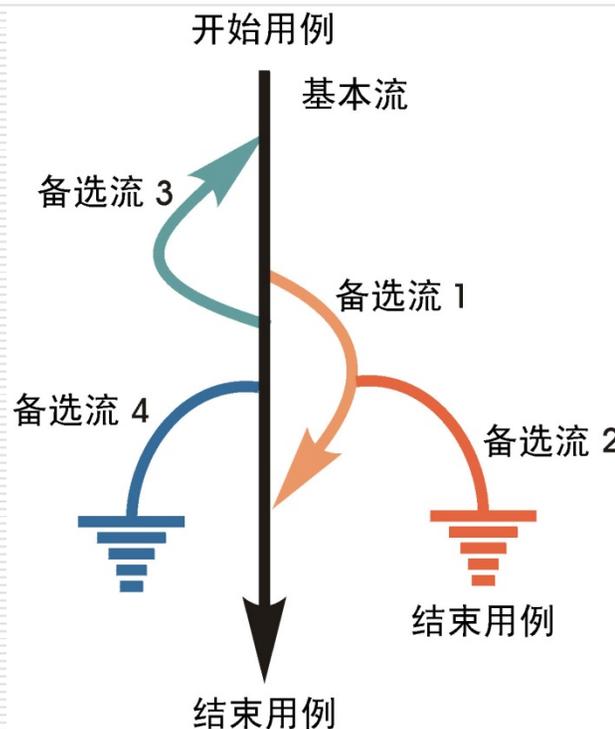
开始用例



场景设计方法

- 按上图所示的每个经过用例的路径，可以确定以下不同的用例场景：（场景5、6、8只考虑了备选流3循环执行一次的情况）

场景1	基本流、
场景2	基本流、备选流1
场景3	基本流、备选流1、备选流2
场景4	基本流、备选流3
场景5	基本流、备选流3、备选流1
场景6	基本流、备选流3、备选流1、备选流2
场景7	基本流、备选流4
场景8	基本流、备选流3、备选流4



场景设计方法

- 使用场景设计方法设计测试用例的步骤：
 - 确定基本流和备选流
 - 根据已确定的基本流和各项备选流生存不同的场景
 - 为确定的场景生成相应的测试用例
 - 复审和验证测试用例，取消多余和等效的。

场景设计方法

□ 例：ATM取款测试的场景设计方法

□ 确定基本流和备选流

基本流	成功提款
备选流2	ATM内没有现金
备选流3	ATM内现金不足
备选流4	PIN有误（还有输入机会）
备选流4	PIN有误（不再有输入机会）
备选流5	账户不存在/账户类型有误
备选流6	账户余额不足

场景设计方法

□ 场景设计：下表所示是客户提款生成的场景。

场景1——成功提款	基本流
场景2——ATM内没有现金	基本流、备选流2
场景3——ATM内现金不足	基本流、备选流3
场景4——PIN有误（还有输入机会）	基本流、备选流4
场景5——PIN有误（不再有输入机会）	基本流、备选流4
场景6——账户不存在/账户类型有误	基本流、备选流5
场景7——账户余额不足	基本流、备选流6

场景设计方法

□ 测试用例设计

对于这7个场景中的每一个场景都需要确定测试用例。如下页显示为前5个场景的测试用例设计，其中各列代表测试用例的信息。

本示例中，对于每个测试用例，存在一个测试用例ID、条件（或说明）、测试用例中涉及的所有数据元素（作为输入或已经存在于数据库中）以及预期结果。

测试用例ID号	场景/条件	PIN	账号	输入或选择的金额	账面金额	ATM内的金额	预期结果
CW1	场景1: 成功提款	V	V	V	V	V	成功提款
CW2	场景2: ATM内没有现金	V	V	V	V	X	提款选项不可用, 用例结束
CW3	场景3: ATM内现金不足	V	V	V	V	X	警告消息, 返回基本流相应步骤, 输入金额
CW4	场景4: PIN有误 (还有不止一次输入机会)	X	V	n/a	V	V	警告消息, 返回基本流相应步骤, 输入 PIN
CW5	场景4: PIN有误 (还有一次输入机会)	X	V	n/a	V	V	警告消息, 返回基本流相应步骤, 输入 PIN
CW6	场景5: PIN有误 (不再有输入机会)	X	V	n/a	V	V	警告消息, 卡予保留, 用例结束

场景设计方法

□ 测试用例生成

一旦确定了所有的测试用例，则应对这些用例进行复审和验证以确保其准确且适度，并取消多余或等效的测试用例。

测试用例一经认可，就可以确定实际数据值，如下页表所示。

测试用例ID号	场景/条件	PIN	账号	输入或选择的金额(元)	账面金额(元)	ATM内的金额(元)	预期结果
CW1	场景1: 成功提款	4987	809-498	50.00	500.00	2000	成功提款。 账户余额被更新为 450.00
CW2	场景2: ATM内没有 现金	4987	809-498	100.00	500.00	0.00	提款选项不可用, 用例结束
CW3	场景3: ATM内现金 不足	4987	809-498	100.00	500.00	70.00	警告消息, 返回基本流相应步 骤, 输入金额
CW4	场景4: PIN有误(还 有不止一次输入 机会)	4978	809-498	n/a	500.00	2000	警告消息, 返回基本流相应步 骤, 输入PIN
CW5	场景4: PIN有误(还 有一次输入机会)	4978	809-498	n/a	500.00	2 000	警告消息, 返回基本流相应步 骤, 输入PIN
CW6	场景5: PIN有误(不 再有输入机会)	4978	809-498	n/a	500.00	2 000	警告消息, 卡予保 留, 用例结束



正交实验设计方法

□ 正交实验设计方法

依据Galois理论(参考数理统计方面的教材),从大量的(实验)数据(测试例)中挑选适量的,有代表性的点(具备“均匀分散,齐整可比”的特点),从而合理地安排测试的一种科学实验设计方法。类似的方法有:聚类分析方法,因子方法方法等。

□ 试验设计方法常用的术语定义如下:

- 因素: 指自变量,用于测试中是指程序的各个输入。
- 水平: 指因素所处的具体状态或情况,又称为等级。用于测试中是指输入的取值个数。

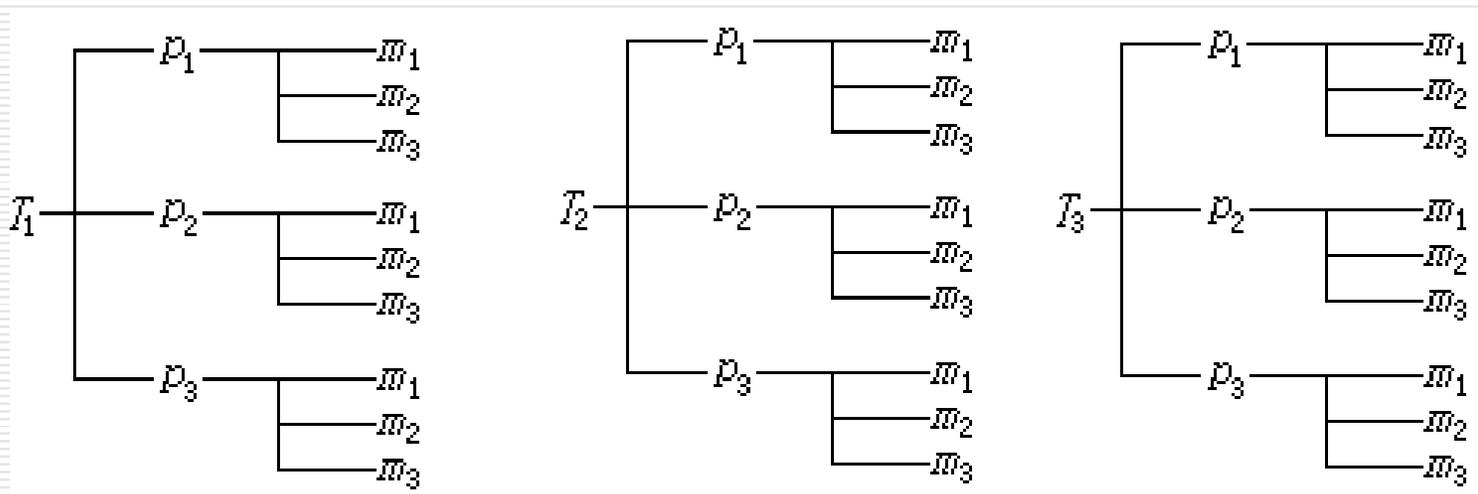
正交实验设计方法

- 例 某化工厂想提高某化工产品的质量和产量，对工艺中三个主要因素各按三个水平进行试验（见表）。试验的目的是为提高合格产品的产量，寻求最适宜的操作条件。

水平	因素	温度℃	压力Pa	加碱量kg
	符号	T	p	M
1		T_1 (80)	p_1 (5.0)	m_1 (2.0)
2		T_2 (100)	p_2 (6.0)	m_2 (2.5)
3		T_3 (120)	p_3 (7.0)	m_3 (3.0)

正交实验设计方法

- 通常有二种方案：全面搭配法和简单对比法
- 全面搭配法：
取三因素所有水平之间的组合，共 $3^3=27$ 次。



优点是能找到最佳

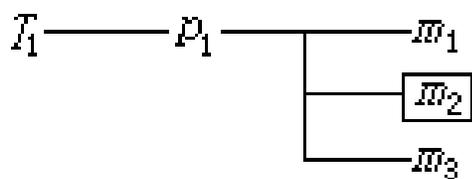
缺点是实验次数太多。如做一个6因素3水平的试验，就需 $3^6=729$ 次实验，显然难以做到。

正交实验设计方法

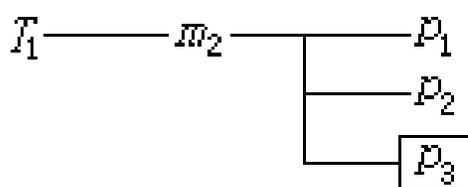
□ 简单对比法

即变化一个因素而固定其他因素，直至所有因素变化为止。

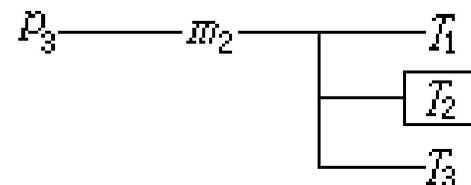
- 1) 先固定 T_1 和 p_1 ，只改变 m ，若发现 $m=m_2$ 时的实验效果最好（用 □ 表示），就认为在后面的实验中因素 m 应取 m_2 水平。



(1)



(2)



(3)

- 2) 固定 T_1 和 m_2 ，改变 p 的三次实验（如图2），若发现 $p=p_3$ 时最好，则固定 p_3 。
- 3) 固定 p_3 和 m_2 ，改变 T 的三次实验（如图3），发现因素 T 宜取 T_2 水平。

于是就认为最好的是 $T_2 p_3 m_2$ 。

正交实验设计方法

与全面搭配法方案相比，简单比较法方案的优点是实验的次数少，只需做9次实验。但简单比较法方案的试验结果是不可靠的。因为，

- ①在改变 m 值（或 p 值，或 T 值）的三次实验中，说 m_2 （或 p_3 或 T_2 ）水平最好是有条件的。在 $T \neq T_1$ ， $p \neq p_1$ 时， m_2 水平不是最好的可能性是有的。
- ②在改变 m 的三次实验中，固定 $T = T_2$ ， $p = p_3$ 是随意的，故在此方案中数据点的分布的均匀性毫无保障。
- ③用这种方法比较条件好坏时，只对单个的试验数据进行数值上的简单比较，不排除必然存在的试验数据误差的干扰。

运用正交试验设计方法，不仅兼有上述两个方案的优点，而且实验次数少，数据点分布均匀，结论的可靠性较好。

正交实验设计方法

□ 正交表的构成

- 正交表的形式：

$$L_{\text{行数}}(\text{水平数}^{\text{因素数}})$$

- 行数(Runs)：正交表中的行的个数，即试验的次数，也是我们通过正交实验法设计的测试用例的个数。
- 因素数(Factors)：正交表中列的个数，即我们要测试的功能点。
- 水平数(Levels)：任何单个因素能够取得的值的最大个数。正交表中的包含的值为从0到数“水平数-1”或从1到“水平数”。即要测试功能点的输入条件。

正交实验设计方法

如：L₈(2⁷)

		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	0	1	0	1
	7	0	0	1	1	0	0	1
	8	0	0	1	0	1	1	0

Diagram annotations: A box labeled "因数" (Factor) has arrows pointing to columns 1 and 2. A box labeled "水平值" (Level Value) has arrows pointing to columns 5 and 6.

正交实验设计方法

□ 正交表的正交性

■ 整齐可比性

每个因素（列）的每个水平值出现的次数是完全相同的。

由于在试验中每个因素的每个水平与其它因素的每个水平参与试验的机率是完全相同的，这就保证在各个水平中最大程度的排除了其它因素水平的干扰。因而，能最有效地进行比较和作出展望，容易找到好的试验条件。

■ 均衡分散性

任意两列（两个因素）的水平搭配（横向形成的数字对）是完全相同的。

这样就保证了试验条件均衡地分散在因素水平的完全组合之中，因而具有很强的代表性，容易得到好的试验条件。

正交实验设计方法

如，在 $L_9(3^4)$ 中：（见下页）

- 1) 每一列中，各不同的数字出现的次数相同。每一列有三个水平，水平1、2、3都是各出现3次。
- 2) 任意两列并列在一起形成的数字对共有9个： $(1,1)$ ， $(1,2)$ ， $(1,3)$ ， $(2,1)$ ， $(2,2)$ ， $(2,3)$ ， $(3,1)$ ， $(3,2)$ ， $(3,3)$ ，每一个数字对各出现一次。

□ 常用的各列水平数均相同的正交表

各列水平均为2的常用正交表有： $L_4(2^3)$ ， $L_8(2^7)$ ， $L_{12}(2^{11})$ ， $L_{16}(2^{15})$ ， $L_{20}(2^{19})$ ， $L_{32}(2^{31})$ 。

各列水平数均为3的常用正交表有： $L_9(3^4)$ ， $L_{27}(3^{13})$ 。

各列水平数均为4的常用正交表有： $L_{16}(4^5)$

各列水平数均为5的常用正交表有： $L_{25}(5^6)$

正交实验设计方法

试验号	列号	1	2	3	4
	因素	温度 $^{\circ}\text{C}$	压力Pa	加碱量 kg	
	符号	T	p	m	
1		1 (T_1)	1 (p_1)	1 (m_1)	1
2		1 (T_1)	2 (p_2)	2 (m_2)	2
3		1 (T_1)	3 (p_3)	3 (m_3)	3
4		2 (T_2)	1 (p_1)	2 (m_2)	3
5		2 (T_2)	2 (p_2)	3 (m_3)	1
6		2 (T_2)	3 (p_3)	1 (m_1)	2
7		3 (T_3)	1 (p_1)	3 (m_3)	2
8		3 (T_3)	2 (p_2)	1 (m_1)	3
9		3 (T_3)	3 (p_3)	2 (m_2)	1

正交实验设计方法

□ 实验次数（行数）

实验次数（行数）= Σ (每列水平数-1) + 1

如4个3水平因素，其实验次数为： $4*(3-1)+1=9$

利用上述关系式可从所要考察的因素水平数来决定最低的实验次数，进而选择合适的正交表。

□ 设计实验方案

只要把所考察的每个因素任意地对应于正交表的一列，然后把每列数字“翻译”成所对应的因素水平即可。这样，每一行就是一个实验条件。

如前例，因素T，P，M都是3水平的，实验次数不少于

$3*(3-1)+1=7$ 次，可考虑用 $L_9(3^4)$ 。因素T，P，M任意对应某3列，然后按行实验。

正交实验设计方法

- 利用正交实验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) 把变量的值映射到表中
 - 5) 把每一行的各因素水平的组合做为一个测试用例
 - 6) 加上你认为可疑且没有在表中出现的组合

正交实验设计方法

□ 如何选择正交表

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

□ 设计测试用例时的二种情况

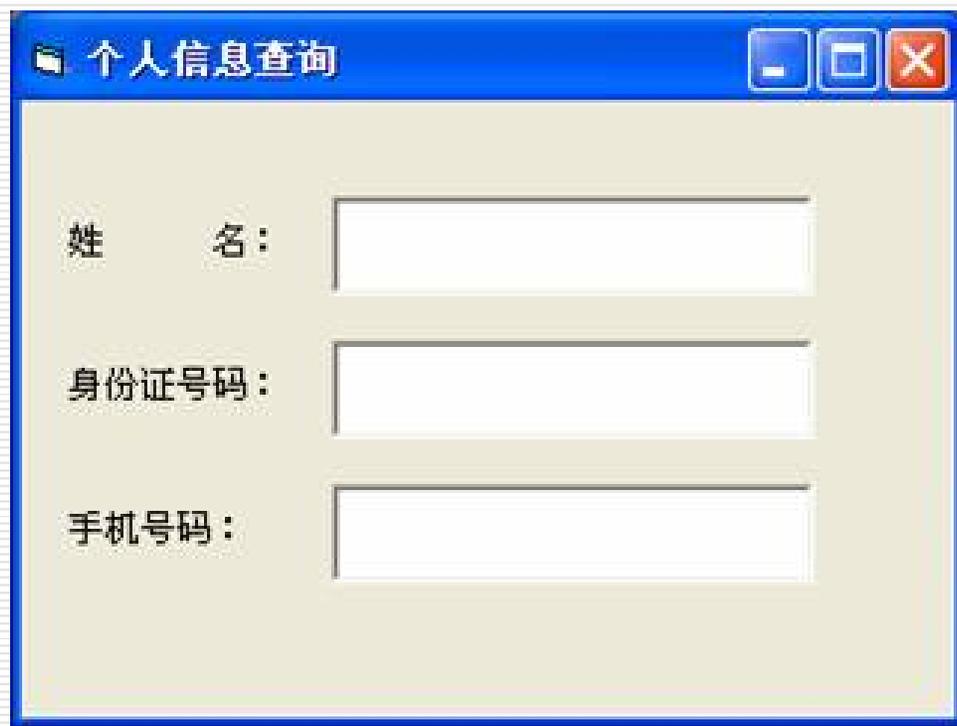
- 1) 水平数（变量值）相同
- 2) 水平数不相同

正交实验设计方法

□ 水平数相同举例

如图窗口，要测试的控件有3个：姓名、身份证号码、手机号码，也就是要考虑的因素有三个；

而每个因素里的状态有两个：填与不填。



The image shows a screenshot of a software window titled "个人信息查询" (Personal Information Query). The window has a blue title bar with standard Windows window controls (minimize, maximize, close). The main content area is light beige and contains three input fields stacked vertically. Each field is preceded by a label: "姓名:" (Name), "身份证号码:" (ID Number), and "手机号码:" (Mobile Number). The input fields are empty text boxes.

正交实验设计方法

选择正交表：

- 1、表中的因素数=3；
- 2、表中至少有3个因素数的水平数=2；
- 3、行数取最少的一个。

从正交表公式中开始查找，结果为： $L_4(2^3)$

变量映射：

		列号					列号		
行号		1	2	3	行号	姓名	身份证号	手机号码	
	1	0	0	0		1	填	填	填
	2	0	1	1		2	填	不填	不填
	3	1	0	1		3	不填	填	不填
	4	1	1	0		4	不填	不填	填
0--填		1--不填							

正交实验设计方法

□ 测试用例如下：

1：填写姓名、填写身份证号、填写手机号

2：填写姓名、不填身份证号、不填手机号

3：不填姓名、填写身份证号、不填手机号

4：不填姓名、不填身份证号、填写手机号

增补测试用例

5：不填姓名、不填身份证号、不填手机号

从测试用例可以看出：如果按每个因素两个水平数来考虑的话，需要8个测试用例，而通过正交实验法进行的测试用例只有5个，大大减少了测试用例数。用最小的测试用例集合去获取最大的测试覆盖率。

正交实验设计方法

□ 水平数不相同举例

假设PowerPoint软件打印功能描述如下：

打印范围分：全部、当前幻灯片、给定范围 共三种情况；

打印内容分：幻灯片、讲义、备注页、大纲视图 共四种方式

；

打印颜色/灰度分：颜色、灰度、黑白 共三种设置；

打印效果分：幻灯片加框和幻灯片不加框两种方式。

正交实验设计方法

□ 因素状态表

状态/因素	A打印范围	B打印内容	C打印颜色/灰度	D打印效果
0	全部	幻灯片	颜色	幻灯片加框
1	当前幻灯片	讲义	灰度	幻灯片不加框
2	给定范围	备注页	黑白	
3		大纲视图		

正交实验设计方法

- 将中文字转换成字母，便于设计。得到：因素状态表：

状态/因素	A	B	C	D
0	A1	B1	C1	D1
1	A2	B2	C2	D2
2	A3	B3	C3	
3		B4		

正交实验设计方法

□ 选择正交表：

1、表中的因素数=4

2、表中至少有4个因素的水平数 ≥ 2

3、行数取最少的一个

$$2^*(3-1) + (4-1) + (2-1) + 1 = 9$$

最后选中正交表公式： $L_{16}(4^5)$

正交矩阵为及用字母替代的正交矩阵为：

	1	2	3	4	5
1	0	0	0	0	0
2	0	1	1	1	1
3	0	2	2	2	2
4	0	3	3	3	3
5	1	0	1	2	3
6	1	1	0	3	2
7	1	2	3	0	1
8	1	3	2	1	0
9	2	0	2	3	1
10	2	1	3	2	0
11	2	2	0	1	3
12	2	3	1	0	2
13	3	0	3	1	2
14	3	1	2	0	3
15	3	2	1	3	0
16	3	3	0	2	1

	1	2	3	4	5
1	A1	B1	C1	D1	0
2	A1	B2	C2	D2	1
3	A1	B3	C3	2	2
4	A1	B4	3	3	3
5	A2	B1	C2	2	3
6	A2	B2	C1	3	2
7	A2	B3	3	D1	1
8	A2	B4	C3	D2	0
9	A3	B1	C3	3	1
10	A3	B2	3	2	0
11	A3	B3	C1	D2	3
12	A3	B4	C2	D1	2
13	3	B1	3	D2	2
14	3	B2	C3	D1	3
15	3	B3	C2	3	0
16	3	B4	C1	2	1

正交实验设计方法

第一列水平值为3、第三列水平值为3、第四列水平值2，未填的都需要由各自的字母替代。

	1	2	3	4	5
1	A1	B1	C1	D1	0
2	A1	B2	C2	D2	1
3	A1	B3	C3	D1	2
4	A1	B4	C1	D2	3
5	A2	B1	C2	D1	3
6	A2	B2	C1	D2	2
7	A2	B3	C2	D1	1
8	A2	B4	C3	D2	0
9	A3	B1	C3	D2	1
10	A3	B2	C3	D1	0
11	A3	B3	C1	D2	3
12	A3	B4	C2	D1	2
13	A1	B1	C1	D2	2
14	A2	B2	C3	D1	3
15	A3	B3	C2	D2	0
16	A1	B4	C1	D1	1

正交实验设计方法

第五列去掉没有意义。通过分析，由于四个因素里有三个的水平值小于3，所以从第13行到16行的测试用例可以忽略。

测试用例编号	PPT—ST—FUNCTION—PRINT—001
测试项目	测试powerpoint打印功能
测试标题	打印PowerPoint文件A全部的幻灯片，有颜色，加框
重要级别	高
预置条件	PowerPoint文件A已被打开，电脑主机已连接有效打印机
输入	文件A： D:\系统测试.ppt
操作步骤	<ol style="list-style-type: none">1、打开打印界面；2、打印范围选择“全部”；3、打印内容选择“幻灯片”；4、颜色/灰度选择“颜色”；5、在“幻灯片加框”前打勾；6、点击“确定”。
预期输出	打印出全部幻灯片，有颜色且已加框。

测试用例编号	PPT—ST— FUNCTION—PRINT—002
测试项目	测试powerpoint打印功能
测试标题	打印PowerPoint文件A全部的幻灯片为讲义，灰度，不加框
重要级别	中
预置条件	PowerPoint文件A已被打开，电脑主机已连接有效打印机
输入	文件A： D:\系统测试.ppt
操作步骤	<ol style="list-style-type: none">1、打开打印界面；2、打印范围选择“全部”；3、打印内容选择“讲义”；4、颜色/灰度选择“灰度”；5、点击“确定”。
预期输出	打印出全部幻灯片为讲义，灰度且不加框。

测试用例编号	PPT—ST—FUNCTION—PRINT—003
测试项目	测试powerpoint打印功能
测试标题	打印PowerPoint文件A全部的备注页，黑白，加框
重要级别	中
预置条件	PowerPoint文件A已被打开，电脑主机已连接有效打印机
输入	文件A： D:\系统测试.ppt
操作步骤	<ol style="list-style-type: none">1、打开打印界面；2、打印范围选择“全部”；3、打印内容选择“备注页”；4、颜色/灰度选择“黑白”；5、在“幻灯片加框”前打勾；6、点击“确定”。
预期输出	打印出全部备注页，黑白且已加框。

测试用例编号	PPT—ST—FUNCTION—PRINT—004
测试项目	测试powerpoint打印功能
测试标题	打印PowerPoint文件A全部的大纲视图，黑白
重要级别	中
预置条件	PowerPoint文件A已被打开，电脑主机已连接有效打印机
输入	文件A： D:\系统测试.ppt
操作步骤	<ol style="list-style-type: none">1、打开打印界面；2、打印范围选择“全部”；3、打印内容选择“大纲视图”；4、颜色/灰度选择“黑白”；5、点击“确定”。
预期输出	打印出全部大纲视图，黑白

测试用例编号	PPT—ST—FUNCTION—PRINT—005
测试项目	测试powerpoint打印功能
测试标题	打印PowerPoint文件A当前幻灯片，灰度，加框
重要级别	中
预置条件	PowerPoint文件A已被打开，电脑主机已连接有效打印机
输入	文件A： D:\系统测试.ppt
操作步骤	<ol style="list-style-type: none">1、打开打印界面；2、打印范围选择“当前幻灯片”；3、打印内容选择“幻灯片”；4、颜色/灰度选择“灰度”；5、在“幻灯片加框”前打勾；6、点击“确定”。
预期输出	打印出当前幻灯片，灰度且已加框。

测试用例编号	PPT—ST—FUNCTION—PRINT—006
测试项目	测试powerpoint打印功能
测试标题	打印PowerPoint文件A当前幻灯片为讲义，黑白，加框
重要级别	中
预置条件	PowerPoint文件A已被打开，电脑主机已连接有效打印机
输入	文件A： D:\系统测试.ppt
操作步骤	<ol style="list-style-type: none">1、打开打印界面；2、打印范围选择“当前幻灯片”；3、打印内容选择“讲义”；4、颜色/灰度选择“黑白”；5、在“幻灯片加框”前打勾；6、点击“确定”。
预期输出	打印出当前幻灯片为讲义，黑白且已加框。

测试用例编号	PPT—ST—FUNCTION—PRINT—007
测试项目	测试powerpoint打印功能
测试标题	打印PowerPoint文件A当前幻灯片的备注页，有颜色，不加框
重要级别	中
预置条件	PowerPoint文件A已被打开，电脑主机已连接有效打印机
输入	文件A： D:\系统测试.ppt
操作步骤	<ol style="list-style-type: none">1、打开打印界面；2、打印范围选择“当前幻灯片”；3、打印内容选择“备注页”；4、颜色/灰度选择“颜色”；5、点击“确定”。
预期输出	打印出当前幻灯片的备注页，有颜色且不加框。

测试用例编号	PPT—ST—FUNCTION—PRINT—008
测试项目	测试powerpoint打印功能
测试标题	打印PowerPoint文件A当前幻灯片的大纲视图，有颜色
重要级别	中
预置条件	PowerPoint文件A已被打开，电脑主机已连接有效打印机
输入	文件A： D:\系统测试.ppt
操作步骤	<ol style="list-style-type: none">1、打开打印界面；2、打印范围选择“当前幻灯片”；3、打印内容选择“大纲视图”；4、颜色/灰度选择“颜色”；5、点击“确定”。
预期输出	打印出当前幻灯片为讲义，黑白且已加框。

测试用例编号	PPT—ST—FUNCTION—PRINT—009
测试项目	测试powerpoint打印功能
测试标题	打印PowerPoint文件A给定范围的幻灯片，黑白，不加框
重要级别	中
预置条件	PowerPoint文件A已被打开，电脑主机已连接有效打印机
输入	文件A： D:\系统测试.ppt
操作步骤	<ol style="list-style-type: none">1、打开打印界面；2、打印范围选择“幻灯片”；3、打印内容选择“幻灯片”；4、颜色/灰度选择“黑白”；5、点击“确定”。
预期输出	打印出给定范围的幻灯片，黑白且不加框。

测试用例编号	PPT—ST—FUNCTION—PRINT—010
测试项目	测试powerpoint打印功能
测试标题	打印PowerPoint文件A给定范围的幻灯片为讲义，有颜色，加框
重要级别	中
预置条件	PowerPoint文件A已被打开，电脑主机已连接有效打印机
输入	文件A： D:\系统测试.ppt
操作步骤	<ol style="list-style-type: none">1、打开打印界面；2、打印范围选择“幻灯片”；3、打印内容选择“幻灯片”；4、颜色/灰度选择“颜色”；5、点击“确定”。
预期输出	打印出给定范围的幻灯片为讲义，有颜色且加框。

测试用例编号	PPT—ST—FUNCTION—PRINT—011
测试项目	测试powerpoint打印功能
测试标题	打印PowerPoint文件A给定范围的幻灯片的备注页，灰度，加框
重要级别	中
预置条件	PowerPoint文件A已被打开，电脑主机已连接有效打印机
输入	文件A： D:\系统测试.ppt
操作步骤	<ol style="list-style-type: none">1、打开打印界面；2、打印范围选择“幻灯片”；3、打印内容选择“备注页”；4、颜色/灰度选择“灰度”；5、在“幻灯片加框”前打勾；6、点击“确定”。
预期输出	打印出给定范围的幻灯片的备注页，灰度且加框。

测试用例编号	PPT—ST—FUNCTION—PRINT—012
测试项目	测试powerpoint打印功能
测试标题	打印PowerPoint文件A给定范围的幻灯片的大纲视图，灰度
重要级别	中
预置条件	PowerPoint文件A已被打开，电脑主机已连接有效打印机
输入	文件A： D:\系统测试.ppt
操作步骤	<ol style="list-style-type: none">1、打开打印界面；2、打印范围选择“幻灯片”；3、打印内容选择“大纲视图”；4、颜色/灰度选择“灰度”；5、点击“确定”。
预期输出	打印出给定范围的幻灯片的大纲视图，灰度

黑盒测试的其他几种方法简介

□ 功能图分析方法

用功能图形式化地表示程序的功能说明,并机械地生成功能图的测试用例

功能性测试小结

程序看成将输入映射到输出的数学函数

局限：未测试的功能漏洞、冗余测试

□ 边界值分析：以输入变量的边界，构造测试用例

- 边界值分析测试
- 边界值健壮性测试
- 边界值最坏情况测试
- 边界值健壮最坏情况测试

功能性测试小结

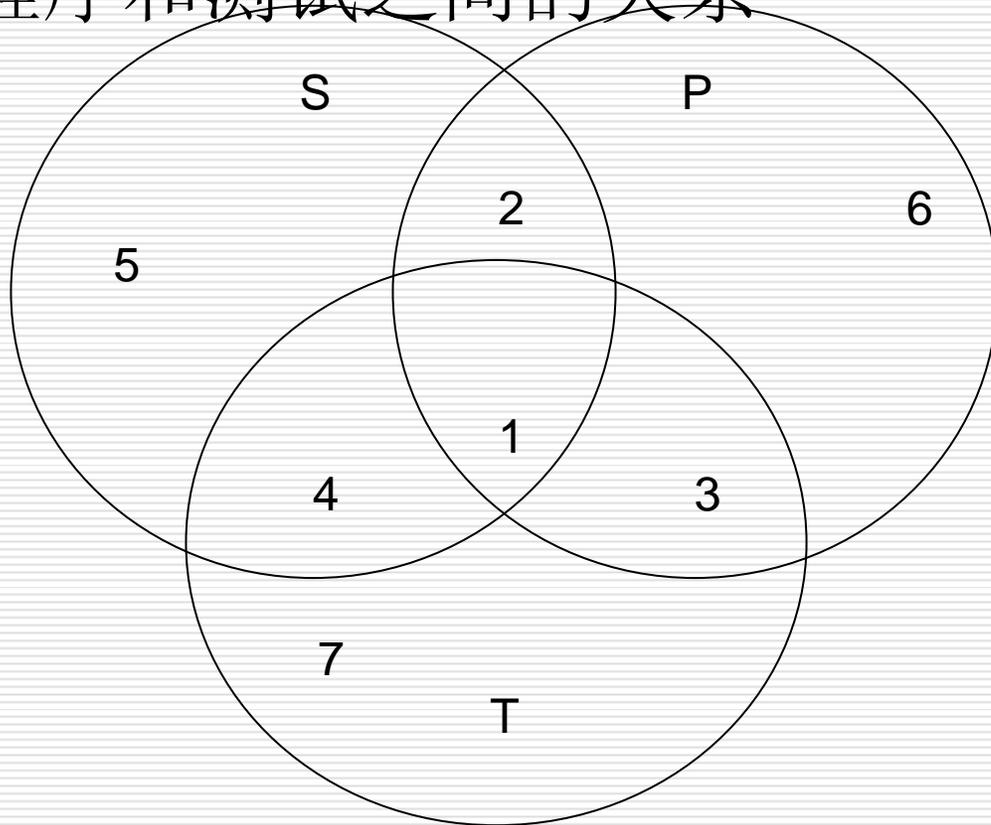
□ 等价类测试

- 弱一般等价类测试，单缺陷假设
- 强一般等价类测试，多缺陷假设
- 弱健壮等价类测试，考虑无效值
- 强健壮等价类测试，既考虑无效值，又考虑多缺陷，输出值域等价类

□ 因果图，基于决策表的测试

功能性测试小结

SRS, 程序和测试之间的关系



功能性测试小结

S: 已描述的行为（规格说明SRS）

P: 已实现的行为（程序实现）

T: 已经过测试的行为

2, 5: 已描述行为，但没有测试

1, 4: 经过测试的已描述行为

3, 7: 未描述行为的测试用例

5, 4: 已描述行为，但没有实现

3, 6: 实现了未描述的行为

Myers测试方法选择的综合策略

Myers提出了使用各种测试方法的综合策略：

- 在任何情况下都必须使用边界值分析方法。经验表明用这种方法设计出测试用例发现程序错误的能力最强。
- 必要时用等价类划分方法补充一些测试用例。
- 用错误推测法再追加一些测试用例。
- 对照程序逻辑，检查已设计出的测试用例的逻辑覆盖程度。如果没有达到要求的覆盖标准，应当再补充足够的测试用例。
- 如果程序的功能说明中含有输入条件的组合情况，则一开始就可选用因果图法。

功能性测试小结：

- 1、如果变量引用的是物理量，可采用定义域测试和等价类测试
- 2、如果变量是独立的，可采用定义域测试和等价类测试
- 3、如果变量不是独立的，可采用决策表测试
- 4、如果可保证是单缺陷假设，可采用边界值分析和健壮性测试
- 5、如果可保证是多缺陷假设，可采用最坏情况测试，健壮最坏情况测试和决策表测试
- 6、如果程序包含大量例外处理，可采用健壮性测试和决策表测试
- 7、如果变量引用的是逻辑量，可采用等价类测试和决策表测试