SOFTWARE TESTING

苏临之

sulinzhi029@nwu.edu.cn

Software Development Lifecycle

- Four frequently used models
  1. Big-Bang
  2. Code-and-Fix
  3. Waterfall
  4. Spiral

Test-to-Pass & Test-to-Fail

- Two fundamental approaches to testing software
  - Test-to-Pass: Designing and running test cases to assure only what the software can minimally work is called test-to-pass.
  - Test-to-Fail: Designing and running test cases with the sole purpose of breaking the software is called test-to-fail or error-forcing.

Four Basic Techniques

- Static Black-Box Testing
- Dynamic Black-Box Testing
- Static White-Box Testing
- Dynamic White-Box Testing

Four Basic Techniques

- **Static Black-Box Testing**
- Dynamic Black-Box Testing
- Static White-Box Testing
- Dynamic White-Box Testing

Static Black-Box Testing

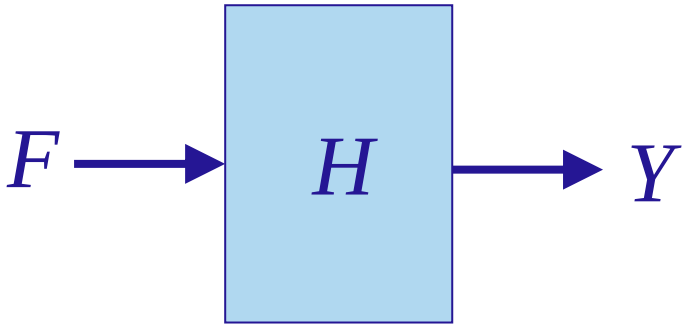
- High-Level Review of the Specification
  - Pretend to be the customer
  - Research existing standards and guidelines
  - Review and test similar software
- Low-Level Review of the Specification
  - Specification attributes
  - Specification terminology

Four Basic Techniques

- Static Black-Box Testing
- **Dynamic Black-Box Testing**
- Static White-Box Testing
- Dynamic White-Box Testing

Dynamic Black-Box Testing

- Suppose the software is a black box, and for each input *F*, the box processes it by using its inner function *H*, and finally we can get an output *Y*.
- So if there is an error in the box (*H*), then *Y* will also be wrong. Thus, we design proper test cases (as *F*) and can check *H* by examining *Y*.



DBBT Techniques

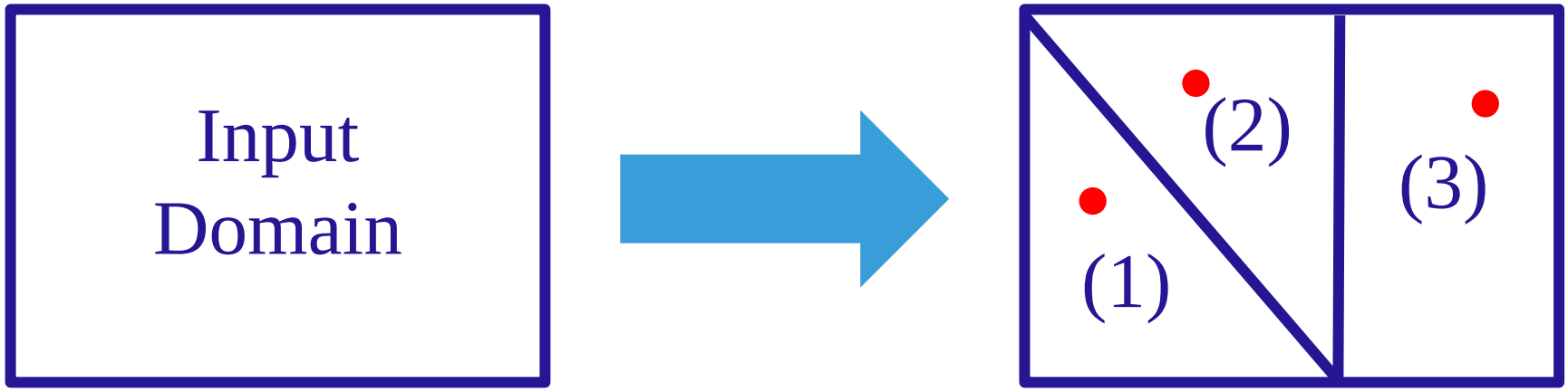
- Equivalence Partitioning
- Boundary Value Analysis
- Decision Table
- Cause-Effect Diagram
- Error Guessing

DBBT Techniques

- **Equivalence Partitioning**
- Boundary Value Analysis
- Decision Table
- Cause-Effect Diagram
- Error Guessing

Equivalence Partitioning

- The input domain is divided into several equivalence classes, and we select only a few (maybe only one) representative test cases from each class, reducing the inputs and thus avoiding redundant full testing. This is called the equivalence partitioning.



Equivalence Partitioning

- The input domain is divided into several equivalence classes, and we select only a few (maybe only one) representative test cases from each class, reducing the inputs and thus avoiding redundant full testing. This is called the equivalence partitioning.
- Two basic steps:
  - Establishing the equivalence class table
  - Designing the corresponding test cases

Example 1

- **Description**
  - 某企业的报表处理系统要求用户输入处理报表的日期，日期限制在2003年1月至2008年12月，即系统只能对该段期间内的报表进行处理，如日期不在此范围内，则显示输入错误信息。输入信息要求：系统日期规定由表征年、月的6位数字字符组成，前4位代表年，后2位代表月。
- **Please design test cases to check the function for date input by using the equivalence partitioning technique.**

Valid Equivalence Class

- In the issue, a legal input consists of 6 digits, in which the former 4 and the latter 2 represent the year from 2003 to 2008 and the month, respectively. Thus, we have the legal input classes as follows.

① 6位数字字符

② 前4位范围2003~2008

③ 后2位范围01~12

- These three legal input classes are called **valid equivalence classes (VECs)**.

Invalid Equivalence Class

- By applying the NOT operation to the legal inputs, we can get the following illegal inputs.

④ 有非数字字符；⑤ 输入少于6位；⑥ 输入多于6位
⑦ 前4位小于2003；⑧ 前4位大于2008
⑨ 后2位小于01；⑩ 后2位大于12

- These seven illegal input classes are called **invalid equivalence classes (IECs)**.

Basic Principles

- Value Range: one VEC & two IECs
- Value Number: one VEC & two IECs
- Multiple Branching: one VEC for each branch value & one IEC
- "Must-Be": one VEC & one IEC
- If the elements are thought to be treated unequally for any reason, or if an EC is too general, then we should divide it into some smaller ECs.

Equivalence Class List

	有效等价类	无效等价类
输入格式	6位数字字符 ①	有非数字字符 ④ 少于6个数字字符 ⑤ 多于6个数字字符 ⑥
前4位范围	2003～2008 ②	小于2003 ⑦ 大于2008 ⑧
后2位范围	01～12 ③	小于01 ⑨ 大于12 ⑩

Legal Test Cases

测试用例	期望结果	覆盖范围
200609	输入有效	①②③

- Here the legal input includes three factors with relationship AND: ①, ② and ③, and hence the corresponding test case should cover these three.

Illegal Test Cases

测试数据	期望结果	覆盖范围
2006ab	输入无效	④（非数字字符）

- Here the illegal input includes only one factor ④, and hence the corresponding test case only need cover this EC with the other factors remaining legal.

Illegal Test Cases

测试数据	期望结果	覆盖范围
2006ab	输入无效	④（非数字字符）
20065	输入无效	⑤（少于6位）
2006005	输入无效	⑥（多于6位）
200105	输入无效	⑦（前4位小于2003）
201005	输入无效	⑧（前4位大于2008）
200600	输入无效	⑨（后2位小于01）
200614	输入无效	⑩（后2位大于12）

Example 2

- **Description**
  - 程序从一个输入对话框中读取3个正整数值。这3个整数值代表了三角形三边的长度。程序显示提示信息，指出该三角形究竟是不等边三角形、等腰三角形还是等边三角形。
- **Please design test cases by using the equivalence partitioning technique.**

Equivalence Class List

	有效等价类	无效等价类
数据性质	数字字符 ①	有非数字字符 ④
数据数量	等于3个 ②	少于3个 ⑤ 多于3个 ⑥
数据格式	正整数 ③	负数 ⑦ 分数 ⑧ 小数 ⑨ 零 ⑩

Equivalence Class List

	有效等价类	无效等价类
数据性质	数字字符 ①	有非数字字符 ④
数据数量	等于3个 ②	少于3个 ⑤ 多于3个 ⑥
数据格式	正整数 ③	零 ⑦

Equivalence Class List

	有效等价类	无效等价类
数据性质	数字字符 ①	有字母字符 ④ 有符号字符 ⑤
数据数量	等于3个 ②	少于3个 ⑥ 多于3个 ⑦
数据格式	正整数 ③	零 ⑧

Example 3

- **Description**

  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”或“1”开头的3位数;
  后　缀：4位数字。

- **The program accepts all the telephone numbers that conform to the terms above and rejects all that does not. Please design test cases by using the equivalence partitioning technique.**

Valid Equivalence Class

地区码 { ① 空白
② 3位数字

前缀 ③ 200~999之间的3位数

后缀 ④ 4位数字

Valid Equivalence Class

地区码 { ① 空白
② **3位数字**

前缀 ③ 200~999之间的3位数

后缀 ④ 4位数字

- "Blank" and "3 numeric characters" are combined with the relation OR, so they are actually two ECs.

Valid Equivalence Class

地区码 { ① 空白
② 3位数字

前缀 ③ **200~999之间的3位数**

后缀 ④ 4位数字

- Suppose "0" = A and "1" = B, then "not 0 or 1" =$\overline{A \cup B}$ = $\overline{A} \cap \overline{B}$, which suggests that the number value ranges from 200 to 999.

Invalid Equivalence Class

地区码：⑤ 有非数字字符；⑥ 多于3位；⑦ 少于3位；

前缀：⑧ 有非数字字符；⑨ 多于3位；⑩ 少于3位；

⑪ 起始位是“0”；⑫ 起始位是“1”；

后缀：⑬ 有非数字字符；⑭ 多于4位；⑮ 少于4位；

Invalid Equivalence Class

地区码：**⑤ 有非数字字符；⑥ 多于3位；⑦ 少于3位；**

前缀：⑧ 有非数字字符；⑨ 多于3位；⑩ 少于3位；

⑪ 起始位是“0”；⑫ 起始位是“1”；

后缀：⑬ 有非数字字符；⑭ 多于4位；⑮ 少于4位；

- "Not blank" indicates the place should be filled with something, and that can be absorbed by ⑤, ⑥ or ⑦.

Invalid Equivalence Class

地区码：⑤ 有非数字字符；⑥ 多于3位；⑦ 少于3位；

前缀：⑧ 有非数字字符；⑨ 多于3位；⑩ 少于3位；

⑪ 起始位是“0”；⑫ 起始位是“1”；

后缀：⑬ 有非数字字符； ⑭ 多于4位； ⑮ 少于4位；

- Suppose "0" = A and "1" = B, then "not 0 or 1" =$\overline{A \cup B}$, and by applying the NOT operation we have $\overline{A \cup B} = A \cup B$, which leads to two IECs.

Equivalence Class List

	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 多于3位 ⑥ 少于3位 ⑦
前缀	200~999的3位数 ③	有非数字字符 ⑧ 多于3位 ⑨ 少于3位 ⑩ 起始位是“0” ⑪ 起始位是“1” ⑫
后缀	4位数字 ④	有非数字字符 ⑬ 多于4位 ⑭ 少于4位 ⑮

Legal Test Cases

测试用例	期望结果	覆盖范围
( ) 456-7890	输入有效	①③④
(029) 345-6789	输入有效	②③④

Illegal Test Cases

测试用例	期望结果	覆盖范围
(29z) 123-4567	输入无效	⑤
(29) 234-5678	输入无效	⑥
(0291) 345-6789	输入无效	⑦
(029) 34z-6789	输入无效	⑧
(029) 34-6789	输入无效	⑨
(029) 2345-6789	输入无效	⑩
(029) 045-6789	输入无效	⑪
(029) 145-6789	输入无效	⑫
(029) 345-678s	输入无效	⑬
(029) 345-678	输入无效	⑭
(029) 345-56789	输入无效	⑮

Additional Principles

- If the condition contains the OR relationship, whether valid or invalid, we usually establish several dependent ECs; provided the relationship is AND, a further consideration is needed.
- "*n* numeric characters" indicates both "numerical digit equals *n*" and "must be numeric characters". By using the NOT operation, three IECs should be established.
- Multiple branches can also be expressed as "must be A, B or C".
- A special requirement for the numerical digit can be translated into the numerical value range.
- Sometimes the De Morgan's laws will facilitate the analysis.

Example 4

- **Description**

  某一种8位计算机，其十六进制的常数定义如下（以下均不区分大小写字母）：
  1、以0x（或0X）为开头字符；
  2、数值字符取值范围是-7f ～ 7f的2位整数。
  符合以上条件的为合法的输入，如0x13、0x6A、0X0e、-0x3c等。

- **Please design test cases by using the equivalence partitioning technique.**

Valid Equivalence Class

开头	以0x或0X开头 ①
数值字符	数字或字母字符a-f ②
数值字符个数	2个 ③
数值本身	-7f ~ 7f ④

Valid Equivalence Class

开头	以0x或0X开头 ①
数值字符	数字或字母字符a-f ②
数值字符个数	2个 ③
数值本身	-7f ~ 7f ④

- The form of letters, whether capital or small, does not have an effect on the test result, so here only one VEC needs establishing.

Valid Equivalence Class

开头	以0x或0X开头 ①
数值字符	**数字或字母字符a-f ②**
数值字符个数	**2个 ③**
数值本身	**-7f ~ 7f ④**

- The second condition is too general to form only one VEC, so it is divided into several aspects, which correspond to four VECs.

Valid Equivalence Class

开头	以0x或0X开头 ①
数值字符	**数字或字母字符a-f ②**
数值字符个数	2个 ③
数值本身	-7f ~ 7f ④

- The numeric and alphabet characters can be mixed together, which is feasible to establish only one VEC even though there exists the OR relationship.

Invalid Equivalence Class

开头	首位不是0 ⑤ 次位不是x或X ⑥
数值字符	含符号字符 ⑦ 含字母字符g-z ⑧
数值字符个数	多于2个 ⑨ 少于2个 ⑩
数值本身	小于-7f ⑪ 大于7f ⑫

Invalid Equivalence Class

开头	**首位不是0 ⑤** **次位不是x或X ⑥**
数值字符	含符号字符 ⑦ 含字母字符g-z ⑧
数值字符个数	多于2个 ⑨ 少于2个 ⑩
数值本身	小于-7f ⑪ 大于7f ⑫

- Please consider the reason why.

Equivalence Class List

开头	以0x或0X开头 ①	首位不是0 ⑤ 次位不是x或X ⑥
数值字符	数字或字母字符a-f ②	含符号字符 ⑦ 含字母字符g-z ⑧
数值字符个数	2个 ③	多于2个 ⑨ 少于2个 ⑩
数值本身	-7f ~ 7f ④	小于-7f ⑪ 大于7f ⑫

Legal Test Cases

测试用例	期望结果	覆盖范围
0x3d	定义正确	①②③④

Illegal Test Cases

测试用例	期望结果	覆盖范围
1x3d	定义错误	⑤
1z3d	定义错误	⑥
0x*2	定义错误	⑦
0x3k	定义错误	⑧
0x3d1	定义错误	⑨
0x3	定义错误	⑩
-0x9d	定义错误	⑪
0x9d	定义错误	⑫

Limitations

- Equivalence partitioning is based on the hypothesis that all the elements in the class are in equal status and that one element can represent the class. But this is not always true.
- For instance, if the inequality $1 \leq x \leq 5$ is mistaken as $1 < x < 5$, and the legal input to the VEC is selected as 4 while the illegal ones to the IECs are selected as 0 and 6, then the mistake (probably vital and fatal) will not be detected.

DBBT Techniques

- Equivalence Partitioning
- **Boundary Value Analysis**
- Decision Table
- Cause-Effect Diagram
- Error Guessing

Boundary Value Analysis

- It is considered that the boundary values along with the adjacent values are more sensitive to those which are far away from boundaries. So they should be paid more attention to and therefore selected as test cases so as to find more hidden bugs.
- Boundary conditions consist of general boundary conditions, sub-boundary conditions, destructive boundary conditions and so on. It is necessary to recognize these conditions according to the specific issue.

Example 5

- **Description**
  - 某企业的报表处理系统要求用户输入处理报表的日期，日期限制在2003年1月至2008年12月，即系统只能对该段期间内的报表进行处理，如日期不在此范围内，则显示输入错误信息。输入信息要求：系统日期规定由表征年、月的6位数字字符组成，前4位代表年，后2位代表月。
- **Please consider the boundary conditions as well as the corresponding values. Then give out the test cases.**

Boundary Value Selection

- 本例中，前4位的有效等价类子集是2003~2008，无效等价类子集是0000~2002的数集和2009~9999的数集。因此我们可以找出这3个等价类的4个边界值：2002、2003、2008、2009（亦可以加上0000和9999）。同样后2位能找到00、01、12、13这4个边界值（亦可加上99）。
- These values are more sensitive than the other values because one is likely to err when it comes to the inequality programming.

Test Cases for BVA

	测试用例说明	测试数据	期望结果	选取理由
输入格式	1个数字字符	5	显示出错	仅有1个合法字符
	6个数字字符	200605	输入有效	类型及长度均有效
	5个数字字符	20065	显示出错	比有效长度少1
	7个数字字符	2006005	显示出错	比有效长度多1
	有1个非数字字符	2006.5	显示出错	只有1个非法字符
	全是非数字字符	May---	显示出错	6个非法字符
前4位	年份为2003年	200305	输入有效	最小年份
	年份为2008年	200805	输入有效	最大年份
	年份为2002年	200205	显示出错	刚好小于最小年份
	年份位2009年	200905	显示出错	刚好大于最大年份
后2位	月份为01	200601	输入有效	最小月份
	月份为12	200612	输入有效	最大月份
	月份为00	200600	显示出错	刚好小于最小月份
	月份为13	200613	显示出错	刚好大于最大月份

Example 6

- **Description**

  Here is a function of two variables $F=(x+2)\times y$ where x and y are two integers. $x \in [10, 20]$ and $y \in [20,30]$. The software has guaranteed the integral inputs.

- **Please design test cases by using boundary value analysis. First give out the graphical analysis and then show the test case list.**

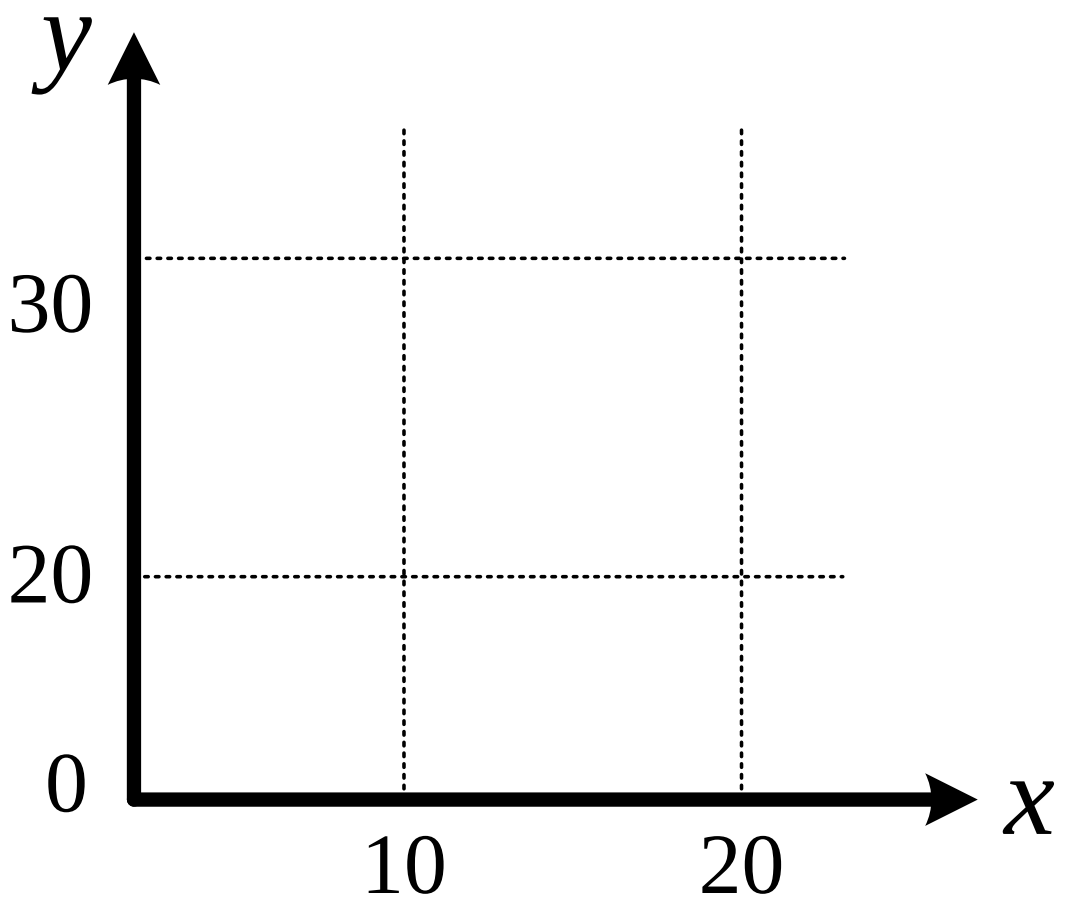
Equivalence Class List

- The EC list is as follows.

	有效等价类	无效等价类
x	10至20的整数 ①	小于10的整数 ③ 大于20的整数 ④
y	20至30的整数 ②	小于20的整数 ⑤ 大于30的整数 ⑥

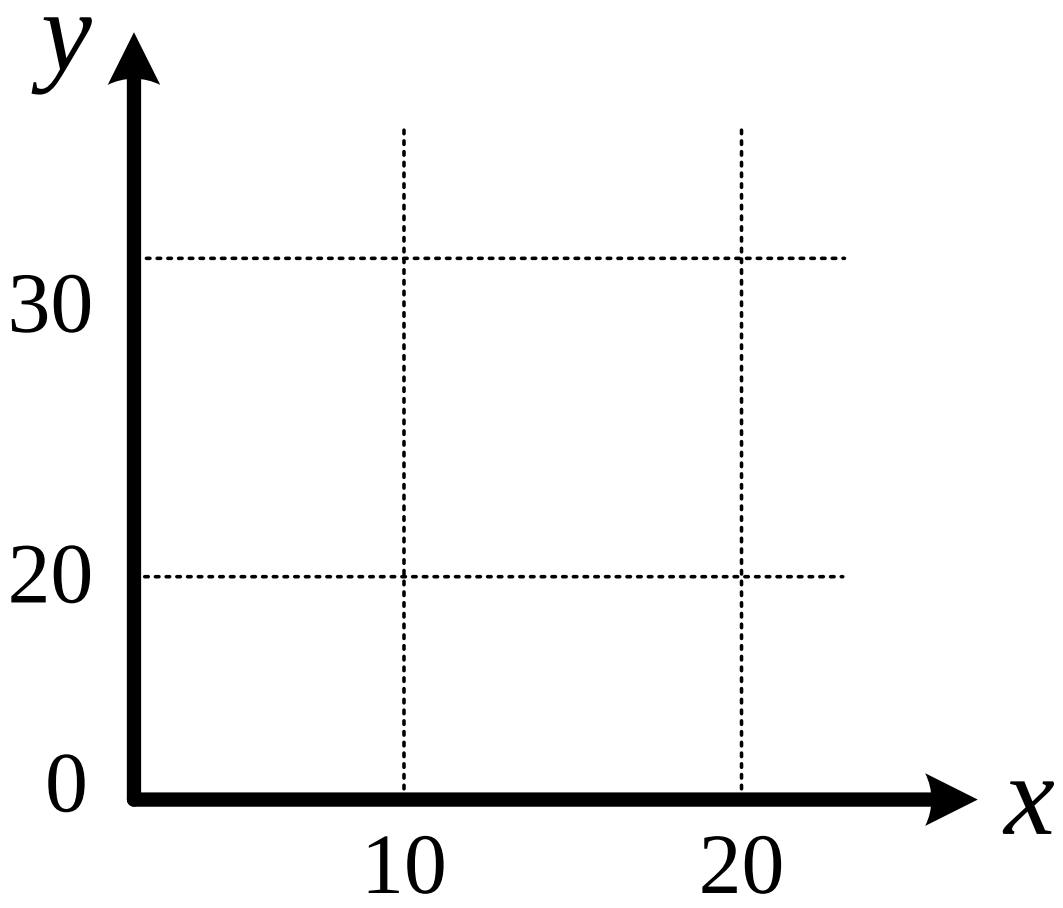
Graphic Analysis

- In the graph, the middle rectangle area is the VEC domain, whereas the other parts are the IEC domain.



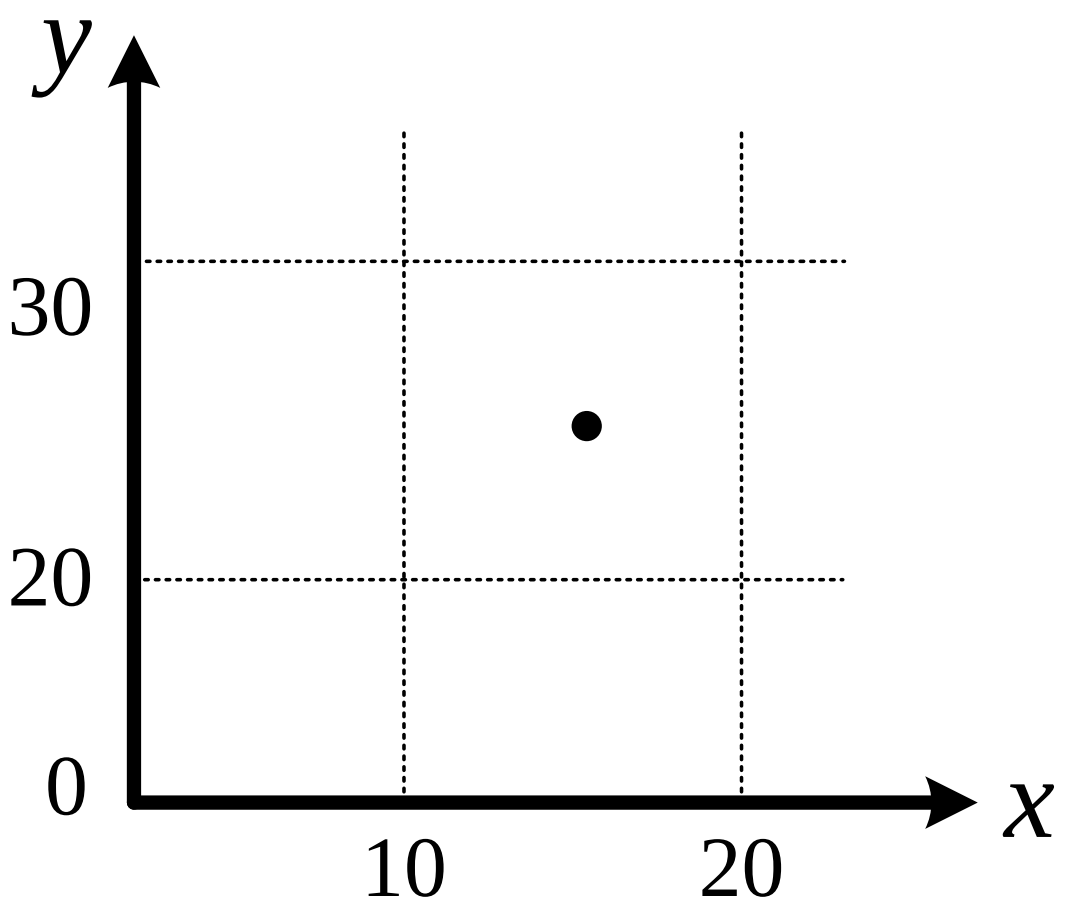
Graphic Analysis

- To check whether there is any boundary problem for only one variable, the other variable should keep valid. So first we should confirm the validation of the central value in the VEC.



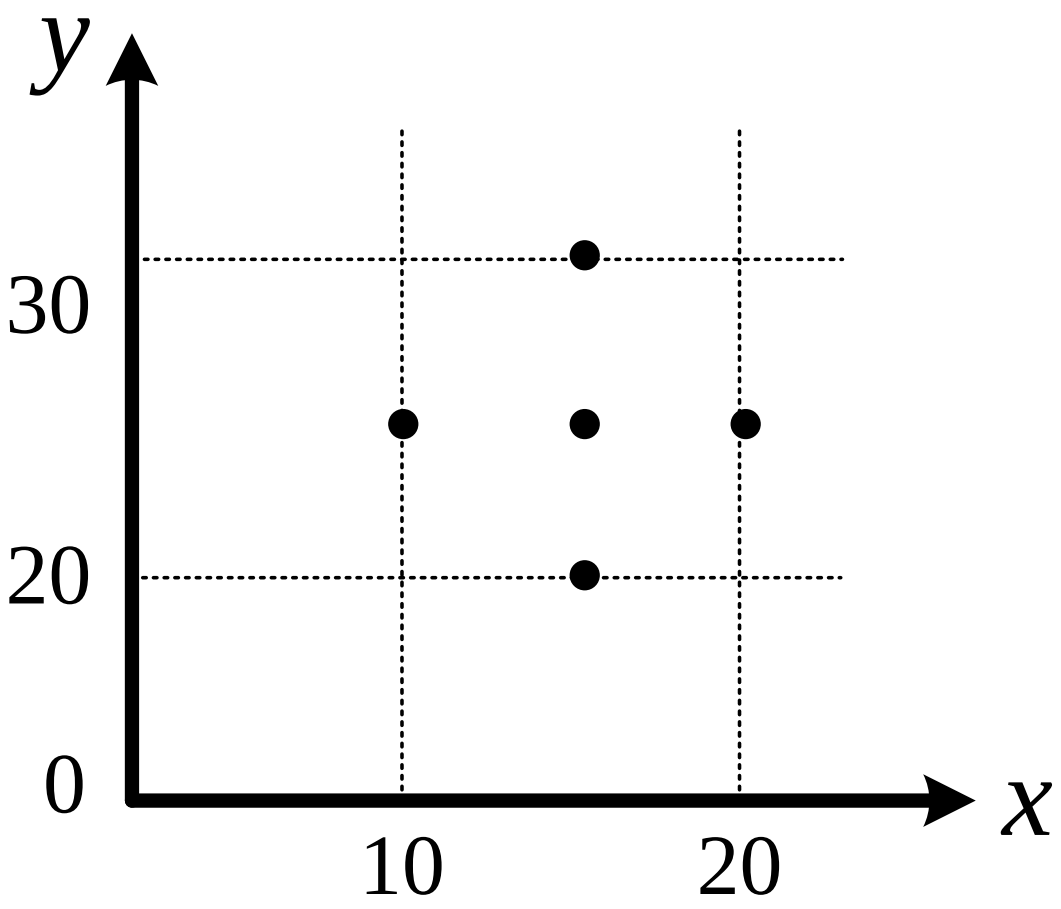
Graphic Analysis

- To check whether there is any boundary problem for only one variable, the other variable should keep valid. So first we should confirm the validation of the central value in the VEC.



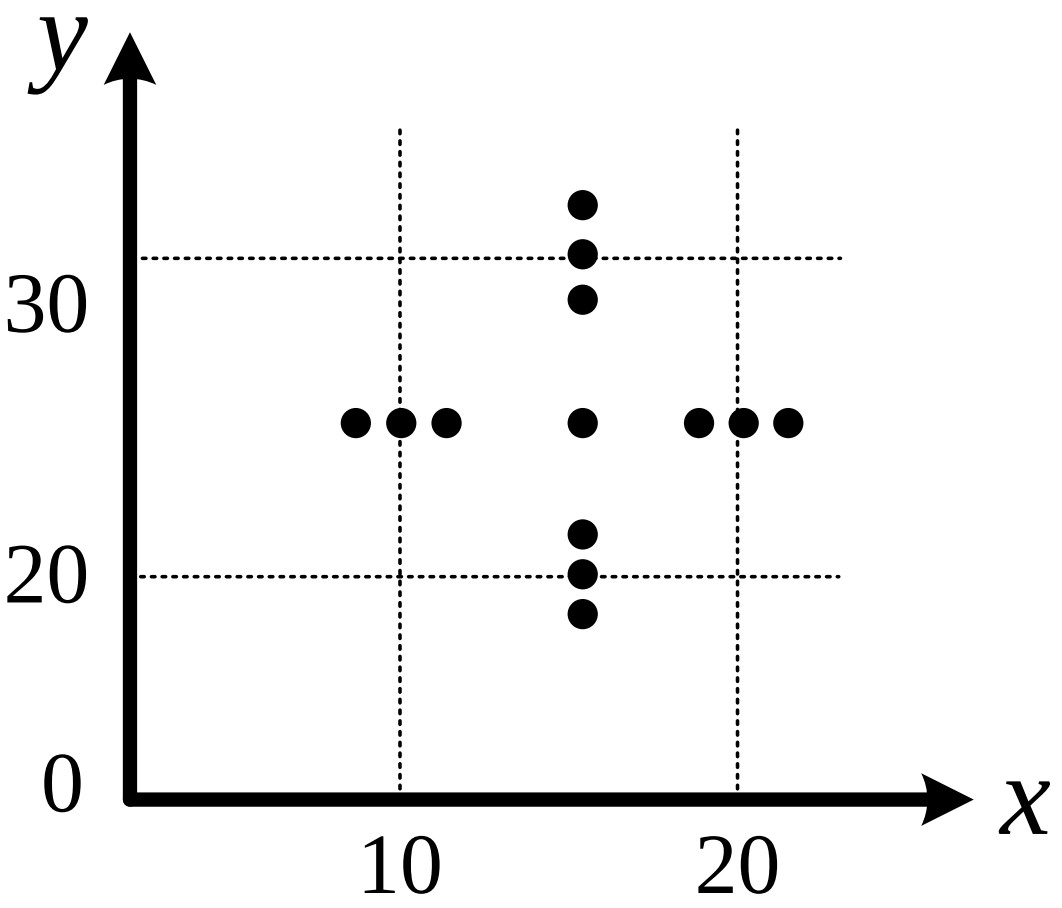
Graphic Analysis

- Then the four boundaries are tested by locating one variable on them. Note that the other variable should keep the valid (unchanged).



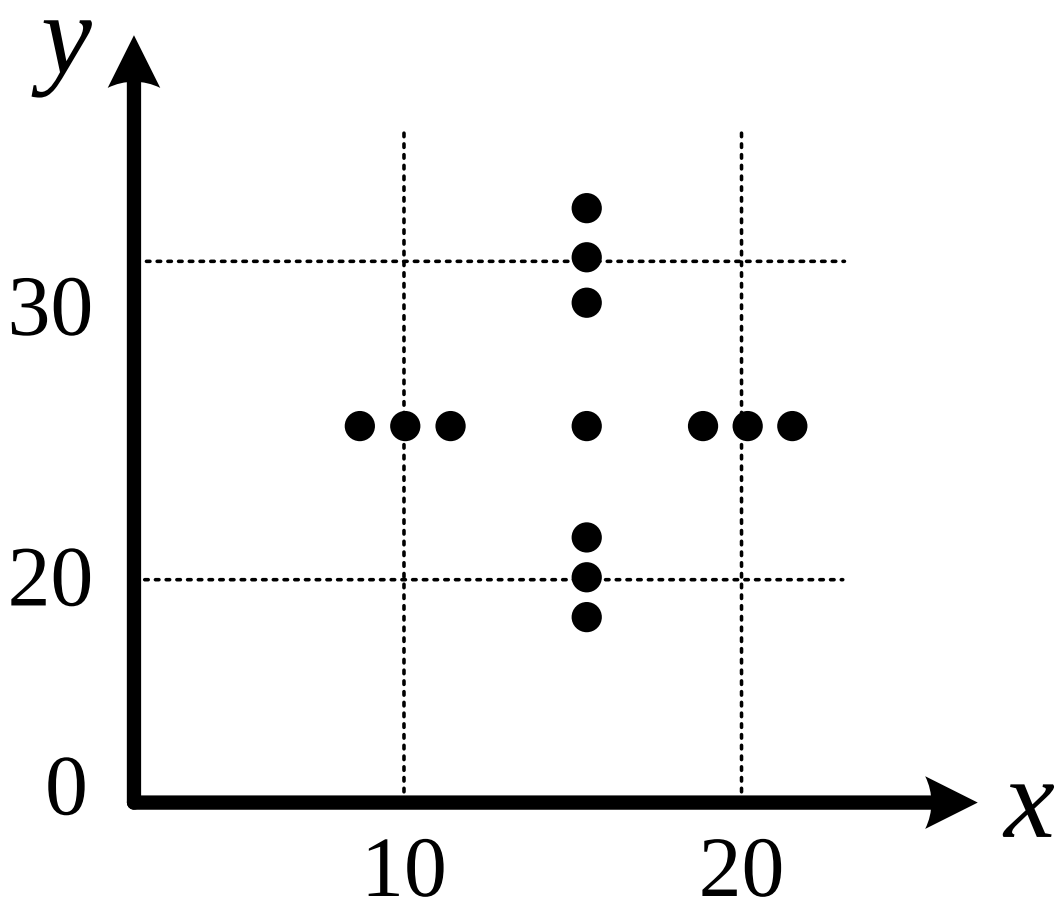
Graphic Analysis

- Finally, the adjacent values (i.e. the boundary value plus one and minus one) are tested, and that is the final graphic analysis. According to the analysis, 13 test cases can be designed.



Graphic Analysis

序号	*x*	*y*	*F*
①	15	25	425
②	10	25	300
③	11	25	325
④	19	25	525
⑤	20	25	550
⑥	15	20	340
⑦	15	21	357
⑧	15	29	493
⑨	15	30	510
⑩	9	25	×
⑪	15	31	×
⑫	21	25	×
⑬	15	19	×



Basic Principles

1. 输入条件规定输入值范围：针对范围的边界设计测试用例，针对刚刚越界的情况设计无效测试用例。
2. 输入条件规定了输入值的数量：针对最小数量、最大数量、最小数量少1、最大数量多1的情况来设计测试用例（此处的“1”指的是最小的精度）。
3. 如果程序的输入或输出是一个有序序列，则应该特别注意序列的第1个和最后1个元素。
4. 边界值分析是具有创造性的方法，发挥聪明才智找出其他的边界（临界）条件。

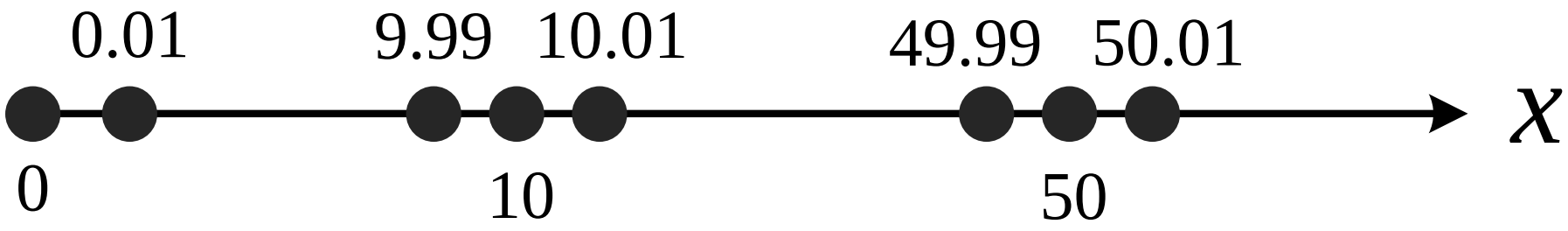
Example 7

- **Description**

  设邮寄某大号包裹时，其质量$0 \le m \le 10$时对应计算公式为$F(m)$，$10 < m \le 50$时对应计算公式为$G(m)$，$m > 50$时对应计算公式为$H(m)$（单位均为kg，精确到0.01kg）。

- **Please design test cases by using boundary value analysis. First give out the graphical analysis and then show the test case list.**

图示分析说明



Limitations

- Boundary value analysis is the complement of the equivalence partitioning. But when it comes to the case where variables are limited by causalities or logical relationships (say $x + y \leq 50$), it will not be easy to use those techniques.
- To solve the problem, the decision table techniques can be employed. To facilitate generating some complicated decision tables, the cause-effect diagram will also be adopted.

Homework-1

- **Description**

  A café called Grace-Berry issues membership cards to attract guests, and the card number involves three parts. The card number must begin with the letters "GB" which stands for the name of the café. The middle part stands for the year one has applied for membership and consists of 4 numerical digits ranging from 2000 to 2022. The last 4 characters should be any numerical digits (i.e. 0000 ~ 9999). For example, a valid card number can be "GB20215678". The corresponding query system will accept the valid numbers and reject the invalid numbers.

- **Please establish the equivalence classes for the issue and the design the corresponding test cases.**

Homework-2

- **Description**

  Here is a function of three variables $F=x+y+z$ where x, y and z are three integers. $x \in [1900, 2100]$, $y \in [1,12]$ and $z \in [1,31]$. The software has guaranteed the integral inputs.

- **Please design test cases by using boundary value analysis. Please list the test cases in a table.**

THANK YOU!