

测试用例设计之等价类划分方法

by:授客 QQ: 1033553122

一. 方法简介

1. 定义

把所有可能的输入数据,即程序的输入域划分成若干部分(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。该方法是一种重要的,常用的黑盒测试用例设计方法。

2. 划分等价类:

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,并合理地假定:测试某等价类的代表值就等于对这一类其它值的测试,因此,可以把全部输入数据合理划分为若干等价类,在每一个等价类中取一个数据作为测试的输入条件就可以用少量代表性的测试数据取得较好的测试结果。

等价类划分可有两种不同的情况:有效等价类和无效等价类。

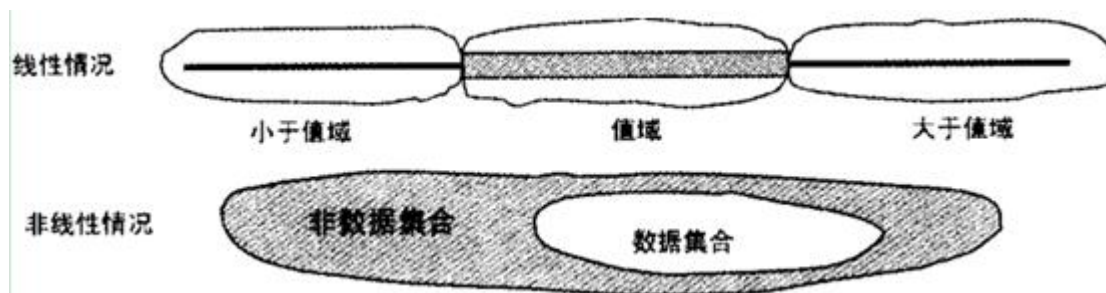
1) 有效等价类

是指对于程序的规格说明来说是合理的、有意义的输入数据构成的集合。利用有效等价类可检验程序是否实现了规格说明中所规定的功能和性能。

2) 无效等价类

与有效等价类的定义恰巧相反。无效等价类指对程序的规格说明是不合理的或无意义的输入数据所构成的集合。对于具体的问题,无效等价类至少应有一个,也可能有多个。

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。



有效等价类和无效等价类的划分

3. 划分等价类的标准:

1) 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;

1.1) 并是整个集合:完备性;

1.2) 子集互不相交:保证一种形式的无冗余性;

1.3) 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”:等价性。

4. 划分等价类的方法

1) 在输入条件规定了取值范围或值的个数的情况下,则可以确立一个有效等价类和两个无效等价类。如:输入值是学生成绩,范围是 $0 \sim 100$;



2) 在输入条件规定了输入值的集合或者规定了“必须如何”的条件的情况下,可确立一个有效等价类和一个无效等价类;

3) 在输入条件是一个布尔量的情况下,可确定一个有效等价类和一个无效等价类。

4) 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。

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。

5) 在规定了输入数据必须遵守的规则的情况下,可确立一个有效等价类(符合规则)和若干个无效等价类(从不同角度违反规则);

6) 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。

5. 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,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。(原因:不同的无效等价类可能相互屏蔽,所以用单一测试用例覆盖无效等价类)

二. 实战演习

1. 某程序规定:“输入三个整数 a 、 b 、 c 分别作为三边的边长构成三角形。通过程序判定所构成的三角形的类型,当此三角形为一般三角形、等腰三角形及等边三角形时,分别作计算...”“。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。)

解答:

方式 1

根据等价分类的定义: 是把所有可能的输入数据, 即程序的输入域划分成若干部分 (子集), 分成有效等价类, 无效等价类. 而有效, 无效的分类是根据题目规定来的。

仔细分析题目:

"输入三个整数 a 、 b 、 c 分别作为三边的边长构成三角形。通过程序判定所构成的三角形的类型, 当此三角形为一般三角形、等腰三角形及等边三角形时, 分别作计算 ... "

有效等价类:

输入三个数 (a, b, c 一个都不能少),

输入整数 (a 为整数, b 为整数, c 为整数),

输入的数构成三角形 ($a > 0, b > 0, c > 0$ && 两边之和大于第三边)

无效等价类: 不满足有效等价类的

根据划分的方法之一: 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。

上述题目中等价类, 输入的数构成三角形, 不同三角形处理不一样, 所以要进一步划分有效等价类为:

输入的数值构成一般三角形, 输入的数值构成等腰三角形, 输入的数值构成等边三角形, 所以, 有效等价类为:

输入三个数 (a, b, c 一个都不能少),

输入整数 (a, b, c 都为整数),

输入的数值构成三角形 ($a > 0, b > 0, c > 0$ && 两边之和大于第三边 -- 子分类 >> 够成一般三角形:

a, b, c 都不相等; 构成等腰三角形: 其中 a, b, c 中仅两个数相等; 构成等边三角形: a, b, c 都相等)

无效等价类:

输入少于三个数 (a, b, c 仅少 1 个, 仅少 2 个);

输入整数 (a, b, c 仅某个不为整数, 仅某 2 个不为整数, 仅 3 个都不为整数);

输入的数值不构成三角形:

1) a, b, c 三个数仅某个数为 0, 仅某两个数为 0, 三个都为 0

2) a, b, c 中仅某个数小于 0, 仅某 2 个数小于 0, 3 个数都为 0

3) 输入三个数: 某两数之和小于第三个数, 某两数之和等于第三个数)

方式二:

按业务流程来, 按等价划分的原则来

输入数据->处理(判断)->输出

一种, 我们按输入进行分类, 这个情况比较复杂, 不好分类

一种, 我们按输出进行分类, 这个情况就比较简单了。所以选择输出入手

输出形状:

{构成三角形, 不构成三角形} → 分成两类, 且并为整个集合

1) 构成三角形--> {一般三角形, 等腰三角形, 等边三角形}

2) 不同三角形判断不一样, 同一等价类中, 出现处理不同, 所以继续分类, 输出形状: 一般三角形, 等腰三角形, 等边三角形, 不构成三角形

有效等价类的要求:

题目显示要求: 整数, 三边,

每边大于 0 (隐性需求)

两边之和大于第三边 (隐性需求): {一般三角: 三边不相等; 等腰三角: 两边相等; 等边三角: 三边相等}

得出最后的有效等价类

整数

存在三边

三边都大于 0

两边之和大于第三边, 且三边不相等 (一般三角形)

两边之和大于第三边, 且仅两边相等 (等腰三角形)

三边相等 (因为三边相等, 所以两边之和必定大于第三边) (等边三角形)

无效等价类的要求 → 根据有效等价类来确定

存在非整数

不满足三边

存在边小于等于 0

两边之和小于等于第三边

得出最后的无效等价类

存在非整数: {一边非整数, 两边非整数, 三边非整数}

不满足三边: {a, b, c 仅少 1 个, 仅少 2 个}

边存在小于 0: {一边小于 0, 两边小于 0, 三边都小于 0}

边存在等于 0: {一边等于 0, 两边等于 0, 三边等于 0}

两边之和小于第三边: {a+b, a+c, b+c}

两边之和等于第三边: {a+b, a+c, b+c}

得出最无效等价类

a 为非整数, b, c 整数

b 为非整数, a, c 整数

c 为非整数, a, b 整数

a&b 为非整数, c 整数

a&c 为非整数, b 整数

c&b 为非整数, a 整数

只给 a

只给 b

只给 c

.....

等价类划分图

编号		有效等价类
1	整数	a为整数 & b为整数 & c为整数
2	存在三边	a, b, c
3	三边都大于0	a>0 & b>0 & c>0
4	一般三角形	a!=b & b!=c & a!=c; a+b>c; a+c>b; a+c>b;
5	等腰三角形	a=b & b!=c; a+b>c; a+c>b; a+c>b;
6		a=c & b!=c; a+b>c; a+c>b; a+c>b;
7		b=c & a!=c; a+b>c; a+c>b; a+c>b;
8	等边三角形	a=b & b=c & a=c;

编号		无效等价类
9	非整数	a不为整数, b & c为整数
10		b不为整数, a & c为整数
11		c不为整数, a & b为整数
12	少于三边(仅给一边)	仅给a
13		仅给b
14		仅给c
15	少于三边(仅给2边)	仅给a, b
16		仅给a, c
17		仅给b, c
18	边小于0(仅1边小于0)	a<0 & b, c>0
19		b<0 & a, c>0
20		c<0 & a, b>0
21	边小于0(仅2边小于0)	a, b<0 & c>0
22		a, c<0 & b>0
23		b, c<0 & a>0
24	边小于0(3边小于0)	a<0 & b<0 & c<0
25	边等于0(仅1边等于0)	a=0 & b, c!=0
26		b=0 & a, c!=0
27		c=0 & a, b!=0
28	边等于0(仅2边等于0)	a, b=0 & c!=0
29		a, c=0 & b!=0
30		b, c=0 & a!=0
31	边等于0(3边等于0)	a=0 & b=0 & c=0
32	两边之和小于第三边	a+b<c
33		a+c<b
34		b+c<a
35	两边之和等于第三边	a+b=c
36		a+c=b
37		b+c=a

用例设计

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) -- (4)

4	4	5	(1) -- (3), (5)
5	4	5	(1) -- (3), (6)
5	4	4	(1) -- (3), (7)
4	4	4	(1) -- (3), (8)

覆盖无效等价类的测试用例:

a	b	c	覆盖编号
0.5	1	1	9
2	0.6	3	10
5	2	0.7	11
5			12
	4		13
		3	14
1	2		15
5		7	16
	6	8	17
-5	3	2	18
5	-8	7	19
.....

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"。

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

2) 设计测试用例,以便覆盖所有的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(用尽可能少的用例尽可能多的覆盖每个有效等价类):

测试数据	期望结果	覆盖的有效等价类
200211	输入有效	①、⑤、⑧

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

等价类划分法细分又可以分为弱健壮等价类，强健壮等价类，见例 3

3.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，并且满足下列条件:

- ① $1 \leq \text{month} \leq 12$
- ② $1 \leq \text{day} \leq 31$
- ③ $1920 \leq \text{year} \leq 2050$

1) 有效等价类为:

M1={月份: $1 \leq \text{月份} \leq 12$ }
D1={日期: $1 \leq \text{日期} \leq 31$ }
Y1={年: $1920 \leq \text{年} \leq 2050$ }

2) 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: "无效输入日期"。其无效等价类为:

M2={月份: 月份<1}
M3={月份: 月份>12}
D2={日期: 日期<1}
D3={日期: 日期>31}
Y2={年: 年<1920}
Y3={年: 年>2050}

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--单缺陷假设;

强--多缺陷假设;

健壮--考虑了无效值

(一)弱健壮等价类测试[包含有效和无效等价类]

用例 ID	月份	日期	年	预期输出
WR1	6	15	1912	1912 年 6 月 16 日
WR2	-1	15	1912	月份不在 1~12 中
WR3	13	15	1912	月份不在 1~12 中
WR4	6	-1	1912	日期不在 1~31 中
WR5	6	32	1912	日期不在 1~31 中
WR6	6	15	1811	年份不在 1812~2012 中
WR7	6	15	2013	年份不在 1812~2012 中

(二)强健壮等价类测试[包含有效和无效等价类]

用例 ID	月份	日期	年	预期输出
SR1	-1	15	1912	月份不在 1~12 中
SR2	6	-1	1912	日期不在 1~31 中
SR3	6	15	1811	年份不在 1812~2012 中
SR4	-1	-1	1912	两个无效一个有效
SR5	6	-1	1811	两个无效一个有效
SR6	-1	15	1811	两个无效一个有效
SR7	-1	-1	1811	三个无效

参考文章: 测试用例设计白皮书_张元礼