

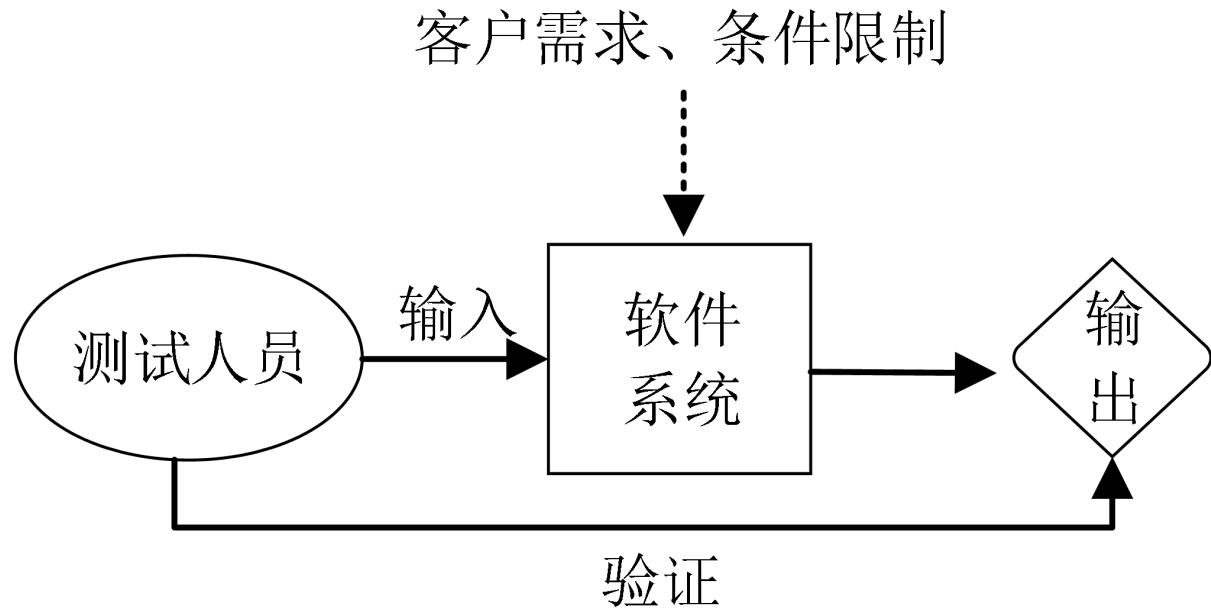
第2章 黑盒测试方法





黑盒测试是软件测试中经常使用的一种测试方法，常用的**黑盒测试方法**包括**等价类划分法、边界值分析法、因果图法与决策表法、正交实验设计法、场景法、状态迁移图法**等，这些方法非常实用，本章将对黑盒测试的常用方法进行详细讲解。

黑盒测试又称**功能测试、数据驱动测试或基于规格说明的测试**,主要是**根据功能需求来测试程序是否按照预期工作**。黑盒测试把程序看作一个**不能打开的黑盒子**,不考虑内部逻辑结构。从用户的角度验证软件功能,实现端到端(end-to-end)的测试。



黑盒测试方法对被测程序主要进行如下三个方面的检查：

- (1) 检查程序能否按需求规格说明书规定正常使用，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、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是否有错。

- (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，**从产品功能角度测试可以最大限度地满足用户的需求**；
- (4) 基于软件开发文档，将测试直接和待测软件要完成的操作相关联，所以能够知道**软件实现了文档中的哪些功能**；
- (5) 依据测试用例有针对性地寻找问题，**定位更为准确，容易生成测试数据**；

- (1) 代码得不到测试，测试覆盖有盲区；
- (2) **如果需求规格说明设计有误，很难发现错误所在；**
- (3) 测试用例设计难度大；
- (4) 存在一定的冗余性。

黑盒测试是穷举输入测试，只有把所有可能的输入都作为测试数据使用，才能查出程序中所有的错误。实际上测试情况有无穷多个，进行测试时不仅要测试所有合法的输入，而且还要对那些不合法的、但是可能的输入进行测试。常用的黑盒测试方法有等价类划分法、边界值法、因果图法、决策表法、正交测试法、错误推测法和场景法等，需要应针对软件开发项目的具体特点，选择合适的测试方法。

2.1

等价类划分法



» 2.1.1 等价类划分法概述



掌握等价类划分法概述，能够灵活[应用等价类划
分法设计测试用例](#)



一个程序可以有多个输入数据，等价类划分法的原理是把所有可能的输入数据，即程序的输入域划分成若干互不相交的子集，称为等价类。所有子集的并集则构成整个输入域。然后从每一个子集中选取少量具有代表性的数据作为测试用例。等价类划分类似于师生站队，男生站左边，女生站右边，老师站中间，这样就把师生群体划分成了3个等价类。

完备性

整个输入域提供测试的完备性

等价类互不相交保证测试数据
的无冗余性

无冗余性

使用等价类划分法测试程序需要经过[划分等价类](#)和[设计测试用例](#)2个步骤。

1. 划分等价类

等价类可以分为[有效等价类](#)和[无效等价类](#)，其含义如下。

有效等价类

是有效值的集合，这些有效值是符合程序要求、合理且有意义的输入数据。

无效等价类

是无效值的集合，这些无效值是不符合程序要求、不合理或无意义的输入数据。

此处应注意，正确的术语应该是“无效值的等价类”，而不是“无效等价类”，因为等价类本身并不是无效的，只是这个等价类对于某个特定的输入值是无效的。

通常在划分等价类时，需要遵守以下原则。

| 原则一 | 在输入条件规定了取值范围的情况下，可以确立一个有效等价类和两个无效等价类。 |
|-----|--|
| 示例 | <p>输入条件：输入值是学生成绩，范围是0-100，则学生成绩的等价类划分如图所示。</p>  <p>有效等价类： $0 \leq \text{成绩} \leq 100$;</p> <p>无效等价类： 成绩<0; 成绩>100。</p> |

| | |
|-----|--|
| 原则二 | <p>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。</p> |
| 示例 | <p>输入条件：输入值必须是一个11位的数字字符串。</p> <p>有效等价类：所有完全符合“11位数字”这一条件的输入； 无效等价类：所有不符合“11位数字”这一条件的输入的集合。</p> |

| | |
|-----|---|
| 原则三 | <p>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。</p> |
| 示例 | <p>输入条件：一个布尔量。用户必须勾选“我已阅读并同意用户协议”复选框才能完成注册。</p> <p>有效等价类：True (复选框被勾选)； 无效等价类：False (复选框未被勾选)。</p> |

| | |
|-----|---|
| 原则四 | <p>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。</p> |
| 示例 | <p>输入条件: 学生的学位。可为学士、硕士、博士三种之一</p> <p>有效等价类: 学士；硕士；博士。</p> <p>无效等价类: 三种学位之外的任何学位。</p> |

| | |
|-----|--|
| 原则五 | <p>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）。</p> |
| 示例 | <p>输入条件：必须同时遵守以下规则：</p> <ul style="list-style-type: none">(1) 长度在8-16位之间。(2) 必须包含至少一个大写字母。(3) 必须包含至少一个小写字母。(4) 必须包含至少一个数字。 <p>有效等价类：同时满足所有4条规则的数据。</p> <p>无效等价类：</p> <ul style="list-style-type: none">(1) 违反长度规则，密码太长；(2) 违反长度规则，密码太短；(3) 违反字母大小写规则，缺少大写字母；(4) 违反字母大小写规则，缺少小写字母；(5) 违反数字规则，缺少数字；(6) 违反多项规则。 <p>.....</p> |

2. 设计测试用例

当确定等价类后，需要建立等价类表，列出所有划分出的等价类，用以设计测试用例。

基于等价类划分法的测试用例设计步骤如下。

1

确定测试对象，保证非测试对象的正确性。

2

为每个等价类规定一个唯一编号。



3

设计有效等价类的测试用例，使其尽可能多地覆盖尚未被覆盖的有效等价类，直到测试用例覆盖所有的有效等价类。

4

设计无效等价类的测试用例，使其覆盖所有的无效等价类。



2.1.2 实例一：QQ账号合法性的等价类划分



掌握QQ账号合法性的等价类划分，能够建立QQ账号的等价类划分表与设计QQ账号的测试用例



2.1.2 实例一：QQ账号合法性的等价类划分



QQ是一款基于互联网的即时通信软件，假设QQ账号的要求是6~10位自然数，在登录QQ时，可以根据QQ账号的长度判断QQ账号的合法性。

下面以QQ账号为测试对象，在判断QQ账号的合法性时，需要明确2个条件：第1个条件是QQ账号的长度为6~10；第2个条件是QQ账号的数据类型为自然数。根据这2个条件可以将QQ账号划分为1个有效等价类和5个无效等价类，具体如下。

- 有效等价类：6~10位自然数。
- 无效等价类：少于6位自然数（包含空值）。
- 无效等价类：多于10位自然数。
- 无效等价类：6~10位非自然数。
- 无效等价类：少于6位非自然数（包含空值）。
- 无效等价类：多于10位非自然数。



2.1.2 实例一：QQ账号合法性的等价类划分



通过前面的分析，下面将6个等价类进行编号，建立等价类划分表。QQ账号的等价类划分表如下表所示。

| 要求 | 有效等价类 | 有效等价类编号 | 无效等价类 | 无效等价类编号 |
|---------------|----------|---------|----------------|---------|
| QQ账号为6~10位自然数 | 6~10位自然数 | 1 | 少于6位自然数（包含空值） | 2 |
| | | | 多于10位自然数 | 3 |
| | | | 6~10位非自然数 | 4 |
| | | | 少于6位非自然数（包含空值） | 5 |
| | | | 多于10位非自然数 | 6 |
| | | | | |



2.1.2 实例一：QQ账号合法性的等价类划分



基于等价类划分法设计QQ账号的测试用例如下表所示。

| 测试用例编号 | 测试数据 | 覆盖等价类编号 | 预期结果 |
|--------|--------------|---------|---------|
| test1 | 123456789 | 1 | QQ账号合法 |
| test2 | 12345 | 2 | QQ账号不合法 |
| test3 | 123456789012 | 3 | QQ账号不合法 |
| test4 | 1234@& | 4 | QQ账号不合法 |
| test5 | null | 5 | QQ账号不合法 |
| test6 | @#&88888888 | 6 | QQ账号不合法 |



2.1.3 实例二：三角形问题的等价类划分



掌握三角形问题的等价类划分，能够建立三角形输入等价类表、设计覆盖有效与无效等价类的测试用例



2.1.3 实例二：三角形问题的等价类划分



案例描述

三角形问题是测试中广泛使用的一个经典案例，它要求输入3个正数a、b、c作为三角形的3条边，**判断这3个数构成的是一般三角形、等边三角形、等腰三角形，还是无法构成三角形。**如果使用等价类划分法设计三角形程序的测试用例，则需要**将所有输入数据划分为不同的等价类。**

案例分析

对该案例进行分析，程序要求输入3个数，并且是正数，在输入3个正数的基础上判断这3个数能否构成三角形；如果能构成三角形，再判断它构成的三角形是一般三角形、等腰三角形还是等边三角形。如此可以按照相关步骤将输入情况划分为不同的等价类。



2.1.3 实例二：三角形问题的等价类划分



第1步

判断是否输入了3个数，可以将输入情况划分成**1个有效等价类，4个无效等价类**，具体如下。

- 有效等价类：输入3个数。
- 无效等价类：输入0个数。
- 无效等价类：只输入1个数。
- 无效等价类：只输入2个数。
- 无效等价类：输入超过3个数。

第2步

在输入3个数的基础上，判断3个数是否为正数，可以将输入情况划分为**1个有效等价类，3个无效等价类**，具体如下。

- 有效等价类：3个数都是正数。
- 无效等价类：有1个数小于等于0。
- 无效等价类：有2个数小于等于0。
- 无效等价类：3个数都小于等于0。



2.1.3 实例二：三角形问题的等价类划分



第3步

在输入3个正数的基础上，判断3个数是否能构成三角形，可以将输入情况划分为**1个有效等价类和1个无效等价类**，具体如下。

- 有效等价类：任意2个数之和大于第3个数。
- 无效等价类：其中2个数之和小于等于第3个数。

第4步

在3个数构成三角形的基础上，判断3个数是否能构成等腰三角形，可以将输入情况划分成**1个有效等价类和1个无效等价类**。

- 有效等价类：其中有2个数相等。
- 无效等价类：3个数均不相等。



2.1.3 实例二：三角形问题的等价类划分



第5步

在构成等腰三角形的基础上，判断这3个数能否构成等边三角形，可以将输入情况划分为**1个有效等价类和1个无效等价类**，具体如下。

- 有效等价类：3个数相等。
- 无效等价类：3个数不相等。





2.1.3 实例二：三角形问题的等价类划分



在前面分析中，一共将三角形程序的输入划分为了15个等价类，下面分别为等价类编号，并建立等价类表，[三角形程序输入的等价类表](#)如下表所示。

| 要求 | 有效等价类 | 有效等价类编号 | 无效等价类 | 无效等价类编号 |
|---------------|----------------------------|---------|-----------------|---------|
| 输入3个数 | 输入3个数 | 1 | 输入0个数 | 2 |
| | | | 只输入1个数 | 3 |
| | | | 只输入2个数 | 4 |
| | | | 输入超过3个数 | 5 |
| 3个数是否都是正数 | 3个数都是正数 | 6 | 有1个数小于等于0 | 7 |
| | | | 有2个数小于等于0 | 8 |
| | | | 3个数都小于等于0 | 9 |
| 3个数是否能构成三角形 | 任意2个数之和大于第3个数 | 10 | 其中2个数之和小于等于第3个数 | 11 |
| 3个数是否能构成等腰三角形 | 其中有2个数相等, $a=b a=c b=c$ | 12 | 3个数均不相等 | 13 |
| 3个数是否能构成等边三角形 | 3个数相等, $a=b=c$ | 14 | 3个数不相等 | 15 |



2.1.3 实例二：三角形问题的等价类划分



下面根据三角形输入等价类表，设计测试用例覆盖等价类。首先设计覆盖有效等价类的测试用例，在设计时，既要考虑测试输入情况的全面性，又要考虑对有效等价类的覆盖情况，**覆盖有效等价类的测试用例如下表所示。**

| 测试用例 | 输入3个数 | 覆盖有效等价类的编号 |
|-------|-------|--------------|
| test1 | 1 2 3 | 1 6 |
| test2 | 3 4 5 | 1 6 10 |
| test3 | 6 6 8 | 1 6 10 12 |
| test4 | 6 6 6 | 1 6 10 12 14 |



2.1.3 实例二：三角形问题的等价类划分



覆盖有效等价类的测试用例表设计了4个测试用例覆盖了全部的有效等价类，无效等价类测试用例的设计原则与有效等价类测试用例的设计原则相同，[覆盖无效等价类的测试用例](#)如下表所示。

| 测试用例 | 输入数值 | 覆盖无效等价类的编号 |
|--------|----------|------------|
| test1 | -1 -1 -1 | 9 |
| test2 | -1 -1 5 | 8 |
| test3 | -1 4 5 | 7 |
| test4 | 输入0个数据 | 2 |
| test5 | 1 | 3 |
| test6 | 1 2 | 4 |
| test7 | 1 3 4 | 11 |
| test8 | 1 2 3 4 | 5 |
| test9 | 3 4 5 | 13 |
| test10 | 3 3 5 | 15 |



2.1.4 实例三：余额宝提现的等价类划分



掌握余额宝提现的等价类划分，能够对余额宝的提现功能进行测试



2.1.4 实例三：余额宝提现的等价类划分



案例描述

余额宝是一个余额增值服务和活期资金管理服务产品，可以把一些零钱存入余额宝产生利息，也可以将余额宝中的钱提现。**余额宝的提现方式有2种：快速到账（2小时），每日最高提现额度为10000元；普通到账，可提取金额为余额宝中的最大余额，但到账时间会慢一些。**



2.1.4 实例三：余额宝提现的等价类划分



对余额宝的提现功能进行测试，首先对余额宝提现功能进行等价类划分。

1. 快速到账

如果选择快速到账，则可将提现功能划分为**1个有效等价类和2个无效等价类**，具体如下。

- 有效等价类： $0元 < \text{提现金额} \leq 10000元$ 。
- 无效等价类： $\text{提现金额} \leq 0元$ 。
- 无效等价类： $\text{提现金额} > 10000元$ 。

2. 普通到账

如果选择普通到账，则可将提现功能划分为**1个有效等价类和2个无效等价类**，具体如下。

- 有效等价类： $0元 < \text{提现金额} \leq \text{余额}$ 。
- 无效等价类： $\text{提现金额} \leq 0元$ 。
- 无效等价类： $\text{提现金额} > \text{余额}$ 。



>>> 2.1.4 实例三：余额宝提现的等价类划分

根据前面分析，余额宝提现功能一共可划分为6个等价类，[余额宝提现功能的等价类表](#)如下表所示。

| 功能 | 有效等价类 | 编号 | 无效等价类 | 编号 |
|------|----------------|----|-------------|----|
| 快速到账 | 0元<提现金额≤10000元 | 1 | 提现金额≤ 0元 | 2 |
| | | | 提现金额>10000元 | 3 |
| 普通到账 | 0<提现金额≤余额 | 4 | 提现金额≤ 0元 | 5 |
| | | | 提现金额>余额 | 6 |



>>> 2.1.4 实例三：余额宝提现的等价类划分

细分后的余额宝提现功能等价类表如下表所示。

| 功能 | 有效等价类 | 编号 | 无效等价类 | 编号 |
|---------------|----------------------|----|------------------|----|
| 快速到账 (第1次) | 0元<提现金额<=10000元 | 1 | 提现金额≤ 0元 | 2 |
| | | | 提现金额>10000元 | 3 |
| 快速到账 (第n次) | 0元<提现金额<=10000-已提现金额 | 7 | 提现金额≤ 0元 | 8 |
| | | | 提现金额>10000-已提现金额 | 9 |
| 普通到账 | 0元<提现金额≤余额 | 4 | 提现金额≤ 0元 | 5 |
| | | | 提现金额>余额 | 6 |



2.1.4 实例三：余额宝提现的等价类划分



下面根据建立的等价类表来设计测试用例进行测试，假如现在余额宝中有50000元余额，则[覆盖有效等价类的测试用例](#)和[覆盖无效等价类的测试用例](#)分别如下所示。

覆盖有效等价类的测试用例

| 测试用例 | 功能 | 金额/元 | 覆盖有效等价类编号 |
|-------|-------------------------|-------|-----------|
| test1 | 快速到账 (第1次) | 1000 | 1 |
| test2 | 快速到账 (第n次, 已提现2000元) | 7000 | 7 |
| test3 | 普通到账 | 40000 | 4 |



2.1.4 实例三：余额宝提现的等价类划分



覆盖无效等价类的测试用例

| 测试用例 | 功能 | 金额/元 | 覆盖无效等价类编号 |
|-------|------------------------|--------|-----------|
| test4 | 快速到账 (第1次) | -10000 | 2 |
| test5 | | 20000 | 3 |
| test6 | 快速到账 (第n次, 已提现2000) | -2000 | 8 |
| test7 | | 9000 | 9 |
| test8 | 普通到账 | -3000 | 5 |
| test9 | | 60000 | 6 |

2.2

边界值分析法



»» 2.2.1 边界值分析法概述



熟悉边界值分析法的概述，能够归纳**什么是边界值分析法**



2.2.1 边界值分析法概述



边界值分析法是对软件的输入或输出边界进行测试的一种方法，它通常作为等价类划分法的一种补充测试。对于软件来说，错误经常发生在输入或输出值的关键点，即从符合需求到不符合需求的关键点，因此边界值分析法在等价类的边界上执行软件测试工作，它的所有测试用例都是在等价类的边界处设计的。



2.2.1 边界值分析法概述



在等价类划分法中，无论是输入值还是输出值，都会有多个边界，而**边界值分析法是在这些边界附近寻找某些点作为测试值**，而不是在等价类内部选择测试值。在使用边界值分析法时，可以通过确定边界的3个点来设计测试用例，这3个点分别是**上点、离点和内点**。上点是指边界上的点，离点是指距离边界最近的点，内点是指需求给定范围内的点。

边界值分析方法与等价类划分方法的区别是：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**；另外，边界值分析不仅**考虑输入条件边界**，还要**考虑输出域边界**产生的测试情况。



2.2.1 边界值分析法概述



按照测试数据的有效性，将边界值分析法分为**标准边界值测试**和**健壮边界值测试**。

(1) 标准边界值测试只考虑**有效数据范围内的边界值**。

对于一个有n个变量的程序，保留其中一个变量，其取值为最小值(Min)、略高于最小值(Min+)、正常值(Normal)、略低于最大值(Max-)、最大值(Max)，让其余变量取正常值，标准边界值分析测试程序会产生 $4n+1$ 个测试用例。

(2) 健壮边界值测试会考虑**有效和无效数据范围内的边界值**。

因此，对于一个含有n个变量的程序，保留其中一个变量，其取值为，其取值为略低于最小值(Min-)、最小值(Min)、略高于最小值(Min+)、正常值(Normal)、略低于最大值(Max-)、最大值(Max)，略高于最大值(Max+)，让其余变量取正常值，健壮边界值分析测试程序会产生 $6n+1$ 个测试用例。



2.2.1 边界值分析法概述



例如，输入条件规定取值范围为1~100，则可以选取1、2、50、99和100作为测试值。如果选取7个测试值，则在取值范围两侧再各选取一个测试值，可选取0、1、2、50、99、100和101作为测试值，1~100边界值选取如下表所示。

| 选取方案 | 选取数据 | | | | | |
|-------|------|---|----|----|-----|-----|
| 选取5个值 | 1 | 2 | 50 | 99 | 100 | |
| 选取7个值 | 0 | 1 | 2 | 50 | 99 | 100 |



- (1) 如果输入条件规定了**值的范围**, 则应选取**刚达到范围边界的值**, 以及**刚刚超越边界的值**作为测试输入数据。
- (2) 如果输入条件规定了**值的个数**, 则用**略低于最小值(Min-)**、**最小值(Min)**、**略高于最小值(Min+)**、**正常值(Normal)**、**略低于最大值(Max-)**、**最大值(Max)**、**略高于最大值(Max+)**作为测试数据。
- (3) 将原则(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**。
- (4) 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**集合的第一个和最后一个元素**作为测试用例
- (5) 如果程序用了一个**内部结构**, 应取这个**内部数据结构的边界值**作为测试用例。
- (6) 分析规格说明, 找出**其他可能的边界条件**。



2.2.2 实例一：QQ账号合法性的边界值分析



在前面，使用等价类划分法设计了QQ账号的测试用例。在实际的测试工作中，通常有大量的缺陷出现在输入或输出范围的边界上，而不是出现在输入或输出范围的内部，下面针对QQ账号的边界情况来设计测试用例。

通过分析需求可以知道，QQ账号要求是6~10位自然数，根据[上点](#)、[离点](#)和[内点](#)即可[确定边界范围](#)，具体如下。

- 上点：6、10。
- 离点：5、7、9、11。
- 内点：8。



2.2.2 实例一：QQ账号合法性的边界值分析



根据上述边界范围即可设计测试用例，[QQ账号边界值分析测试用例如下表所示。](#)

| 测试用例编号 | 测试数据 | 被测边界值 | 预期结果 |
|--------|-------------|-------|---------|
| test1 | 14725 | 5 | QQ账号不合法 |
| test2 | 123456 | 6 | QQ账号合法 |
| test3 | 1345678 | 7 | QQ账号合法 |
| test4 | 23616666 | 8 | QQ账号合法 |
| test5 | 564789100 | 9 | QQ账号合法 |
| test6 | 3316556666 | 10 | QQ账号合法 |
| test7 | 33165566897 | 11 | QQ账号不合法 |



2.2.3 实例二：三角形问题的边界值分析



在2.1.3节中，讲解了三角形问题的等价类划分，在等价类划分中，除了要求输入数据为3个正数外，没有给出其他限制条件。如果要求三角形边长取值范围为1~100，则可以[使用边界值分析法对三角形边界边长的边界进行测试](#)。在设计测试用例时，分别选取1、2、50、99、100这5个值作为测试值，[三角形边长的边界值分析测试用例如下表所示](#)。

| 测试用例 | 输入3个数 | 被测边界值 | 预期结果 |
|-------|-----------|-------|--------|
| test1 | 50 50 1 | 1 | 等腰三角形 |
| test2 | 50 50 2 | | 等腰三角形 |
| test3 | 50 50 50 | 无 | 等边三角形 |
| test4 | 50 50 99 | 100 | 等腰三角形 |
| test5 | 50 50 100 | | 不构成三角形 |



2.2.4 实例三：余额宝提现的边界值分析



在 2.1.4 小节中，讲解了余额宝提现功能的等价类划分，余额宝快速到账的日提现额度最高为 10000 元，如果余额宝中的余额小于 1000 元，则普通到账的提现额度最高为余额。假设余额宝中余额为 50000 元，则在进行边界值分析时，如果是第 1 次快速到账提现，则分别对 **0** 和 **10000** 这 2 个边界值进行测试，分别取**-1**、**0**、**1**、**5000**、**9999**、**10000**、**10001** 这 7 个值作为测试值；如果是第 n 次快速到账提现（假设已提现 2000 元），则分别对 **0** 和 **8000** 这 2 个边界值进行测试，分别取**-1**、**0**、**1**、**5000**、**7999**、**8000**、**8001** 这 7 个值作为测试值；如果是普通到账提现，则对 **0** 和 **50000** 这 2 个边界值进行测试，分别取**-1**、**0**、**1**、**20000**、**49999**、**50000**、**50001** 这 7 个值作为测试值。



2.2.4 实例三：余额宝提现的边界值分析



根据前面分析即可设计测试用例，余额宝提现边界值分析的测试用例如下表所示。

| 测试用例 | 功能 | 金额 | 被测边界 | 预期输出 |
|-------|---------------|-------|-------|-------|
| test1 | 快速到账 (第1次) | -1 | 0 | 无法提现 |
| test2 | | 0 | | 无法提现 |
| test3 | | 1 | | 1 |
| test4 | | 5000 | 无 | 5000 |
| test5 | | 9999 | 10000 | 9999 |
| test6 | | 10000 | | 10000 |
| test7 | | 10001 | | 无法提现 |



2.2.4 实例三：余额宝提现的边界值分析



| 测试用例 | 功能 | 金额 | 被测边界 | 预期输出 |
|--------|---------------|------|------|------|
| test8 | 快速到账 (第n次) | -1 | 0 | 无法提现 |
| test9 | | 0 | | 无法提现 |
| test10 | | 1 | | 1 |
| test11 | | 5000 | 无 | 5000 |
| test12 | | 7999 | 8000 | 7999 |
| test13 | | 8000 | | 8000 |
| test14 | | 8001 | | 无法提现 |



2.2.4 实例三：余额宝提现的边界值分析



| 测试用例 | 功能 | 金额 | 被测边界 | 预期输出 |
|--------|------|-------|-------|-------|
| test15 | 普通到账 | -1 | 0 | 无法提现 |
| test16 | | 0 | | 无法提现 |
| test17 | | 1 | | 1 |
| test18 | | 20000 | 无 | 20000 |
| test19 | | 49999 | 50000 | 49999 |
| test20 | | 50000 | | 50000 |
| test21 | | 50001 | | 无法提现 |

2.3

因果图法与决策表法





2.3.1 因果图法概述



前面介绍的等价类划分方法和边界值分析法都是着重考虑输入条件，但未考虑输入条件之间的联系、相互组合等。而因果图法是一种解决办法，它采用适合于描述对于多种条件的组合，相应产生多个动作形式来考虑测试用例。



2.3.1 因果图法概述



因果图法是一种利用图解法分析输入的各种组合情况的测试方法，它考虑了输入条件的各种组合及输入条件之间的相互制约关系，并考虑了输出情况。例如，某一软件要求输入的地址具体到市区，例如“北京→昌平区” “天津→南开区”，其中第二个输入受到第一个输入的约束，输入的城区只能在输入的城市中选择，否则输入的地址无效。像这样多个输入之间有相互制约关系的情况，就无法使用等价类划分法和边界值分析法设计测试用例。因果图法就是为了解决多个输入之间的作用关系而产生的测试用例设计方法。



2.3.1 因果图法概述

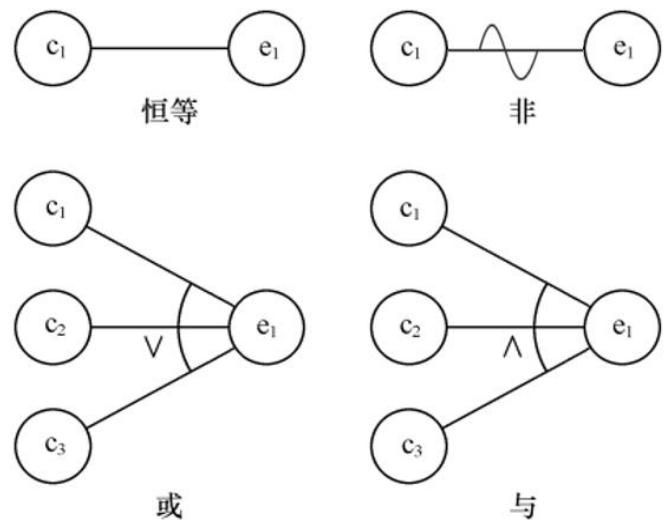


下面介绍如何使用因果图展示多个输入、输出之间的关系，并且学习如何通过因果图法设计测试用例。

1. 使用因果图展示多个输入、输出之间的关系

因果图需要处理输入之间的作用关系，还要考虑输出情况，因此它包含了复杂的逻辑关系。这些复杂的逻辑关系通常用图来展现，这些图就是因果图。

因果图使用一些简单的逻辑符号和直线将程序的原因（输入）与结果（输出）连接起来，一般原因用 c_i 表示，结果用 e_i 表示， c_i 与 e_i 可以取值 “0” 或 “1”，其中 “0” 表示状态不出现，“1” 表示状态出现。 c_i 与 e_i 之间有恒等、非 (\sim)、或 (\vee)、与 (\wedge) 这4种关系，如下图所示。





2.3.1 因果图法概述



上一页中展示了输入、输出之间的4种关系，每种关系的具体含义如下。

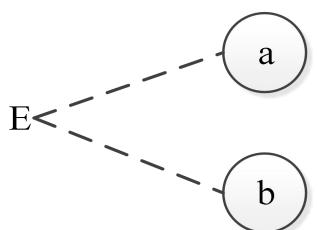
- **恒等**: 在恒等关系中，要求程序有一个输入和一个输出，输出与输入保持一致。若 c_1 为1，则 e_1 也为1，若 c_1 为0，则 e_1 也为0。
- **非**: 非使用符号“~”表示，在非关系中，要求程序有一个输入和一个输出，输出是输入的取反。若 c_1 为1，则 e_1 为0，若 c_1 为0，则 e_1 为1。
- **或**: 或使用符号“ \vee ”表示，或关系可以有多个输入，只要这些输入中有一个为1，则输出为1，否则输出为0。
- **与**: 与使用符号“ \wedge ”表示，与关系也可以有多个输入，但只有这些输入全部为1，输出才能为1，否则输出为0。



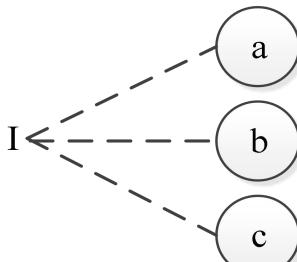
2.3.1 因果图法概述



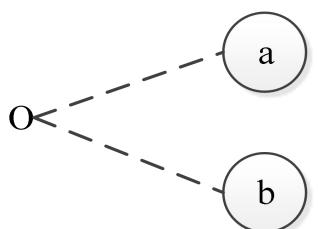
在软件测试中，如果程序有多个输入，那么除了输入与输出之间的作用关系之外，这些输入之间往往也会存在某些依赖关系，即某些输入条件本身不能同时出现或某一种输入可能会影响其他输入。例如，某一软件用于统计体检信息，在输入个人信息时，性别只能输入男或女，这两种输入不能同时存在，而且如果输入性别为女，那么体检项就会受到限制。这些依赖关系在软件测试中称为“约束”，约束的类别可分为4种：E（Exclusive，异）**、I（at least one，或）**、O（one and only one，唯一）**、R（Requires，要求）**。在因果图中，用特定的符号表明这些约束关系，多个输入之间的约束符号如下图所示。********



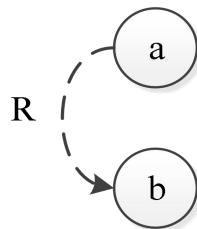
(1) 异



(2) 或



(3) 唯一



(4) 要求



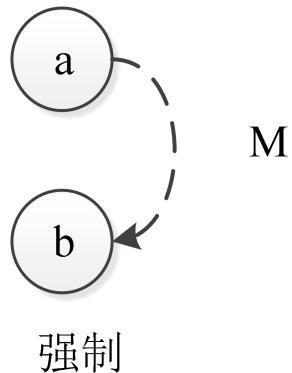
2.3.1 因果图法概述



上一页中的图展示了多个输入之间的约束符号，这些约束关系的含义具体如下所示。

- E (异) : a和b中最多只能有一个为1, 即a**和b不能同时为1**。
- I (或) : a、b和c中至少有一个必须是1, 即a、b、c**不能同时为0**。
- O (唯一) : a**和b有且仅有一个为1**。
- R (要求) : a**和b必须保持一致**, 即a为1时, b也必须为1, a为0时, b也必须为0。

上面这4种都是关于输入条件的约束。除了输入条件, 输出条件也会相互约束。**输出条件的约束只有一种, 即M (Mask, 强制)**。在因果图中, 使用特定的符号表示输出条件之间的强制约束关系, 如下图所示。



在输出条件的强制约束关系中, 如果a**为1**, 则b**强制为0**, 如果a**为0**, 则b**强制为1**。



2.3.1 因果图法概述



2. 使用因果图法设计测试用例的步骤

因果图法设计测试用例的5个步骤

(1) 分析程序规格说明书描述内容，确定程序的输入与输出，即确定“原因”和“结果”。

(2) 分析输入与输入之间、输入与输出之间的对应关系，将这些输入与输出之间的关系使用因果图表示出来。

(3) 由于语法与环境的限制，有些输入与输入之间、输入与输出之间的组合情况是不可能出现的，对于这种情况，使用符号标记它们之间的限制或约束关系。

(4) 将因果图转换为决策表。（决策表将在下一小节学习）

(5) 根据决策表设计测试用例。



2.3.2 决策表法概述



掌握决策表法的概述，能够通过决策表设计出完整的测试用例集合



2.3.2 决策表法概述



在实际测试中，如果输入条件较多，再加上各种输入与输出之间相互作用关系，画出的因果图就会比较复杂，让人不易理解。为了避免这种情况出现，测试人员往往[使用决策表法代替因果图法](#)。

决策表也称为判定表，[其实质就是一种逻辑表](#)。在程序设计发展初期，判定表就已经被当作程序开发的辅助工具了，用于帮助开发人员设计开发模式和整理开发流程，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，利用决策表可以设计出完整的测试用例集合。



2.3.2 决策表法概述



下面以一个“图书阅读指南”为例来制作一个决策表，“图书阅读指南”指明了图书阅读过程中可能出现的情况，以及针对各种情况给读者的建议。在图书阅读过程中可能会出现3种情况：是否疲倦、是否对内容感兴趣、对书中内容是否感到不理解，如果回答是“是”，则使用“Y”标记，如果回答是“否”，则使用“N”标记。这3种情况可以有 $2^3=8$ 种组合，针对这8种组合，阅读指南给读者提供了4条建议：回到本章开头重读、继续读下去、跳到下一章去读、停止阅读并休息，据此制作的“图书阅读指南”决策表如下表所示。



2.3.2 决策表法概述



“图书阅读指南” 决策表

| 情况与建议 | | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
|-------|--------------|---|---|---|---|---|---|---|---|
| 情况 | 是否疲倦 | Y | Y | Y | Y | N | N | N | N |
| | 是否对内容感兴趣 | Y | Y | N | N | N | Y | Y | N |
| | 对书中内容是否感到不理解 | Y | N | N | Y | Y | Y | N | N |
| 建议 | 回到本章开头重读 | | | | | | √ | | |
| | 继续读下去 | | | | | | | √ | |
| | 跳到下一章去读 | | | | | √ | | | √ |
| | 停止阅读并休息 | √ | √ | √ | √ | | | | |



2.3.2 决策表法概述



决策表通常由4个部分组成，具体如下。

- 条件桩：用于列出问题的所有条件，除了某些问题对条件的先后次序有要求外，通常决策表中所列条件的先后次序都无关紧要。
- 条件项：条件桩的所有可能取值。
- 动作桩：对问题可能采取的动作，这些动作一般没有先后次序之分。
- 动作项：指出在条件项的各组取值情况下应采取的动作。

这4个组成部分对应到“图书阅读指南”决策表中，**条件桩**包括是否疲倦、是否对内容感兴趣、对书中内容是否感到不理解；**条件项**包括“Y”与“N”；**动作桩**包括回到本章开头重读、继续读下去、跳到下一章去读、停止阅读并休息；**动作项**是指在综合情况下所采取的具体动作，例如“√”表示确认执行综合情况下采取的具体动作，动作项与条件项紧密相关，它的取值取决于条件项的各组取值情况。

在决策表中，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一条规则，即**决策表中的每一列就是一条规则，每一列都可以用于设计一个测试用例**。根据决策表设计测试用例可以**避免遗漏**。



2.3.2 决策表法概述



在实际测试中，条件桩通常有多个，而且每个条件桩都有真、假2个条件项，有 n 个条件桩的决策表就会有 2^n 条规则。如果为每条规则都设计一个测试用例，不仅工作量大，而且有些工作量可能是重复的、无意义的。例如，在“图书阅读指南”决策表中，以第1、2条规则为例，第1条规则取值为：Y、Y、Y，执行结果为“停止阅读并休息”；第2条规则取值为：Y、Y、N，执行结果也为“停止阅读并休息”。对于这2条规则来说，前2个问题的取值相同，执行结果一样，因此第3个问题的取值对结果并无影响，这个问题就称为无关条件项，使用“-”表示。忽略无关条件项，可以将这2条规则进行合并。



2.3.2 决策表法概述



合并规则1与规则2如下图所示。

合并 →

| | |
|---|---|
| 1 | 2 |
| Y | Y |
| Y | Y |
| Y | N |
| | |
| | |
| √ | √ |

| |
|---|
| Y |
| Y |
| - |
| |
| |
| |
| √ |

规则1与规则2合并成了一条规则。由于合并之后的无关条件项包含其他条件项取值，因此具有相同动作的规则还可[进一步合并](#)，如下图所示。

合并 →

合并 →

| | |
|---|---|
| Y | Y |
| Y | Y |
| - | N |
| | |
| | |
| | |
| √ | √ |

| |
|---|
| Y |
| Y |
| - |
| |
| |
| |
| √ |

| | |
|---|---|
| Y | Y |
| Y | Y |
| - | Y |
| | |
| | |
| | |
| √ | √ |

上图中只是演示合并后的规则，其还可以与其他规则进一步合并，但规则1与规则2合并之后就不再存在于决策表中。

简化后的“图书阅读指南”决策表

| 情况与建议 | | 1 | 2 | 3 | 4 |
|-------|--------------|---|---|---|---|
| 情况 | 是否疲倦 | Y | N | N | N |
| | 是否对内容感兴趣 | Y | N | Y | Y |
| | 对书中内容是否感到不理解 | - | - | Y | N |
| 建议 | 回到本章开头重读 | | | √ | |
| | 继续读下去 | | | | √ |
| | 跳到下一章去读 | | √ | | |
| | 停止阅读并休息 | √ | | | |



2.3.2 决策表法概述



构建决策表的步骤

- 步骤一：确定规则的个数。一般来说，有n个条件的决策表有 2^n 个规则。
- 步骤二：列出所有的条件桩和动作桩。
- 步骤三：填入条件项。
- 步骤四：填入动作项，得到初始决策表。
- 步骤五：简化决策表，合并相似规则。

2.3.3 实例一：字符问题的因果图与决策表绘制



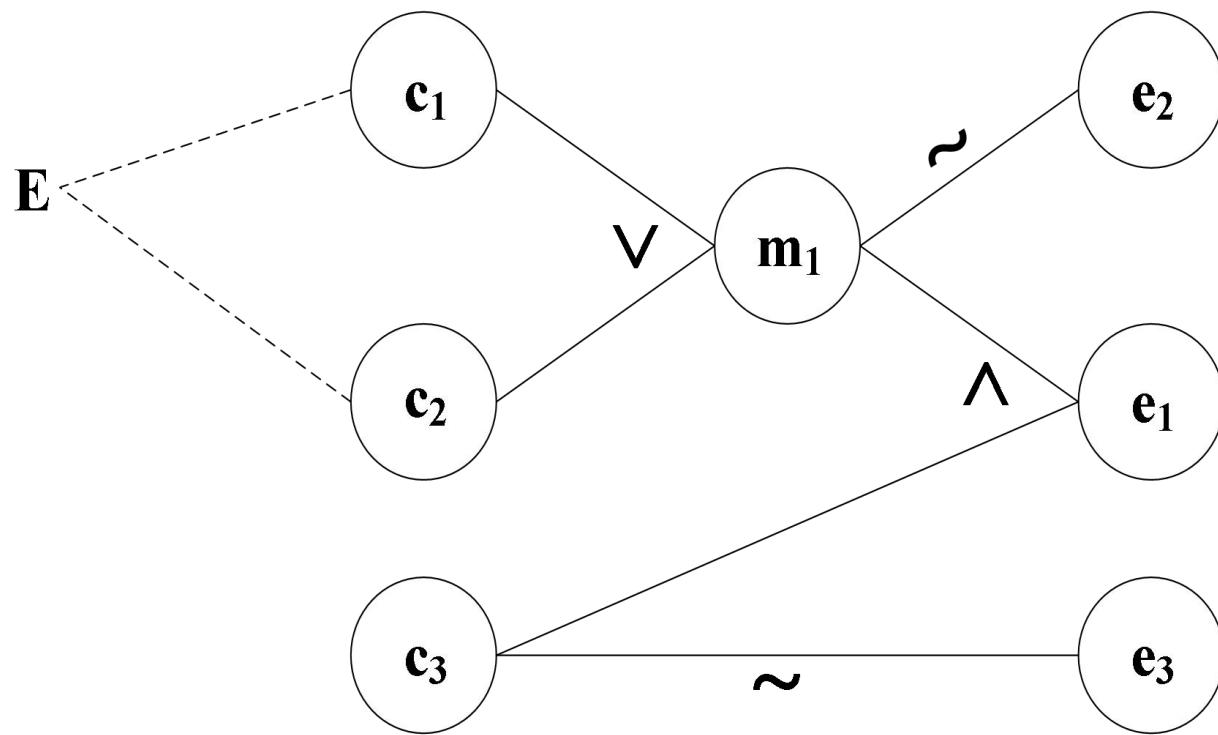
程序的规格说明要求：输入的第一个字符必须是A或B，第二个字符必须是一个数字，此情况下进行文件的修改；如果第一个字符不是A或B，则给出信息L，如果第二个字符不是数字，则给出信息M

| 原因 | 结果 | | |
|-----------|-------|-------|-------|
| 第一个字符是A | c_1 | 修改文件 | e_1 |
| 第一个字符是B | c_2 | 给出信息L | e_2 |
| 第二列字符是一数字 | c_3 | 给出信息M | e_3 |

2.3.3 实例一：字符问题的因果图与决策表绘制



程序规格说明因果图如图所示。其中， m_1 为中间状态，考虑到 c_1 和 c_2 不可能同时为1，因此，在 c_1 和 c_2 上施加E约束



2.3.3 实例一：字符问题的因果图与决策表绘制



字符问题的决策表

| 原因与结果 | | 1 | 2 | 3 | 4 | 5 | 6 |
|-------|-------|---|---|---|---|---|---|
| 原因 | c_1 | Y | Y | N | N | N | N |
| | c_2 | N | N | Y | Y | N | N |
| | c_3 | Y | N | Y | N | Y | N |
| 结果 | e_1 | √ | | √ | | | |
| | e_2 | | | | | √ | √ |
| | e_3 | | √ | | √ | | √ |

2.3.3 实例一：字符问题的因果图与决策表绘制

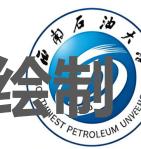


字符问题的测试用例

| 测试用例 | 第一个字符 | 第二个字符 | 预期结果 |
|-------|-------|-------|-------------------------------|
| test1 | A | 6 | e ₁ |
| test2 | A | a | e ₃ |
| test3 | B | 9 | e ₁ |
| test4 | B | P | e ₃ |
| test5 | C | 5 | e ₂ |
| test6 | D | G | e _{2, e₃} |



2.3.4 实例二：零食自动售货机售货情况的因果图与决策表绘制



- 掌握零食自动售货机售货情况的因果图与决策表的绘制，能够通过因果图与决策表设计零食自动售货机的测试用例



2.3.4 实例二：零食自动售货机售货情况的因果图与决策表绘制



为了演示因果图与决策表的应用，下面以零食自动售货机为例，讲解使用5角的硬币和1元的硬币在零食自动售货机上购买零食的过程，并绘制因果图与决策表。

假设零食自动售货机主要售卖糖果和饼干，其中糖果和饼干的单价均为5角，每次只能投入一枚5角的硬币或一枚1元的硬币，并按“糖果”按钮或“饼干”按钮进行购买，不能同时按“糖果”按钮和饼干“按钮”。零食自动售货机的具体使用说明如下。

- 如果顾客投入5角的硬币并按“糖果”按钮或“饼干”按钮，则送出糖果或饼干。
- 如果顾客投入1元的硬币并按“糖果”按钮或“饼干”按钮，且售货机有5角的零钱找回，则退出一枚5角的硬币并送出糖果或饼干。
- 如果顾客投入1元的硬币并按下“糖果”按钮或“饼干”按钮，若售货机没有5角的零钱找回，则亮红灯，然后退出1元的硬币，无法送出糖果或饼干。



2.3.4 实例二：零食自动售货机售货情况的因果图与决策表绘制



通过分析前面4条使用说明，列出原因与结果，零食自动售货机的原因与结果如下表所示。

| 原因 | | 结果 | |
|-------------|-------|------------------|-------|
| 投入5角的硬币 | c_1 | 送出糖果 | e_1 |
| 投入1元的硬币 | c_2 | 送出饼干 | e_2 |
| 按“糖果”按钮 | c_3 | 退出5角的硬币 | e_3 |
| 按“饼干”按钮 | c_4 | 退出1元的硬币 | e_4 |
| 售货机有5角的零钱找回 | c_5 | 售货机没有5角的零钱找回，亮红灯 | e_5 |



2.3.4 实例二：零食自动售货机售货情况的因果图与决策表绘制



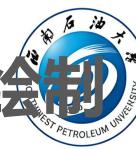
西南石油大學
Southwest Petroleum University

零食自动售货机的决策表

| 原因与结果 | | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
|-------|----------------|---|---|---|---|---|---|---|---|
| 原因 | C ₁ | Y | Y | Y | Y | | | | |
| | C ₂ | | | | | Y | Y | Y | Y |
| | C ₃ | Y | | N | | Y | | Y | |
| | C ₄ | | Y | | N | | Y | | Y |
| | C ₅ | N | N | Y | Y | Y | Y | N | N |
| 结果 | e ₁ | √ | | | | √ | | | |
| | e ₂ | | √ | | | | √ | | |
| | e ₃ | | | √ | √ | √ | √ | | |
| | e ₄ | | | | | | | √ | √ |
| | e ₅ | | | | | | | √ | √ |



2.3.4 实例二：零食自动售货机售货情况的因果图与决策表绘制



西南石油大學
Southwest Petroleum University

简化后的零食自动售货机的决策表

| 原因与结果 | | 1 | 2 | 3 | 4 | 5 | 6 |
|-------|----------------|---|---|---|---|---|---|
| 原因 | C ₁ | Y | Y | Y | | | |
| | C ₂ | | | | Y | Y | Y |
| | C ₃ | Y | | - | Y | | - |
| | C ₄ | | Y | - | | Y | - |
| | C ₅ | N | N | Y | Y | Y | N |
| 结果 | e ₁ | √ | | | √ | | |
| | e ₂ | | √ | | | √ | |
| | e ₃ | | | √ | √ | √ | |
| | e ₄ | | | | | | √ |
| | e ₅ | | | | | | √ |



2.3.4 实例二：零食自动售货机售货情况的因果图与决策表绘制



根据合并后的决策表设计6个零食自动售货机的测试用例，如下表所示。

| 测试用例 | 投入硬币 | 选择零食 | 是否有零钱 | 预期结果 |
|-------|------|------|-------|-------------|
| test1 | 5角 | 选择糖果 | 没有 | 送出糖果 |
| test2 | 5角 | 选择饼干 | 没有 | 送出饼干 |
| test3 | 5角 | 不选择 | 有 | 退出5角硬币 |
| test4 | 1元 | 选择糖果 | 有 | 送出糖果并退出5角硬币 |
| test5 | 1元 | 选择饼干 | 有 | 送出饼干并退出5角硬币 |
| test6 | 1元 | 不选择 | 没有 | 亮红灯并退出1元硬币 |



2.3.5 实例三：三角形问题的因果图与决策表绘制



掌握三角形问题的因果图与决策表的绘制，能够通过因果图与决策表设计三角形的测试用例



2.3.5 实例三：三角形问题的因果图与决策表绘制



三角形问题是一个非常经典的案例，下面将使用三角形讲解决策表的绘制与测试用例的设计。三角形的三边是否能构成三角形，如果能构成三角形，那么是构成一般三角形、等腰三角形还是等边三角形，据此分析，假设三角形的三边分别为 a 、 b 、 c ，则三角形问题有4个原因：是否构成三角形、 $a=b?$ 、 $b=c?$ 、 $c=a?$ ；有5个结果：不构成三角形、一般三角形、等腰三角形、等边三角形、不符合逻辑，三角形问题的原因与结果如下表所示。

| 原因 | | 结果 | |
|---------|-------|--------|-------|
| 是否构成三角形 | c_1 | 不构成三角形 | e_1 |
| $a=b?$ | c_2 | 一般三角形 | e_2 |
| $b=c?$ | c_3 | 等腰三角形 | e_3 |
| $a=c?$ | c_4 | 等边三角形 | e_4 |
| | | 不符合逻辑 | e_5 |



2.3.5 实例三：三角形问题的因果图与决策表绘制

三角形决策表

| 原因与结果 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 |
|-------|----------------|---|---|---|---|---|---|---|---|----|----|----|----|----|----|----|
| 原因 | c ₁ | Y | Y | Y | Y | Y | Y | Y | N | N | N | N | N | N | N | N |
| | c ₂ | Y | N | Y | N | N | Y | Y | N | Y | N | Y | Y | N | Y | N |
| | c ₃ | Y | N | N | Y | N | Y | N | Y | Y | N | Y | N | N | Y | N |
| | c ₄ | Y | N | N | N | Y | N | Y | Y | Y | Y | N | Y | N | N | N |
| 结果 | e ₁ | | | | | | | | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| | e ₂ | | | ✓ | | | | | | | | | | | | |
| | e ₃ | | | | ✓ | ✓ | ✓ | | | | | | | | | |
| | e ₄ | ✓ | | | | | | | | | | | | | | |
| | e ₅ | | | | | | ✓ | ✓ | ✓ | | | | | | | |



2.3.5 实例三：三角形问题的因果图与决策表绘制



简化后的三角形决策表

| 原因与结果 | | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
|-------|-------|---|---|---|---|---|---|---|---|---|
| 原因 | c_1 | Y | Y | Y | Y | Y | Y | Y | Y | N |
| | c_2 | Y | N | Y | N | N | Y | Y | N | - |
| | c_3 | Y | N | N | Y | N | Y | N | Y | - |
| | c_4 | Y | N | N | N | Y | N | Y | Y | - |
| 结果 | e_1 | | | | | | | | | ✓ |
| | e_2 | | | ✓ | | | | | | |
| | e_3 | | | | ✓ | ✓ | ✓ | | | |
| | e_4 | ✓ | | | | | | | | |
| | e_5 | | | | | | ✓ | ✓ | ✓ | |



2.3.5 实例三：三角形问题的因果图与决策表绘制



三角形测试用例

| 测试用例 | a | b | c | 预期结果 |
|-------|---|---|---|--------|
| test1 | 3 | 3 | 3 | 等边三角形 |
| test2 | 3 | 4 | 5 | 一般三角形 |
| test3 | 3 | 3 | 4 | 等腰三角形 |
| test4 | 4 | 3 | 3 | 等腰三角形 |
| test5 | 3 | 4 | 3 | 等腰三角形 |
| test6 | ? | ? | ? | 不符合逻辑 |
| test7 | ? | ? | ? | 不符合逻辑 |
| test8 | ? | ? | ? | 不符合逻辑 |
| test9 | 1 | 2 | 3 | 不构成三角形 |



2.3.6 实例四：工资发放情况的因果图与决策表绘制



掌握工资发放情况的因果图与决策表的绘制，能够通过因果图与决策表设计员工工资的测试用例



2.3.6 实例四：工资发放情况的因果图与决策表绘制



某公司的薪资管理制度如下：员工工资分为年薪制与月薪制2种，员工的犯错类型包括普通错误与严重错误2种，如果是**年薪制的员工**，犯**普通错误扣款2%**，犯**严重错误扣款4%**；如果是**月薪制的员工**，犯**普通错误扣款5%**，犯**严重错误扣款10%**。该公司编写了一款软件用于对员工工资的计算发放，现在要对该软件进行测试。



2.3.6 实例四：工资发放情况的因果图与决策表绘制



对公司员工工资管理进行分析，可得出员工工资由4个因素决定，即年薪、月薪、普通错误、严重错误，其中年薪与月薪不可能并存，但普通错误与严重错误可以并存；而员工最终扣款结果有7种，即未扣款、扣款2%、扣款4%、扣款6%（2%+4%）、扣款5%、扣款10%、扣款15%（5%+10%）。由此总结出员工工资发放情况的原因与结果，如下表所示。

| 原因 | | 结果 | |
|------|-------|-------|-------|
| 年薪 | c_1 | 未扣款 | e_1 |
| 月薪 | c_2 | 扣款2% | e_2 |
| | | 扣款4% | e_3 |
| 普通错误 | c_3 | 扣款6% | e_4 |
| | | 扣款5% | e_5 |
| 严重错误 | c_4 | 扣款10% | e_6 |
| | | 扣款15% | e_7 |



2.3.6 实例四：工资发放情况的因果图与决策表绘制



员工工资原因与结果表中有4个原因，每个原因有“Y”和“N”2个取值，理论上可以组成 $2^4=16$ 条规则，由于c₁与c₂不能并存，所以只有 $2^3=8$ 条规则，如下表所示。

| 原因与结果 | | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
|-------|----------------|---|---|---|---|---|---|---|---|
| 原因 | c ₁ | Y | Y | Y | Y | | | | |
| | c ₂ | | | | | Y | Y | Y | Y |
| | c ₃ | N | Y | N | Y | N | Y | N | Y |
| | c ₄ | N | N | Y | Y | N | N | Y | Y |
| 结果 | e ₁ | √ | | | | √ | | | |
| | e ₂ | | √ | | | | | | |
| | e ₃ | | | √ | | | | | |
| | e ₄ | | | | √ | | | | |
| | e ₅ | | | | | | √ | | |
| | e ₆ | | | | | | | √ | |
| | e ₇ | | | | | | | | √ |



2.3.6 实例四：工资发放情况的因果图与决策表绘制



员工工资测试用例

| 测试用例 | 薪资制度 | 薪资 | 错误程度 | 扣款 |
|-------|------|--------|-------|-------|
| test1 | 年薪制 | 200000 | 无 | 0 |
| test2 | | 250000 | 普通 | 5000 |
| test3 | | 300000 | 严重 | 12000 |
| test4 | | 350000 | 普通+严重 | 21000 |
| test5 | 月薪制 | 8000 | 无 | 0 |
| test6 | | 10000 | 普通 | 500 |
| test7 | | 15000 | 严重 | 1500 |
| test8 | | 8000 | 普通+严重 | 1200 |



适合使用决策表设计测试用例的情况



适合于使用决策表设计测试用例的条件：

- (1) 规格说明以决策表形式给出，或是很容易转换成决策表。
- (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) 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。

决策表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，简明易懂，也可避免遗漏。其缺点是不能表达重复执行的动作，例如循环结构。



因果图和决策表的关系



因果图只是清晰地表达了需求分析的内容。如果要得到测试用例，就必须借助于决策表，也就是需要将因果图转化成决策表。

在因果图中已经分析了“因”和“果”，“因”和“果”直接就可以作为条件桩和动作桩。根据条件桩的取值得到条件项，利用条件项和因果图中原因与结果的关系，可以得到相应的规则，最终生成决策表。

2.4

正交实验设计法





掌握正交实验设计法的概述，能够通过正交实验设计法设计测试用例



2.4.1 正交实验设计法概述



正交实验设计法 (Orthogonal experimental design)

是指从大量的实验点中挑选出适量的、有代表性的点，依据 Galois 理论导出“正交表”，从而合理地安排实验的一种实验设计方法。正交实验设计法是研究多因素、多水平问题的一种实验方法，在上生物课时，经常会用这种方法研究植物的生长状况。一株植物的生长状况会受到多种因素的影响，包括种子质量等内部因素的影响，还包括阳光、空气、水分、土壤等外部因素的影响。在软件测试中，如果软件比较复杂，也可以利用正交实验法设计测试用例对软件进行测试。



2.4.1 正交实验设计法概述



正交实验设计法包含3个关键因素，具体如下。

3个关键因素

指标 ➤ 判断实验结果优劣的标准。



因子 ➤ 也称为因素，它是指所有影响实验指标的条件。



因子的状态 ➤ 也叫因子的水平，它是指因子变量的取值。





2.4.1 正交实验设计法概述



利用正交实验设计法设计测试用例时，可以按照以下3个步骤进行。

1. 提取因子，构造因子-状态表

分析软件的需求规格说明书得到影响软件功能的因子，确定因子可以有哪些取值，即**确定因子的状态**。例如，某一软件的运行受到操作系统和数据库的影响，因此影响其运行的因子有操作系统和数据库，而**操作系统有Windows、Linux、macOS这3个取值，数据库有MySQL、MongoDB、Oracle这3个取值**，所以**操作系统的因子状态数为3，数据库的因子状态数为3**。据此构造该软件运行功能的**因子-状态表**，如下表所示。

| 因子 | 因子的状态 | | |
|------|---------|---------|--------|
| 操作系统 | Windows | Linux | macOS |
| 数据库 | MySQL | MongoDB | Oracle |



2.4.1 正交实验设计法概述



2. 加权筛选，简化因子-状态表

在实际软件测试中，软件的因子及因子的状态会有很多，每个因子及其状态对软件的影响也大不相同，如果把这些因子及其状态都划分到因子-状态表中，最后生成的测试用例会相当庞大，从而会影响软件测试的效率。因此需要根据因子及其状态的重要程度进行加权筛选，选出重要的因子及其状态，简化因子-状态表。

加权筛选是指根据因子或因子的状态的重要程度、出现频率等因素计算因子和因子的状态的权值，权值越大，表明因子或因子的状态越重要，而权值越小，表明因子或因子的状态的重要性越小。加权筛选之后，可以去掉一部分权值较小的因子或因子的状态，使得最后生成的测试用例集缩减到允许的范围。



2.4.1 正交实验设计法概述



3. 构建正交表，设计测试用例

正交表的表示形式为 $L_n(t^c)$ ，具体说明如下。

- L 表示正交表。
- n 为正交表的行数，正交表的每一行可以用于设计一个测试用例，因此行数 n 也表示可以设计的测试用例的数目。
- c 表示正交实验的因子数目，即正交表的列数，因此正交表是一个 n 行 c 列的表。
- t 称为水平数，表示每个因子能够取得的最大值，即因子有多少个状态。

$$n = (t-1)*c + 1$$



2.4.1 正交实验设计法概述



例如 $L_4(2^3)$ 是较为简单的正交表，它表示该实验有3个因子，每个因子有2个状态，可以做4次实验。如果用0和1表示每个因子的2种状态，则该正交表就是一个4行3列的表。 $L_4(2^3)$ 正交表如下表所示。

| 行 | 列 | | |
|---|---|---|---|
| | 1 | 2 | 3 |
| 1 | 1 | 1 | 1 |
| 2 | 1 | 0 | 0 |
| 3 | 0 | 1 | 0 |
| 4 | 0 | 0 | 1 |



2.4.1 正交实验设计法概述



混合正交表往往难以确定测试用例的数目，此时用n表示测试用例的数目，这种情况下，读者可以[登录正交表的权威网站，查询n值](#)。正交表查询网站主页如下图所示。

https://support.sas.com/techsup/technote/ts723_Designs.txt

The screenshot shows a web browser window with the URL https://support.sas.com/techsup/technote/ts723_Designs.txt. The page displays three sets of orthogonal array designs:

- 2^3 n=4:
000
011
101
110
- $2^4 4^1$ n=8:
00000
00112
01011
01103
10013
10101
11002
11110
- 3^4 n=9:
0000
0121
0212
1022
1110
1201
2011
2102
2220



2.4.1 正交实验设计法概述



上一页的图中，读者[可以查询到不同因子数、不同水平数的正交表的n值](#)。在正交表查询网站查找到 $2^4 \times 4^1$ 的正交表n值为8。 $L8(2^4 \times 4^1)$ 的正交表设计如下表所示。

| 行 | 列 | | | | |
|---|---|---|---|---|---|
| | 1 | 2 | 3 | 4 | 5 |
| 1 | 0 | 0 | 0 | 0 | 0 |
| 2 | 0 | 0 | 1 | 1 | 2 |
| 3 | 0 | 1 | 0 | 1 | 1 |
| 4 | 0 | 1 | 1 | 0 | 3 |
| 5 | 1 | 0 | 0 | 1 | 3 |
| 6 | 1 | 0 | 1 | 0 | 1 |
| 7 | 1 | 1 | 0 | 0 | 2 |
| 8 | 1 | 1 | 1 | 1 | 0 |



2.4.1 正交实验设计法概述



对于受多因子、多水平影响的软件，正交实验法可以高效、适量地生成测试用例，减少测试工作量，并且利用正交实验法得到的测试用例具有一定的覆盖率，检错率可在50%以上。正交实验法虽然好用，但在选择正交表时要注意先确定实验因子、因子的状态及它们之间的交互作用，同时还要考虑实验的精度要求、费用、时长等因素。



2.4.2 实例一：微信Web页面运行环境正交实验设计



掌握微信Web页面运行环境正交实验设计，能够根据正交表生成微信Web页面运行环境的测试用例



2.4.2 实例一：微信Web页面运行环境正交实验设计



微信是一款手机App软件，但也有Web版微信可以登录。如果要测试微信Web页面运行环境，需要考虑多种因素，在众多的因素中，可以选出几个**影响比较大的因素**，例如**服务器、操作系统、网络和浏览器**。对于选取出的4个影响因素，每个因素又有不同的取值。同样，在每个因素的多个值中，可以选出几个比较重要的值，具体如下。

- 服务器：[IIS](#)、[Apache](#)、[Jetty](#)。
- 操作系统：[MacOS](#)、[Linux](#)、[Windows](#)。
- 网络：[4G](#)、[5G](#)、[5GA](#)。
- 浏览器：[Edge](#)、[FireFox](#)、[Chrome](#)。



2.4.2 实例一：微信Web页面运行环境正交实验设计



微信Web页面运行环境正交实验中有4个因子，即服务器、操作系统、插件、浏览器，每个因子又有3个水平，因此该正交表是一个4因子3水平正交表。通过查询正交表查询网站可得其n值为9，即 $L_9(3^4)$ 。如果按照上述水平的所列顺序，从左至右为每个水平编号0、1、2，则生成一个9行4列的正交表。 $L_9(3^4)$ 正交表如下表所示。

| 行 | 列 | | | |
|---|---|---|---|---|
| | 1 | 2 | 3 | 4 |
| 1 | 0 | 0 | 0 | 0 |
| 2 | 0 | 1 | 2 | 1 |
| 3 | 0 | 2 | 1 | 2 |
| 4 | 1 | 0 | 2 | 2 |
| 5 | 1 | 1 | 1 | 0 |
| 6 | 1 | 2 | 0 | 1 |
| 7 | 2 | 0 | 1 | 1 |
| 8 | 2 | 1 | 0 | 2 |
| 9 | 2 | 2 | 2 | 0 |



2.4.2 实例一：微信Web页面运行环境正交实验设计



$L_9(3^4)$ 正交表中的水平编号分别代表因子的不同取值，将因子、状态映射到正交表，可生成具体的测试用例，[微信Web页面运行环境测试用例](#)如下表所示。

| 行 | 列 | | | |
|---|--------|---------|-----|---------|
| | 服务器 | 操作系统 | 网络 | 浏览器 |
| 1 | IIS | MacOS | 4G | Edge |
| 2 | IIS | Windows | 5G | Chrome |
| 3 | IIS | Linux | 5GA | FireFox |
| 4 | Apache | MacOS | 5G | FireFox |
| 5 | Apache | Windows | 5GA | Edge |
| 6 | Apache | Linux | 4G | Chrome |
| 7 | Jetty | MacOS | 5GA | Chrome |
| 8 | Jetty | Windows | 4G | FireFox |
| 9 | Jetty | Linux | 5G | Edge |



2.4.3 实例二：用户筛选功能正交实验设计



掌握用户筛选功能正交实验设计，能够根据正交表生成用户筛选功能的测试用例



2.4.3 实例二：用户筛选功能正交实验设计



假设有一个招聘求职软件，招聘人员可以根据多个因素来筛选求职者。下面选择**城市、招聘岗位、学历、计算机等级和工作经验作为关键因素**，每个因素都有不同的取值，具体如下。

- 城市：**北京、上海、深圳、广州。**
- 招聘岗位：**产品运营、产品经理、软件测试、软件工程师。**
- 学历：**高中、专科、本科、研究生。**
- 计算机等级：**计算机一级、计算机二级、计算机三级、计算机四级。**
- 工作经验：**1年、2年、3年、4年。**



2.4.3 实例二：用户筛选功能正交实验设计



在官网[下载Allpairs工具。](https://www.softpedia.com/dyn-search.php?search_term=allpairs) https://www.softpedia.com/dyn-search.php?search_term=allpairs

1

The screenshot shows the Softpedia search interface. The search bar at the top contains 'allpairs'. Below it, the search results are displayed under the heading 'Windows Search'. A single result is shown: 'Allpairs June 7, 2023'. The result card includes the application icon, the name 'Allpairs', the release date 'June 7, 2023', a brief description 'A console application designed for scenarios where testing complex systems with numerous variables...', the number of downloads '26,001', and the file size '3.7 MB'. The entire result card is highlighted with a red border.

3

The screenshot shows the 'Allpairs Download Links' page on Softpedia. It features a large 'Softpedia Secure Download (US)' button, which is highlighted with a red border. Below the button, there is a note that the download is 'FREE of charge' and a statement that 'All files are original'. At the bottom of the page, there is a 'Genuine DOWNLOAD' seal.

2

The screenshot shows the 'Allpairs FOR WINDOWS' page on Softpedia. It includes a prominent 'DOWNLOAD NOW' button, which is highlighted with a red border. Below the button, there is a brief description of the application: 'A console application designed for scenarios where testing complex systems with numerous variables for various combinations of printers, print options, and document types. #TEST CASE GENERATOR'. To the right of the description is a 'SOFTPIEDIA 100% FREE' badge. The page also features a rating section, a sidebar with links like 'Add to watchlist', and a screenshot of the application's interface.

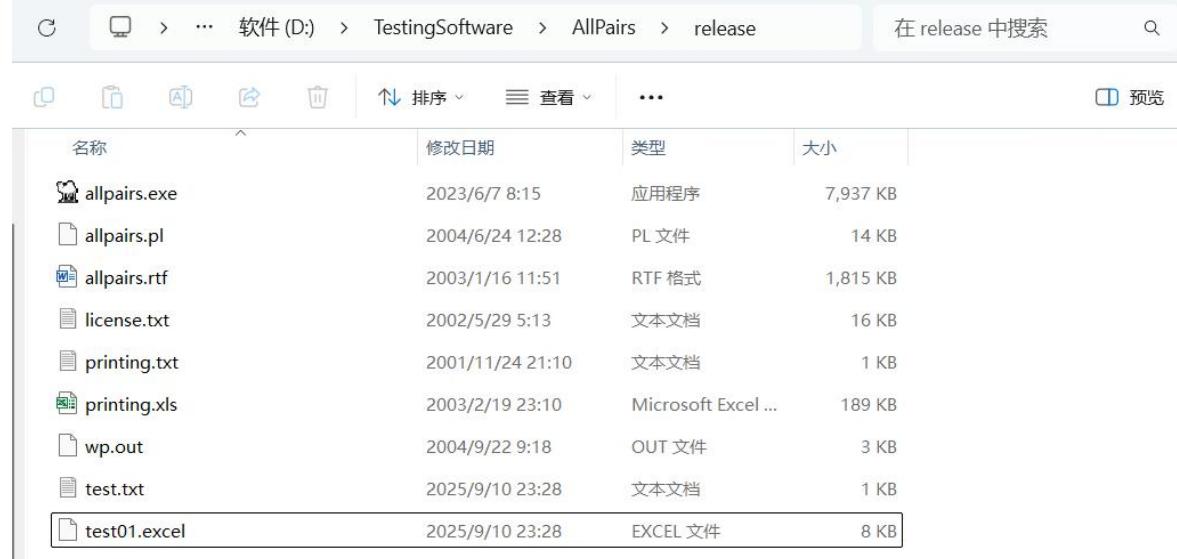
>>> 2.4.3 实例二：用户筛选功能正交实验设计

使用Allpairs工具自动生成一个5因子4水平的正交表。

4 test.txt文件

| 城市 | 专业 | 学历 | 计算机等级 | 工作经验 |
|----|-----|----|-------|-------|
| 北京 | 经济学 | 专科 | 计算机一级 | 1年 |
| 上海 | 法学 | 本科 | 计算机二级 | 2年 |
| 深圳 | 教育学 | 硕士 | 计算机三级 | 3年 |
| 广州 | 工学 | 博士 | 计算机四级 | 4年及以上 |

6 执行过后的目录



| 名称 | 修改日期 | 类型 | 大小 |
|--------------|------------------|---------------------|----------|
| allpairs.exe | 2023/6/7 8:15 | 应用程序 | 7,937 KB |
| allpairs.pl | 2004/6/24 12:28 | PL 文件 | 14 KB |
| allpairs.rtf | 2003/1/16 11:51 | RTF 格式 | 1,815 KB |
| license.txt | 2002/5/29 5:13 | 文本文档 | 16 KB |
| printing.txt | 2001/11/24 21:10 | 文本文档 | 1 KB |
| printing.xls | 2003/2/19 23:10 | Microsoft Excel ... | 189 KB |
| wp.out | 2004/9/22 9:18 | OUT 文件 | 3 KB |
| test.txt | 2025/9/10 23:28 | 文本文档 | 1 KB |
| test01.xlsx | 2025/9/10 23:28 | EXCEL 文件 | 8 KB |

cmd命令窗口

D:\TestingSoftware\AllPairs\release>allpairs.exe test.txt>test01.xlsx



2.4.3 实例二：用户筛选功能正交实验设计



test01.xlsx正交表

7

| 1 | TEST CASES | | | | | | |
|----|------------|----|-----|----|-------|-------|----------|
| 2 | case | 城市 | 专业 | 学历 | 计算机等级 | 工作经验 | pairings |
| 3 | 1 | 广州 | 工学 | 本科 | 计算机四级 | 2年 | 10 |
| 4 | 2 | 广州 | 经济学 | 硕士 | 计算机一级 | 1年 | 10 |
| 5 | 3 | 广州 | 教育学 | 博士 | 计算机三级 | 4年及以上 | 10 |
| 6 | 4 | 广州 | 法学 | 专科 | 计算机二级 | 3年 | 10 |
| 7 | 5 | 深圳 | 工学 | 硕士 | 计算机三级 | 3年 | 10 |
| 8 | 6 | 深圳 | 经济学 | 本科 | 计算机二级 | 4年及以上 | 10 |
| 9 | 7 | 深圳 | 教育学 | 专科 | 计算机四级 | 1年 | 10 |
| 10 | 8 | 深圳 | 法学 | 博士 | 计算机一级 | 2年 | 10 |
| 11 | 9 | 北京 | 工学 | 博士 | 计算机二级 | 1年 | 10 |
| 12 | 10 | 北京 | 经济学 | 专科 | 计算机三级 | 2年 | 10 |
| 13 | 11 | 北京 | 教育学 | 本科 | 计算机一级 | 3年 | 10 |
| 14 | 12 | 北京 | 法学 | 硕士 | 计算机四级 | 4年及以上 | 10 |
| 15 | 13 | 上海 | 工学 | 专科 | 计算机一级 | 4年及以上 | 10 |
| 16 | 14 | 上海 | 经济学 | 博士 | 计算机四级 | 3年 | 10 |
| 17 | 15 | 上海 | 教育学 | 硕士 | 计算机二级 | 2年 | 10 |
| 18 | 16 | 上海 | 法学 | 本科 | 计算机三级 | 1年 | 10 |

2.5

场景法



2.5.1 场景法概述



- 掌握场景法的概述，能够通过场景法设计测试用例



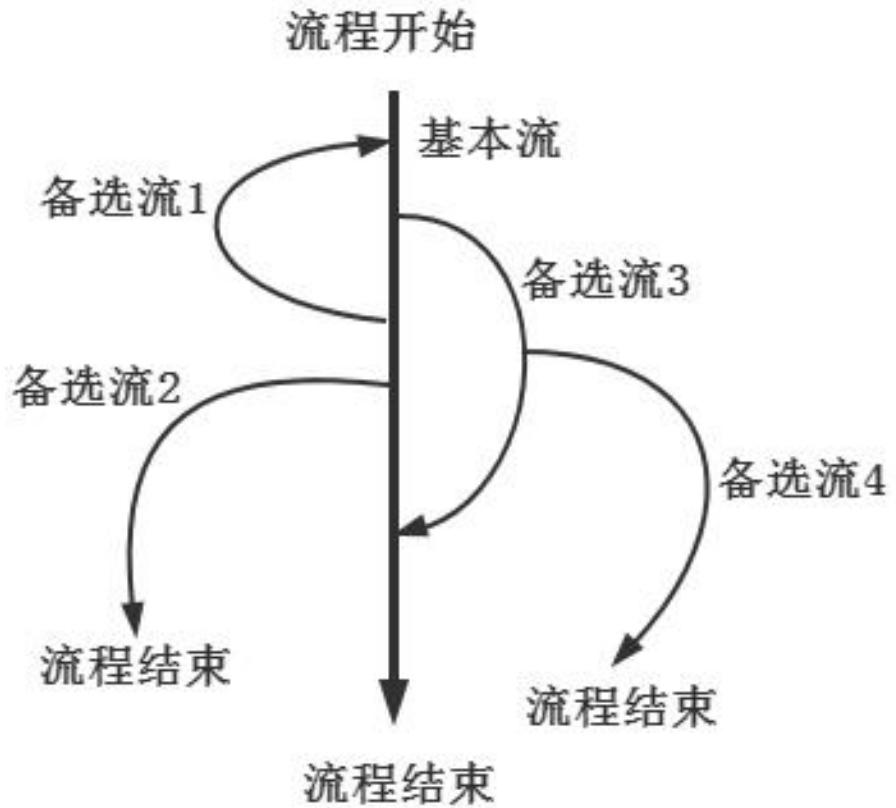
2.5.1 场景法概述



场景法也叫流程图法，是指通过模拟用户操作软件时的场景来对系统的功能或业务流程进行测试。场景法通常用于测试多个功能之间的组合使用情况，以及用于集成测试、系统测试和验收测试阶段。

2.5.1 场景法概述

根据用户操作流程的正确性来划分时，场景法将用户的操作流程分为基本流和备选流。基本流也称为有效流，用来模拟用户正确的操作流程；备选流也称为无效流、错误流，用来模拟用户错误的操作流程。基本流和备选流如下图所示。



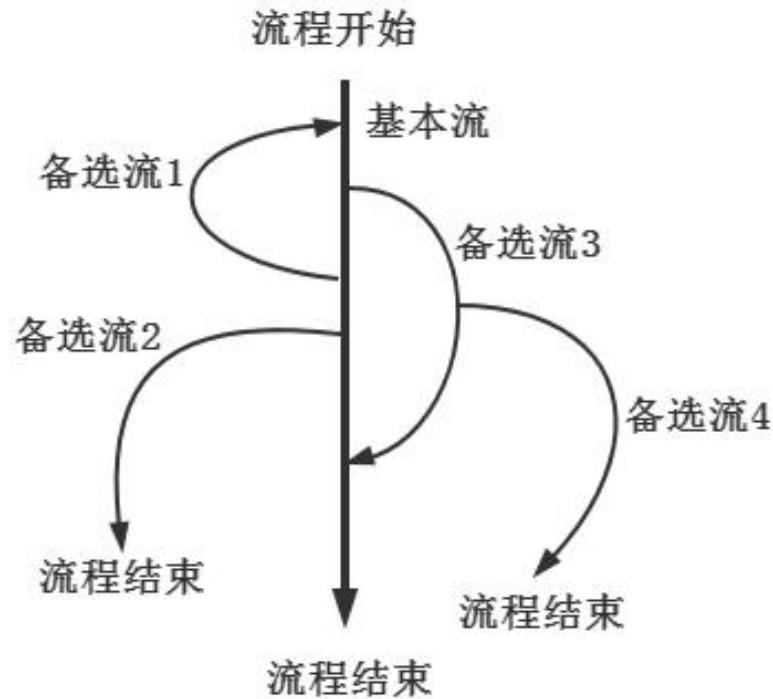


2.5.1 场景法概述



由上一页中的基本流和备选流图可知，基本流有1条，备选流有4条。备选流可以从基本流开始，例如备选流1、备选流2和备选流3；备选流也可以从备选流开始，例如备选流4。通过分析上一页中的基本流和备选流图，可以确定的测试场景如下。

- 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1→备选流3。
- 场景5：基本流→备选流1→备选流3→备选流4。
- 场景6：基本流→备选流2。
- 场景7：基本流→备选流3。
- 场景8：基本流→备选流3→备选流4。





2.5.1 场景法概述



在场景法中每一个场景是一条流程路径，根据[流程路径的数量即可设计测试用例](#)。使用场景法设计测试用例可以按照以下4个步骤进行。

- 步骤1：分析需求规格说明书。
- 步骤2：根据需求规格说明书[绘制流程图](#)。
- 步骤3：根据流程图[确定测试场景](#)。
- 步骤4：根据测试场景[设计测试用例](#)。

[流程图常用的符号名称与说明](#)如下表所示。

| 符号 | 名称 | 说明 |
|----|-------|------------|
| □ | 椭圆 | 表示流程的开始或结束 |
| □ | 平行四边形 | 表示流程的输入或输出 |
| □ | 长方形 | 表示处理或执行 |
| ◇ | 菱形 | 表示对某个条件的判断 |
| → | 箭头 | 表示流程进行的方向 |



2.5.2 实例一：电商网站购物场景分析



掌握电商网站购物场景分析，能够通过场景法测试用户在电商网站购物的过程



2.5.2 实例一：电商网站购物场景分析



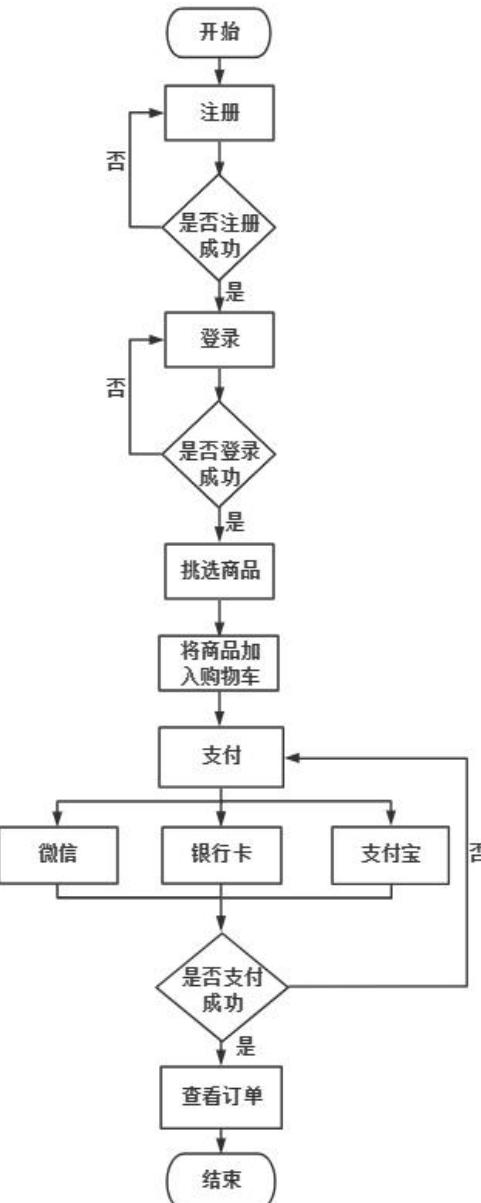
如今电商行业的发展非常迅速，许多公司开始研发电商网站，为用户提供更多的购物渠道。假设某公司研发了一个电商网站，现需要测试人员按照“注册→登录→挑选商品→将商品加入购物车→支付→查看订单”的流程进行测试。在使用电商网站进行购物时，首先进行注册，如果注册失败，则需要重新注册，直到注册成功后才可以登录电商网站。如果登录失败，则需要重新登录。该电商网站的支付方式有3种，分别是微信、银行卡和支付宝，如果这3种方式都支付失败，则需要返回支付环节重新支付，直到支付成功后才能查看订单。



2.5.2 实例一：电商网站购物场景分析



下面通过场景法测试用户在电商网站购物的过程。首先通过[分析前面的需求描述](#)，然后画出用户在电商网站购物的流程图。[购物流程图](#)如右图所示。





2.5.2 实例一：电商网站购物场景分析



分析上一页中的购物流程图可知，基本流有1条，备选流有3条，具体如下。

- 基本流：注册→登录→挑选商品→将商品加入购物车→支付→查看订单。
- 备选流1：注册失败。
- 备选流2：注册成功→登录失败。
- 备选流3：注册成功→登录成功→挑选商品→将商品加入购物车→支付失败。

通过对基本流和备选流的分析，可以得出4个测试场景，具体如下。

- 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2。
- 场景4：基本流+备选流3。



2.5.2 实例一：电商网站购物场景分析



在使用场景法设计测试用例时，每一个场景对应一个测试用例。下面根据这4个测试场景来设计测试用例，电商网站购物的测试用例如下表所示。

| 测试用例 | 测试场景 | 测试数据 | 预期结果 |
|-------|------|---------------------------|------|
| test1 | 场景1 | 有效的账号密码，微信、银行卡、支付宝均支付成功 | 成功购物 |
| test2 | 场景2 | 无效的注册账号 | 注册失败 |
| test3 | 场景3 | 账号或密码错误 | 登录失败 |
| test4 | 场景4 | 有效的账号和密码，用微信、银行卡、支付宝均支付失败 | 支付失败 |



2.5.3 实例二：ATM取款场景分析



掌握ATM取款场景分析，能够通过场景法测试某银行ATM取款业务流程



2.5.3 实例二：ATM取款场景分析



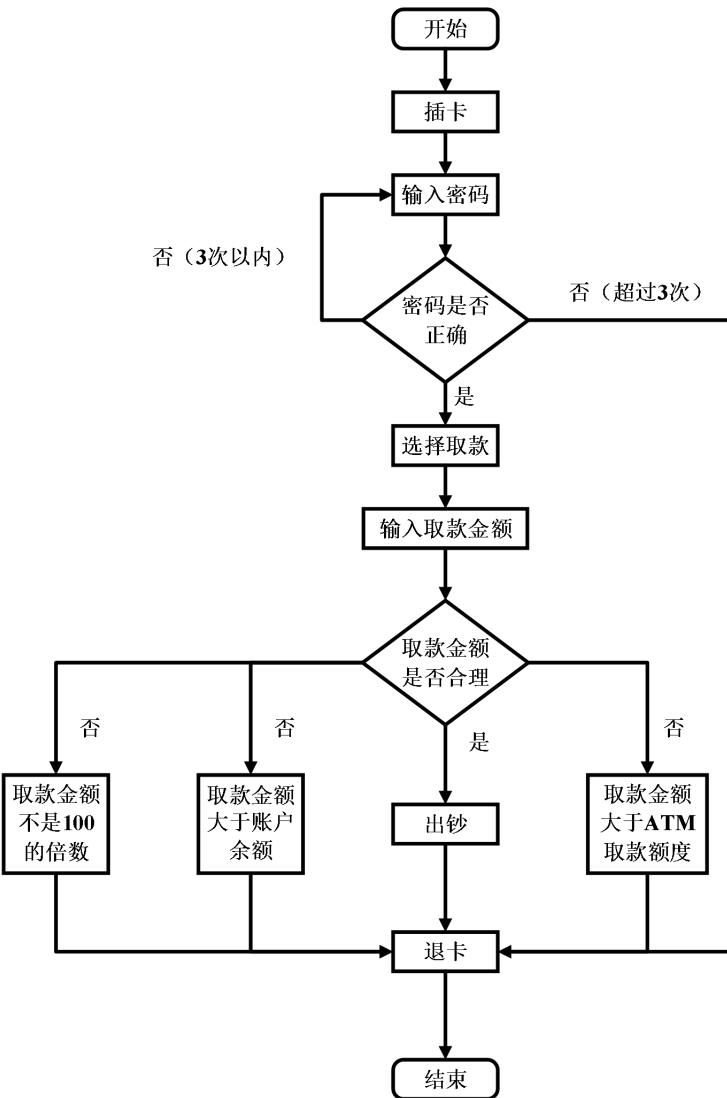
ATM(Automated Teller Machine, 自动柜员机) 可用于提取现金、查询存款余额、转账等。

假设需要使用场景法测试某银行ATM的取款业务流程，银行给出的需求规格说明是：用户在ATM中插入有效的银行卡，输入正确的密码后选择取款业务，然后输入取款金额，待出钞后选择退卡即可完成取款。在取款的过程中，如果出现以下4种情况将取款失败，结束流程。

- 密码输入错误的次数超过3次。
- 输入的取款金额不是100的倍数。
- 输入的取款金额大于账户余额。
- 输入的取款金额大于ATM取款额度。

2.5.3 实例二：ATM取款场景分析

下面通过前面需求规格说明，画出取款的流程图。ATM取款流程图如下图所示。





2.5.3 实例二：ATM取款场景分析



分析ATM取款流程图可知，基本流有1条，备选流有5条，具体如下。

- 基本流：插卡→输入密码→选择取款→输入取款金额→出钞→退卡。
- 备选流1：插卡→输入密码错误（3次以内）→选择取款→输入取款金额→出钞→退卡。
- 备选流2：插卡→输入密码错误（超过3次）→退卡。
- 备选流3：插卡→输入密码→选择取款→输入的取款金额不是100的倍数→退卡。
- 备选流4：插卡→输入密码→选择取款→输入的取款金额大于账户余额→退卡。
- 备选流5：插卡→输入密码→选择取款→输入的取款金额大于ATM取款额度→退卡。

通过对基本流和备选流的分析，可以得出6个测试场景，具体如下。

- 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3。
- 场景4：基本流+备选流1+备选流4。
- 场景5：基本流+备选流1+备选流5。
- 场景6：基本流+备选流2。



2.5.3 实例二：ATM取款场景分析



下面根据前面列出的6个测试场景来设计测试用例，假设测试的银行卡有效，密码为123456，账户余额为5000元，ATM取款额度为3000元。**ATM取款的测试用例如下表所示。**

| 测试用例 | 测试场景 | 测试数据 | 预期结果 |
|-------|------|--|--------------|
| test1 | 场景1 | 1.插入有效的银行卡； 2.输入密码123456； 3.输入取款金额为1000 | 取出1000元钞票并退卡 |
| test2 | 场景2 | 1.插入有效的银行卡； 2.输入密码123455（第1次输入）； 3.输入密码123450（第2次输入）； 4.输入密码123456（第3次输入）； 5.输入取款金额为1000 | 取出1000元钞票并退卡 |
| test3 | 场景3 | 1.插入有效的银行卡； 2.输入密码123455（第1次输入）； 3.输入密码123450（第2次输入）； 4.输入密码123456（第3次输入）； 5.输入取款金额为1551 | 取款失败并退卡 |



2.5.3 实例二：ATM取款场景分析



| 测试用例 | 测试场景 | 测试数据 | 预期结果 |
|-------|------|--|---------|
| test4 | 场景4 | 1.插入有效的银行卡； 2.输入密码123455（第1次输入）； 3.输入密码123450（第2次输入）； 4.输入密码123456（第3次输入）； 5.输入取款金额为6000 | 取款失败并退卡 |
| test5 | 场景5 | 1.插入有效的银行卡； 2.输入密码123455（第1次输入）； 3.输入密码123450（第2次输入）； 4.输入密码123456（第3次输入）； 5.输入取款金额为4000 | 取款失败并退卡 |
| test6 | 场景6 | 1.插入有效的银行卡； 2.输入密码123455（第1次输入）； 3.输入密码123450（第2次输入）； 4.输入密码123451（第3次输入） | 取款失败并退卡 |

多学一招



错误推测法

错误推测法是指测试人员在测试程序的过程中，根据测试经验或直觉推测程序中可能存在的错误，从而有针对性地设计测试用例的方法，该方法通常作为设计测试用例的补充方法。错误推测法不是一个有章可循的方法，其通常做法是测试人员在阅读需求规格说明书时，根据平时测试工作过程中发现的缺陷相关数据和总结猜测可能被忽略的内容。错误推测法能够充分体现测试人员的经验，但是对于经验或测试技能不足的测试人员，不建议使用该方法，可以先使用其他方法（例如等价类划分法、边界值分析法等）设计测试用例。如果其他方法不行，再使用错误推测法。

2.6

状态迁移图法





2.6.1 状态迁移图法概述



掌握状态迁移图法的概述，能够通过状态迁移图法设计测试用例



2.6.1 状态迁移图法概述



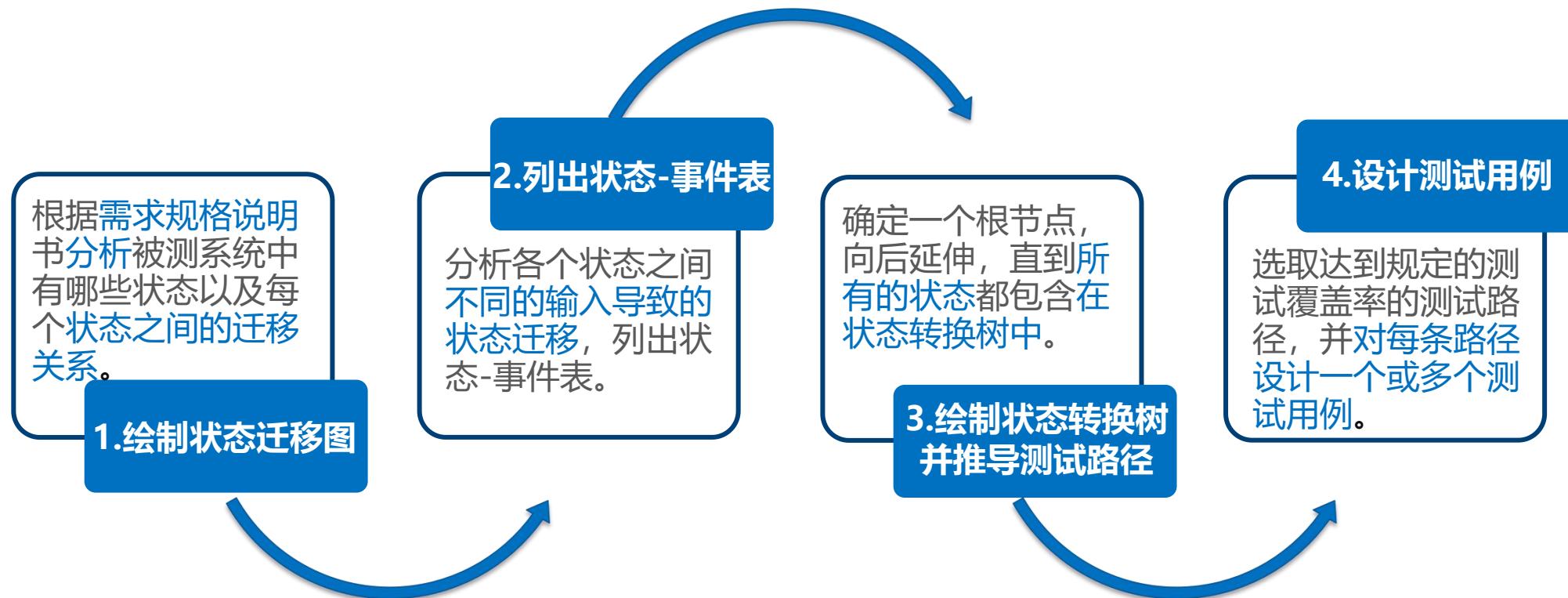
状态迁移图法(State Transition Diagram, STD)是黑盒测试的一种方法，状态迁移图用来描述系统或对象的**状态**，以及导致系统或对象状态发生改变的**事件**。状态迁移图法是通过**分析被测系统的状态，以及这些状态之间的转换条件和路径**来设计测试用例的一种方法，它主要用于验证在给定的条件下，系统对象**是否能够发生状态的改变**，以及**是否存在不可能达到的状态或非法的状态**等。在状态迁移图中，由一个状态、事件所确定的下一个状态可能会有多个，实际迁移到哪一个状态，由触发条件决定。



2.6.1 状态迁移图法概述



状态迁移图法主要关注测试状态转移的正确性，将被测系统中业务流程的每个节点用状态来描述，通过触发的事件来完成各个状态之间的迁移。使用状态迁移图法设计测试用例的具体步骤如下。





2.6.2 实例一：小兔鲜商城订单状态迁移图



掌握小兔鲜商城订单状态迁移图的内容，能够通过状态迁移图法设计小兔鲜商城订单状态迁移的测试用例



2.6.2 实例一：小兔鲜商城订单状态迁移图



以小兔鲜商城项目为例，[使用状态迁移图法讲解小兔鲜商城订单状态的迁移。](#)

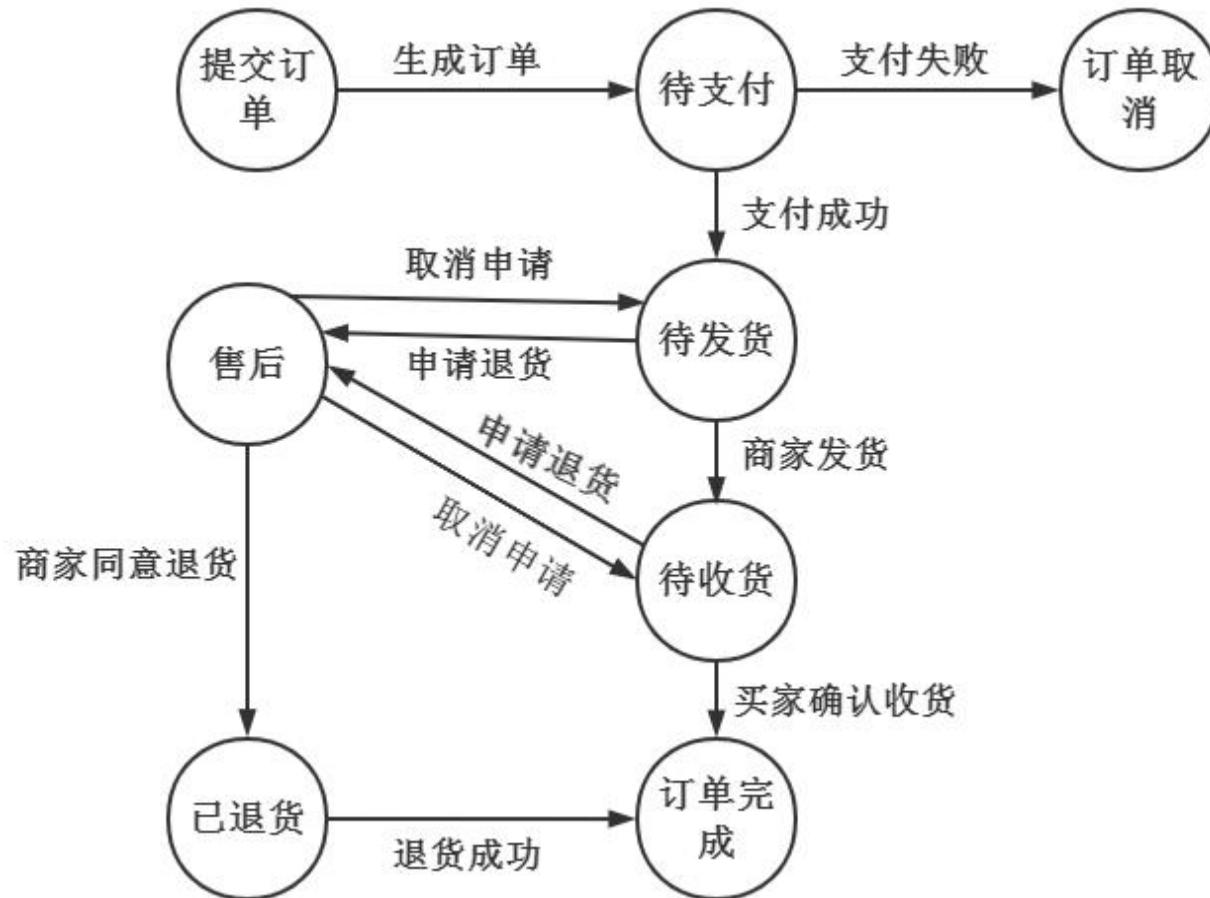
假设小兔鲜商城的需求是：用户在搜索商品后，将商品加入购物车进行购买，用户提交订单后生成订单，订单状态转变为待支付，若支付失败，则订单状态转变为订单取消，若支付成功，则订单状态转变为待发货；商家发货后，订单状态转变为待收货；买家确认收货后，订单状态转变为订单完成；用户可在待发货状态和待收货状态下申请退货或取消申请，若用户申请退货，则订单状态均转变为售后；商家同意退货后，订单状态转变为已退货；退货成功时，订单状态转变为订单完成；如果用户在申请退货后，又取消申请，则订单状态转变为待发货或待收货；商家发货并且买家确认收货后订单状态才转变为订单完成。



2.6.2 实例一：小兔鲜商城订单状态迁移图



根据上述需求描述，可以画出小兔鲜商城订单状态迁移图，如下图所示。





2.6.2 实例一：小兔鲜商城订单状态迁移图



小兔鲜商城订单状态-事件表如下表所示。

| 前一状态 | 事件 | 后一状态 |
|------|------|------|
| 提交订单 | 生成订单 | 待支付 |
| 待支付 | 支付失败 | 订单取消 |
| 待支付 | 支付成功 | 待发货 |
| 待发货 | 申请退货 | 售后 |
| 售后 | 取消申请 | 待发货 |
| 待发货 | 商家发货 | 待收货 |
| 待收货 | 申请退货 | 售后 |



2.6.2 实例一：小兔鲜商城订单状态迁移图



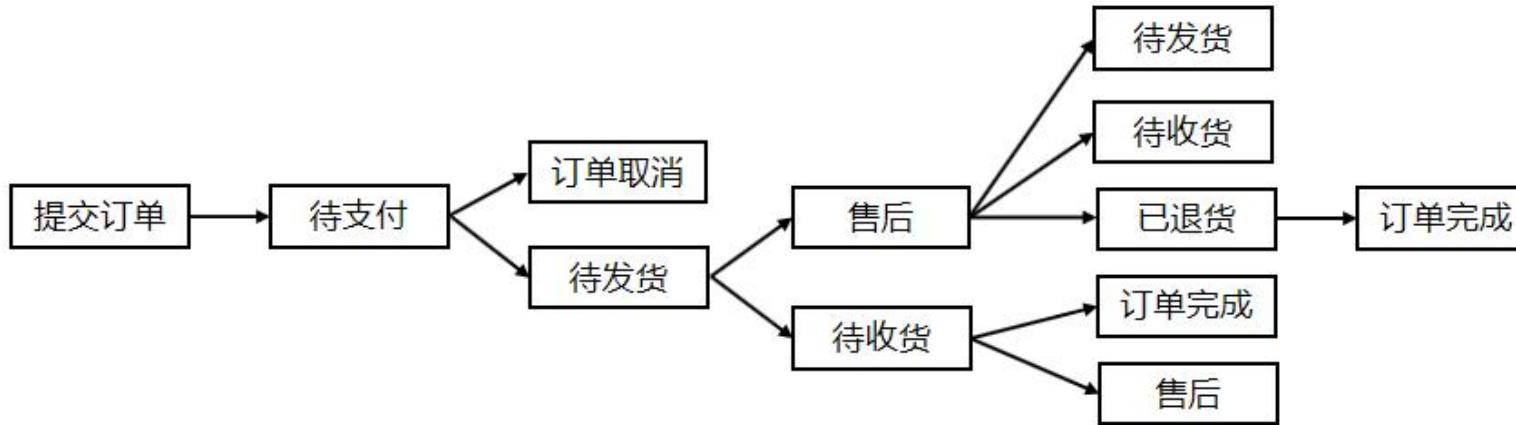
| 前一状态 | 事件 | 后一状态 |
|------|--------|------|
| 售后 | 取消申请 | 待收货 |
| 待收货 | 买家确认收货 | 订单完成 |
| 售后 | 商家同意退货 | 已退货 |
| 已退货 | 退货成功 | 订单完成 |



2.6.2 实例一：小兔鲜商城订单状态迁移图



根据状态迁移画出小兔鲜商城订单状态转换树，如下图所示。



通过分析上图可知，一共有6条测试路径，具体如下。

- 测试路径1：提交订单→待支付→订单取消。
- 测试路径2：提交订单→待支付→待发货→售后→待发货。
- 测试路径3：提交订单→待支付→待发货→售后→待收货。
- 测试路径4：提交订单→待支付→待发货→售后→已退货→订单完成。
- 测试路径5：提交订单→待支付→待发货→待收货→订单完成。
- 测试路径6：提交订单→待支付→待发货→待收货→售后。



2.6.2 实例一：小兔鲜商城订单状态迁移图



小兔鲜商城订单状态迁移的测试用例如下表所示。

| 测试用例 | 测试路径 | 前置条件 | 测试步骤 | 预期结果 |
|-------|------|---------------------------------|---|--------|
| test1 | 1 | 1.成功注册、登录的用户; 2.成功搜索商品并加入购物车 | 1.选择购物车中的商品提交订单; 2.使用余额不足的银行卡支付; 3.选择取消订单 | 成功取消订单 |
| test2 | 2 | 1.成功注册、登录的用户; 2.成功搜索商品并加入购物车 | 1.选择购物车中的商品提交订单; 2.使用可正常支付的银行卡支付; 3.申请退货; 4.商家同意退货; 5.取消申请 | 成功取消申请 |
| test3 | 3 | 1.成功注册、登录的用户; 2.成功搜索商品并加入购物车 | 1.选择购物车中的商品提交订单; 2.使用可正常支付的银行卡支付; 3.商家发货; 4.申请退货; 5.商家同意退货; 6.取消申请 | 成功取消申请 |



2.6.2 实例一：小兔鲜商城订单状态迁移图



| 测试用例 | 测试路径 | 前置条件 | 测试步骤 | 预期结果 |
|-------|------|---------------------------------|--|--------|
| test4 | 4 | 1.成功注册、登录的用户； 2.成功搜索商品并加入购物车 | 1.选择购物车中的商品提交订单； 2.使用可正常支付的银行卡支付； 3.商家发货； 4.申请退货； 5.商家同意退货 | 成功退货 |
| test5 | 5 | 1.成功注册、登录的用户； 2.成功搜索商品并加入购物车 | 1.选择购物车中的商品提交订单； 2.使用可正常支付的银行卡支付； 3.商家发货； 4.买家确认收货 | 成功购物 |
| test6 | 6 | 1.成功注册、登录的用户； 2.成功搜索商品并加入购物车 | 1.选择购物车中的商品提交订单； 2.使用可正常支付的银行卡支付； 3.商家发货； 4.买家申请退货 | 成功申请退货 |



2.6.3 实例二：飞机售票系统状态迁移图



- 掌握飞机售票系统状态迁移图的内容，能够通过状态迁移图法设计用户预定机票到使用机票过程中机票状态迁移的测试用例



2.6.3 实例二：飞机售票系统状态迁移图



下面以飞机售票系统为例，讲解[用户预定机票到使用机票过程中机票状态的迁移情况。](#)

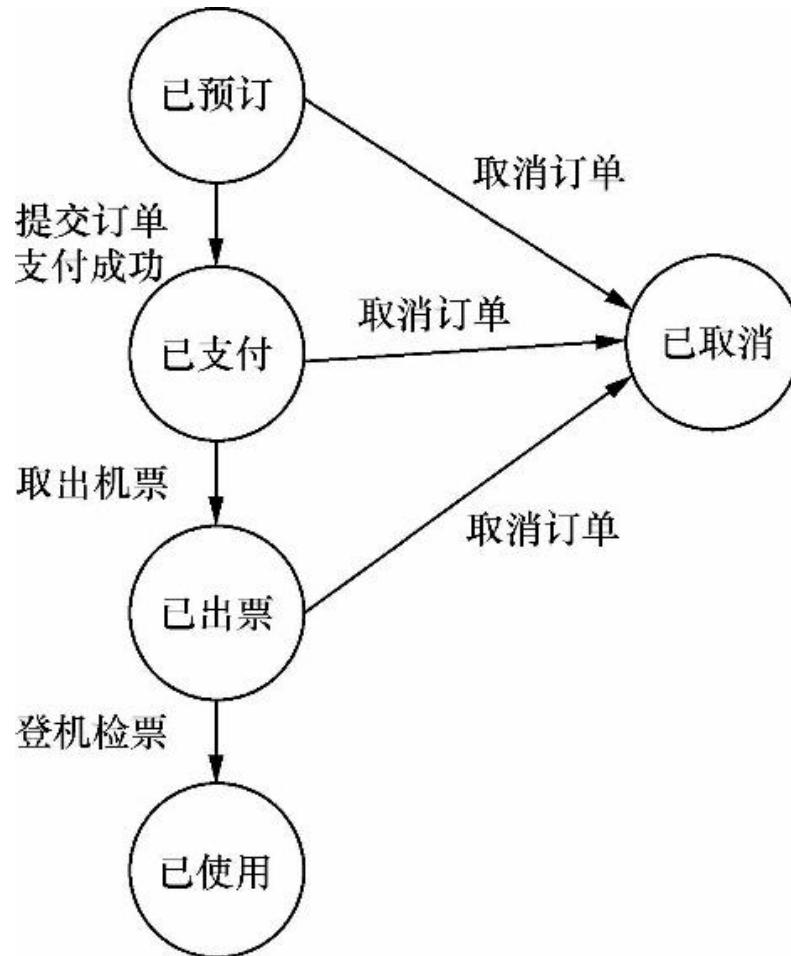
假设飞机售票系统的需求是：乘客可以通过小程序预约购买机票，预约成功时，机票状态为已预订；乘客提交订单并成功支付机票费用后，机票状态为已支付；乘客到机场取出机票后，机票状态为已出票；乘客登机检票后，机票状态为已使用；在登机前，例如在已预订、已支付或已出票的状态下，乘客可以取消订单，在这3种情况下取消订单时，机票状态都为已取消。



2.6.3 实例二：飞机售票系统状态迁移图



现需要根据前面给出的需求测试飞机售票系统中机票状态迁移的过程，首先通过分析需求，画出状态迁移图。机票状态迁移图如下图所示。





2.6.3 实例二：飞机售票系统状态迁移图



机票状态迁移图中一共由5个状态组成，分别是已预订、已支付、已出票、已使用和已取消。根据机票状态迁移图画出机票状态-事件表，如下表所示。

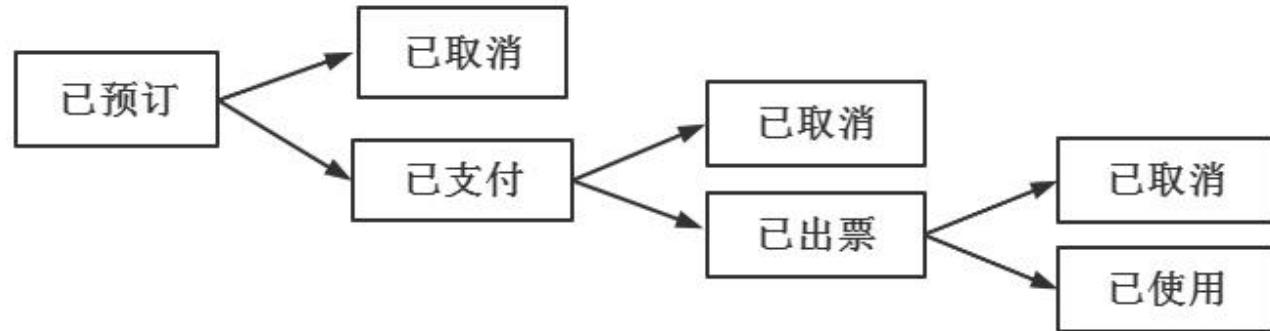
| 前一状态 | 事件 | 后一状态 |
|------|------|------|
| 已预订 | 取消订单 | 已取消 |
| 已预订 | 提交订单 | 已支付 |
| 已支付 | 取消订单 | 已取消 |
| 已支付 | 支付成功 | 已出票 |
| 已出票 | 取消订单 | 已取消 |
| 已出票 | 登机检票 | 已使用 |



2.6.3 实例二：飞机售票系统状态迁移图



为了能够更好地确定测试路径，需要根据状态迁移画出状态转换树，机票状态转换树如下图所示。



通过分析上图可知，一共有[4条测试路径](#)，具体如下所示。

- 测试路径1：已预订→已取消。
- 测试路径2：已预订→已支付→已取消。
- 测试路径3：已预订→已支付→已出票→已取消。
- 测试路径4：已预订→已支付→已出票→已使用。



2.6.3 实例二：飞机售票系统状态迁移图



假设需要测试用户预定机票到使用机票过程中机票状态的迁移过程，根据上一页中的4条测试路径可以设计4条测试用例，机票状态迁移的测试用例如下表所示。

| 测试用例 | 测试路径 | 前置条件 | 测试步骤 | 预期结果 |
|-------|------|---|--|--------|
| test1 | 1 | 1.成功注册、登录的用户； 2.成功设置出发地和目的地； 3.成功选择所需舱位 | 1.预订所选舱位的机票； 2.取消订单 | 成功取消订单 |
| test2 | 2 | 1.成功注册、登录的用户； 2.成功设置出发地和目的地； 3.成功选择所需舱位 | 1.预订所选舱位的机票； 2.支付机票费用； 3.取消订单 | 成功取消订单 |
| test3 | 3 | 1.成功注册、登录的用户； 2.成功设置出发地和目的地； 3.成功选择所需舱位 | 1.预订所选舱位的机票； 2.支付机票费用； 3.取出机票； 4.取消订单 | 成功取消订单 |
| test4 | 4 | 1.成功注册、登录的用户； 2.成功设置出发地和目的地； 3.成功选择所需舱位 | 1.预订所选舱位的机票； 2.支付机票费用； 3.取出机票； 4.检票 | 成功登机 |



2.6.3 实例二：飞机售票系统状态迁移图



多学一招



设计测试用例的方法选择

在使用黑盒测试方法设计测试用例的过程中，如果测试模块具有输入功能，但是输入功能之间没有组合关系，则选择等价类划分法；如果测试模块的功能对输入有边界限制，例如长度范围、数值类型等方面的限制，则选择边界值分析法；如果测试模块具有多输入、多输出、输入与输入之间存在组合关系、输入与输出之间存在依赖或制约关系的情况，则可以选择因果图与决策表法；如果想要用最少的测试用例获得测试模块的最大测试覆盖率，则可以选择正交实验设计法；如果测试模块包含多个功能的组合，则可以选择场景法；如果测试模块在特定条件下会发生状态的改变，则可以选择状态迁移图法。通常，对于测试经验丰富的测试人员来说，还会使用错误推测法来进一步补充测试用例的设计。

本 章 小 结

本章主要讲解了黑盒测试常用的方法，包括等价类划分法、边界值分析法、因果图法与决策表法、正交实验设计法、场景法和状态迁移图法。通过本章的学习，读者能够掌握每种测试方法的原理与测试用例的设计方法，可为后续学习实际软件项目测试奠定基础。

谢谢

