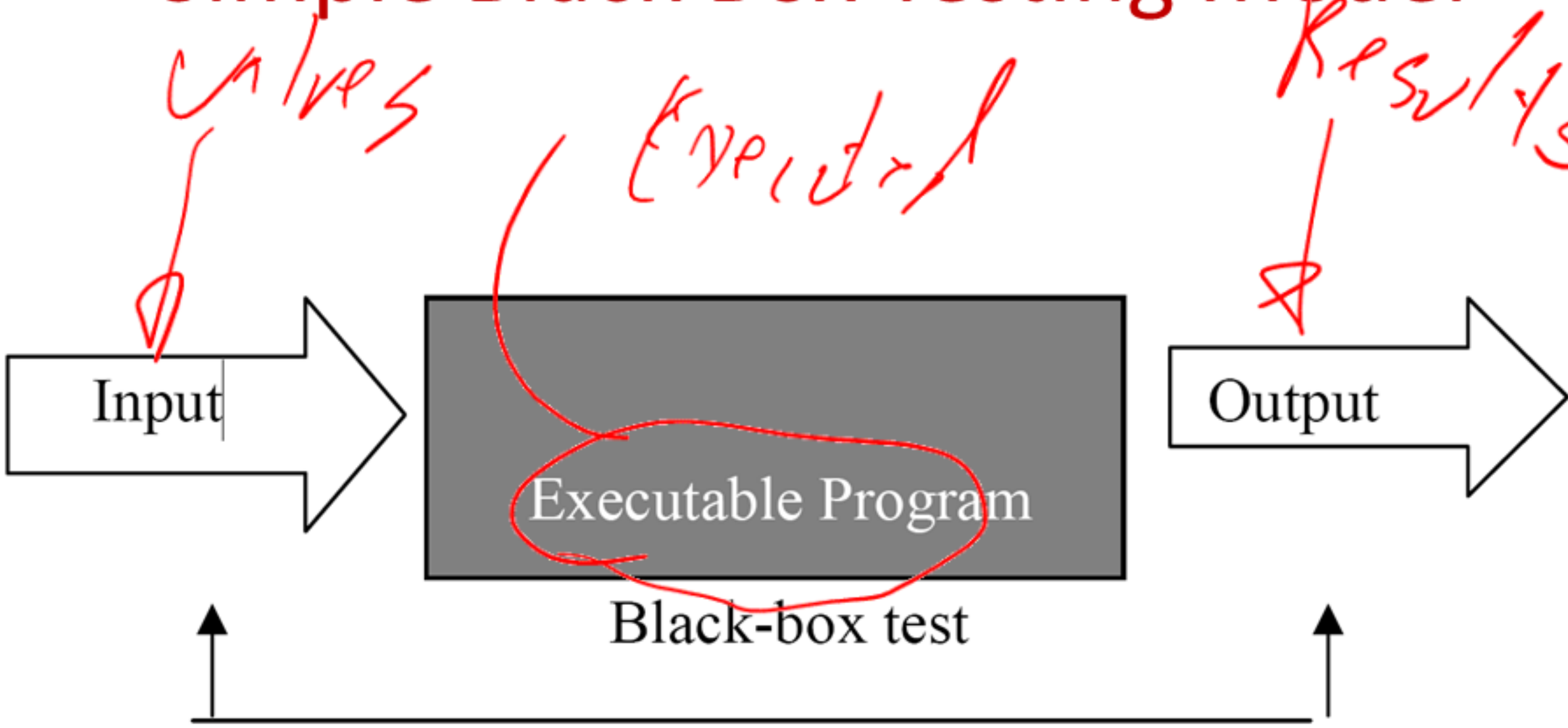
Boundary Value Testing

Lecture Objectives:

1) Define a Software Boundary condition
2) Explain why boundary value conditions represent commonly occurring mistakes
3) Given a software description, construct test cases using the boundary value testing method
4) Compare and contrast boundary value testing with equivalence class testing
5) Based on boundary value testing, determine the minimum number of tests required to test a given system
6) Construct test cases combining equivalence classes and boundary value testing to test multi-variable problems.

Simple Black Box Testing Model



Boundary Conditions

$$y = x^3 + 6x^2 - 50x + 3$$

⇒ intercepts

⇒ Power is 3.

⇒ y given and x

⇒ Zeros?

⇒ Local maxs mins

$y' = 3x^2 + 12x - 50 = 0$

to find local max/min

solve for x

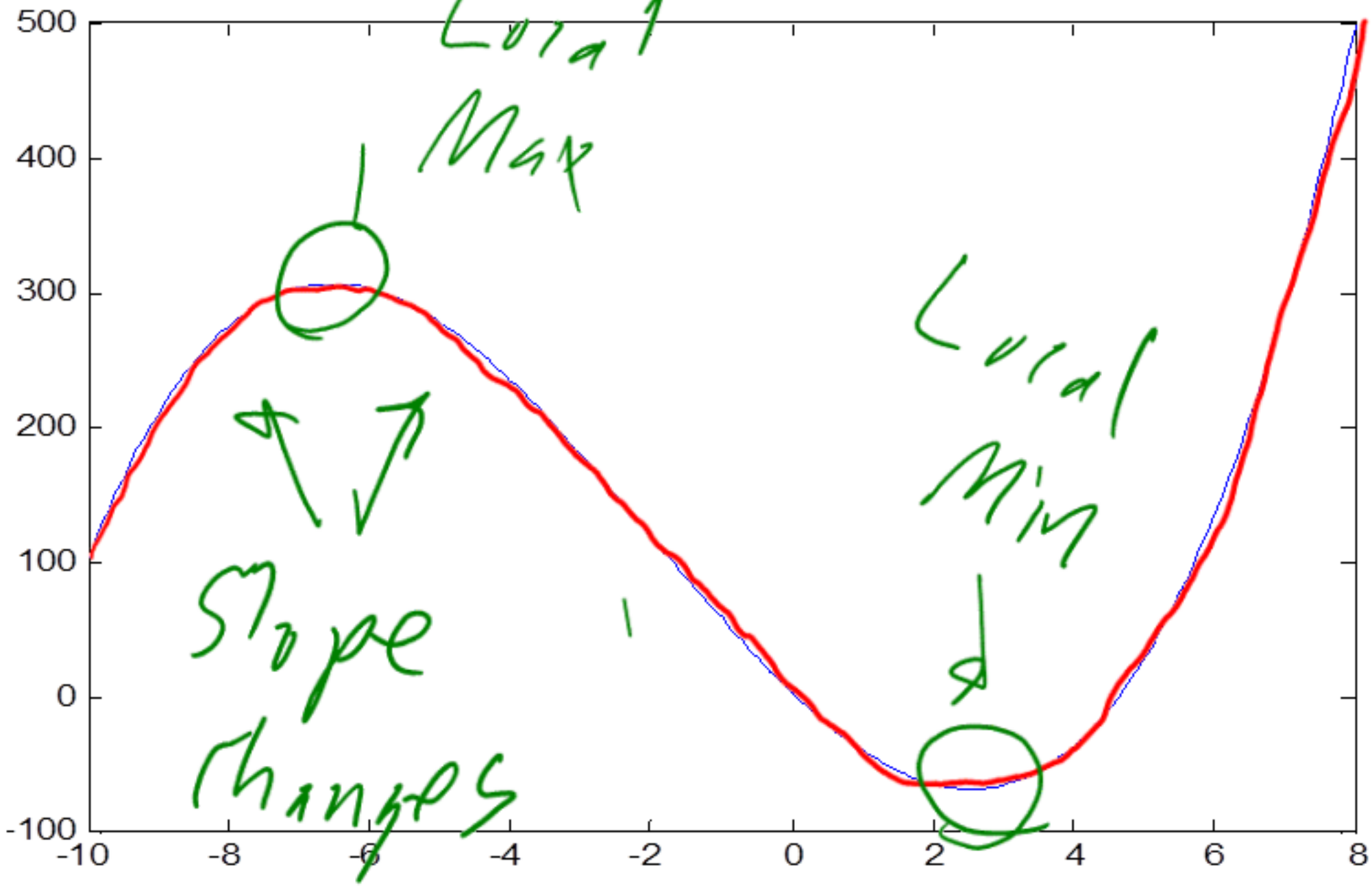
zeros.

$x = 2.546, -6.546$

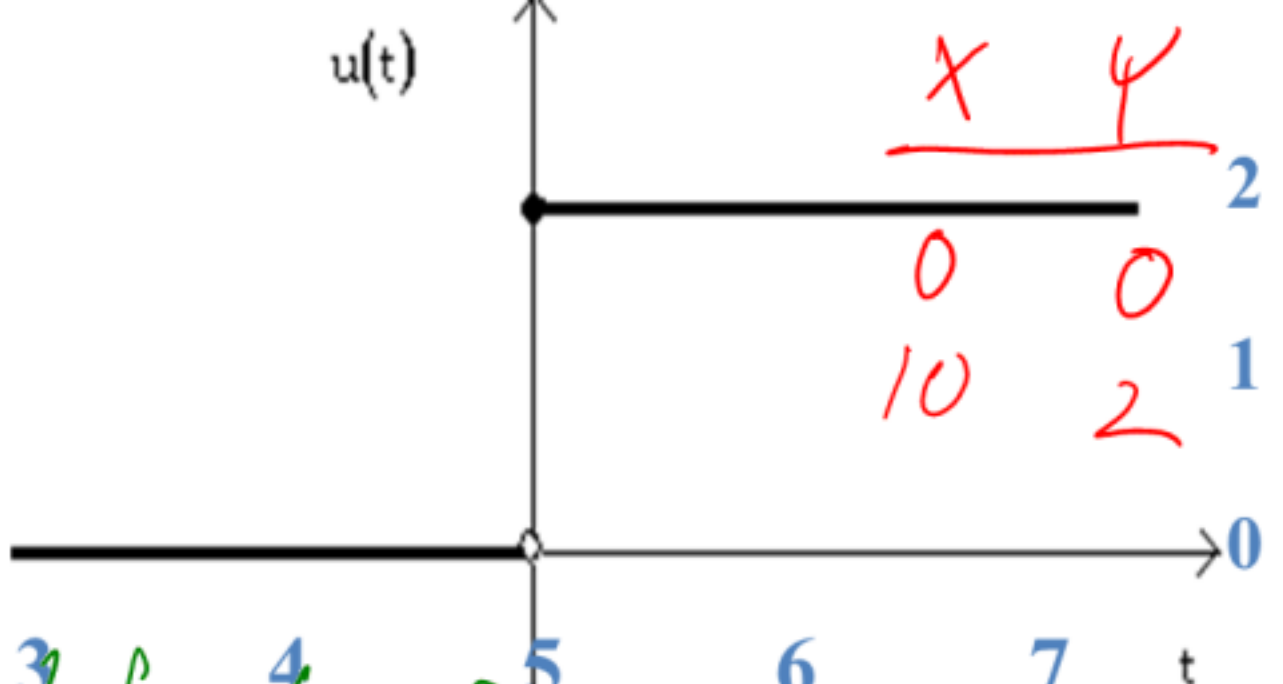
Boundary Conditions

$$y = x^3 + 6x^2 - 50x + 3$$

$$y' = 3x^2 + 12x - 50$$



Boundary Conditions




```
int funct(int x)
{ if (x<5) return 0;
  else return 2;
}
```

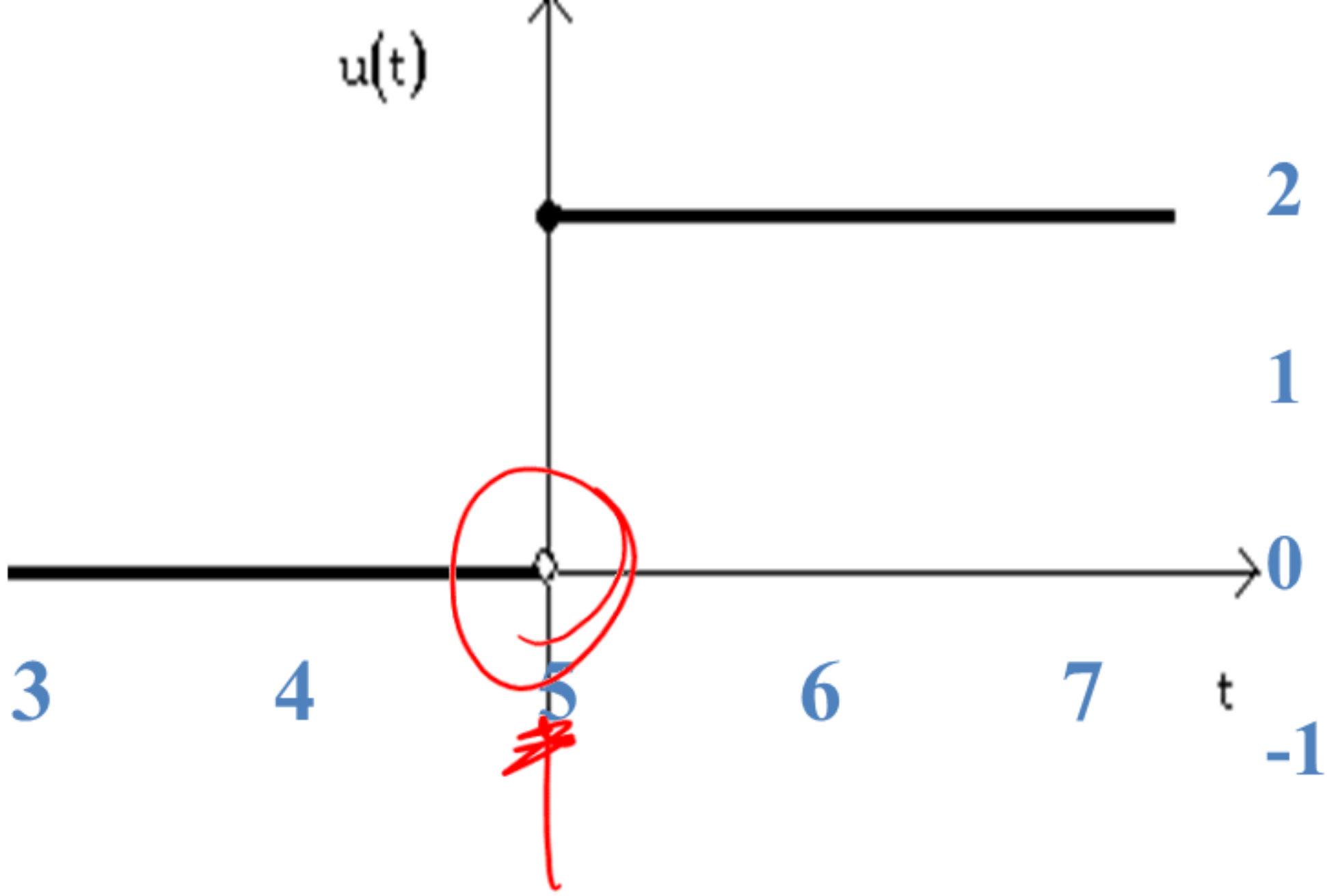
x y
0 0
10 2

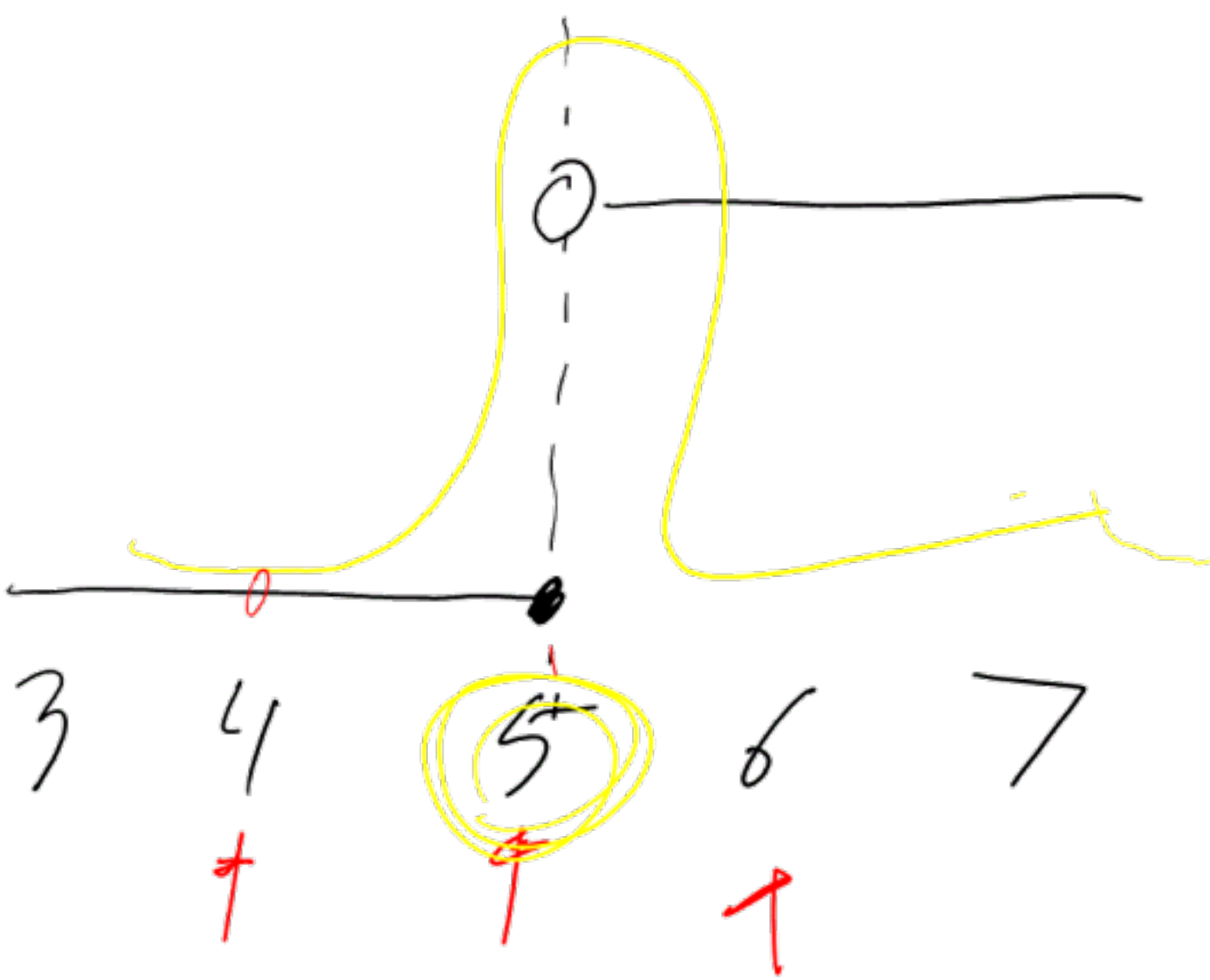
Boundary value analysis

Right at 4, 5, 6 & Right above

Right below

Boundary Conditions

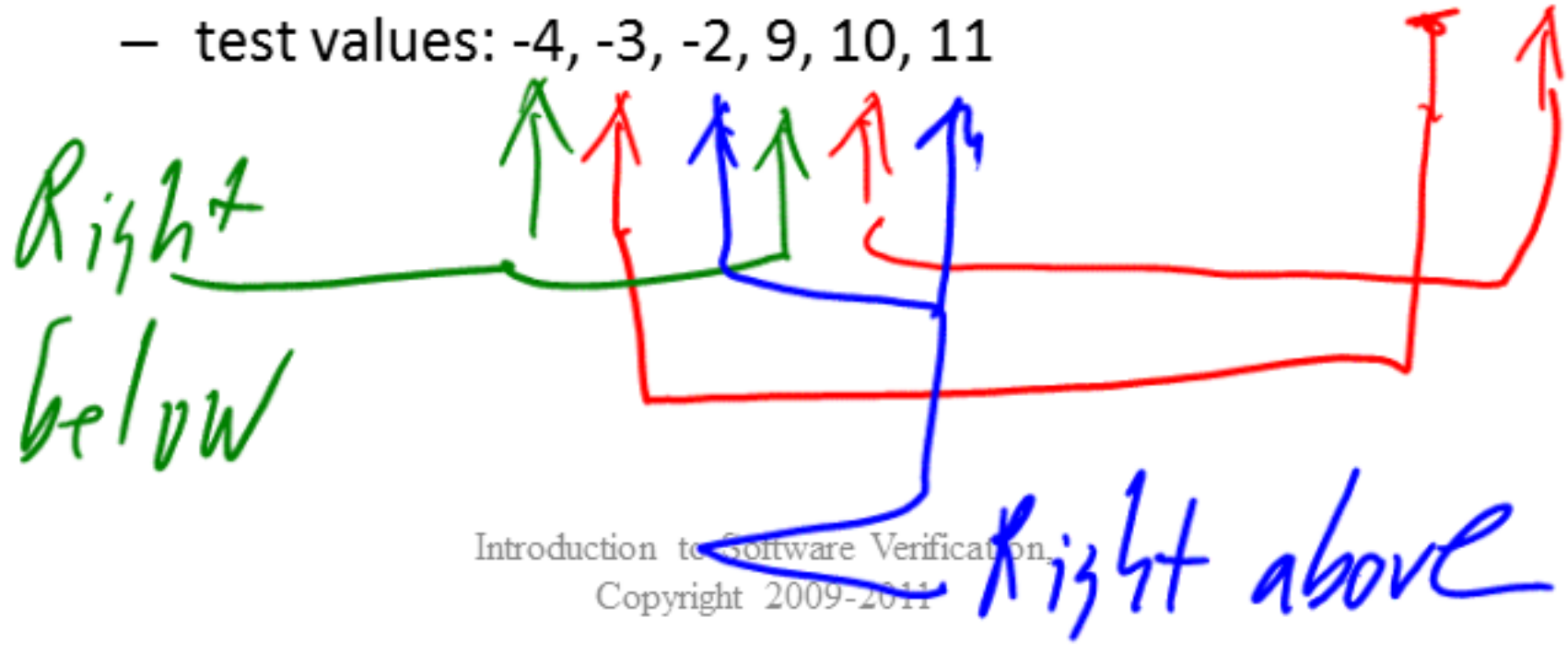





```
if( x != 5 )
{ y = 2x;
```

Boundary Value Analysis

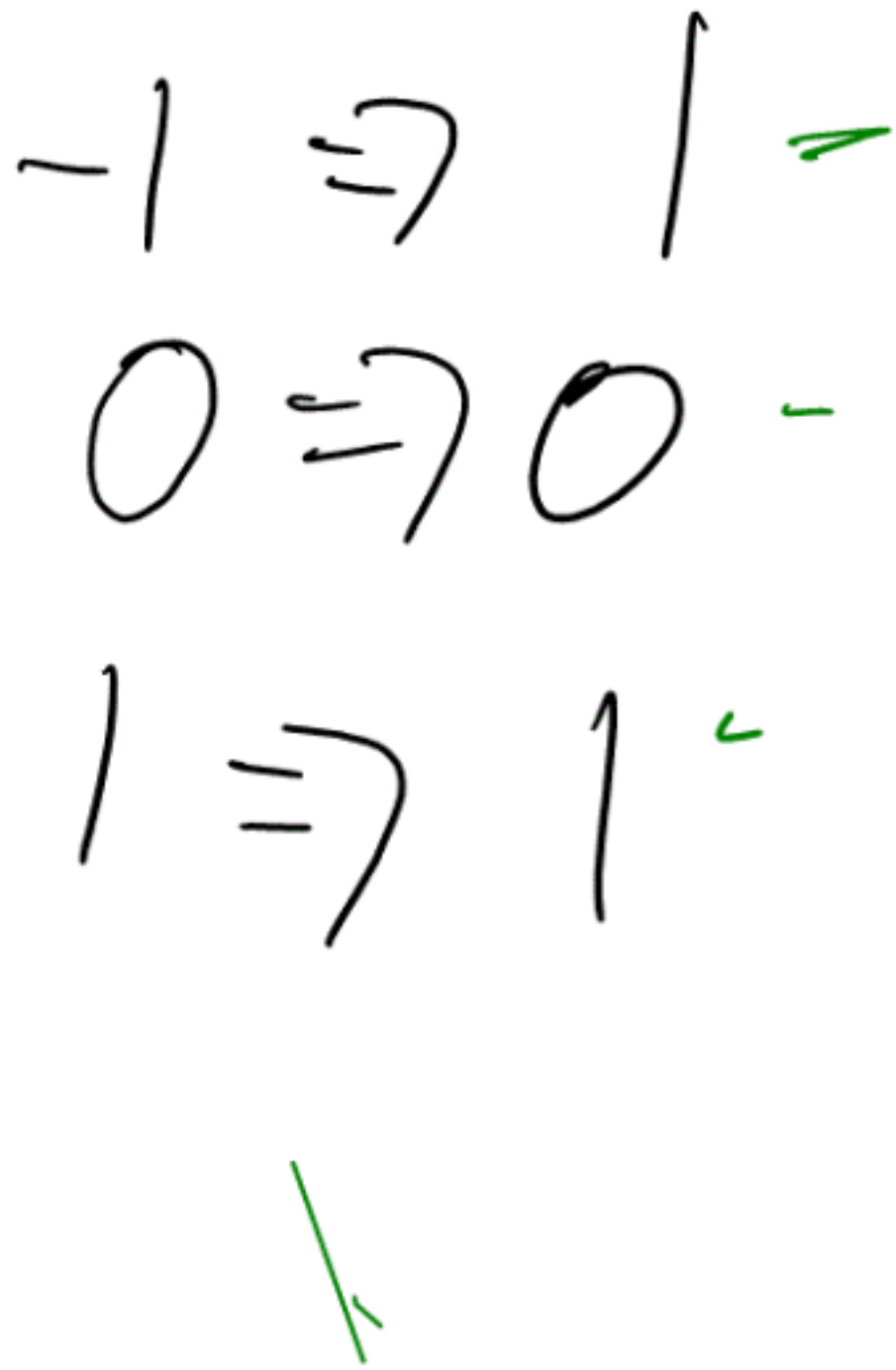
- Boundary value analysis(BVA) - a test case design technique
  - complements to equivalence partition
- Objective:
  - Boundary value analysis leads to a selection of test cases that exercise bounding values.
- Guidelines:
  - If an input condition specifies a range bounded by values a and b, test cases should be designed with value a and b, just above and below a and b.
- Example: Integer D with input condition [-3, 10]
  - test values: -4, -3, -2, 9, 10, 11



Class Example

Equivalence

- Math.absoluteValue(int val)



Equivalence Class Example

- Math.absoluteValue(int val)

{

val < 1

if (val < 0)

return 0-val;

else return val;

}

Boundary Value Analysis

- Think of a store which offers bulk discounts on a purchase.
- Number of units Price per unit
  - First 10 Units $5.00
  - Next 10 Units $4.75
  - Next 10 units $4.50
  - More than 30 units at a time $4.00
- Lets write test cases for a routine
  - Double calculateTotalPrice(int quantity)

quantity	total price
-1	?
0	0
1	5
9	45
10	50
11	54.75
19	...
20	...
21	

29
30 ...
31

```
int price(int quantity)
{
    int retValue = 45 +
        (quantity - 10) * 4.75;
}
```

Why is boundary value testing effective?

- <=, <, >, and >= are commonly problematic in implementation
  - Off by one error
- Loops are controlled by boundary conditions
- Requirements may not be understood when code is implemented

How Many Test Cases are Necessary

Boundaries	Equivalence Classes	Boundary Value Analysis
1	2	3
2	3	6
3	4	9
4	5	12

X X X X X

1 Bunking

Eg - 2 + Cs

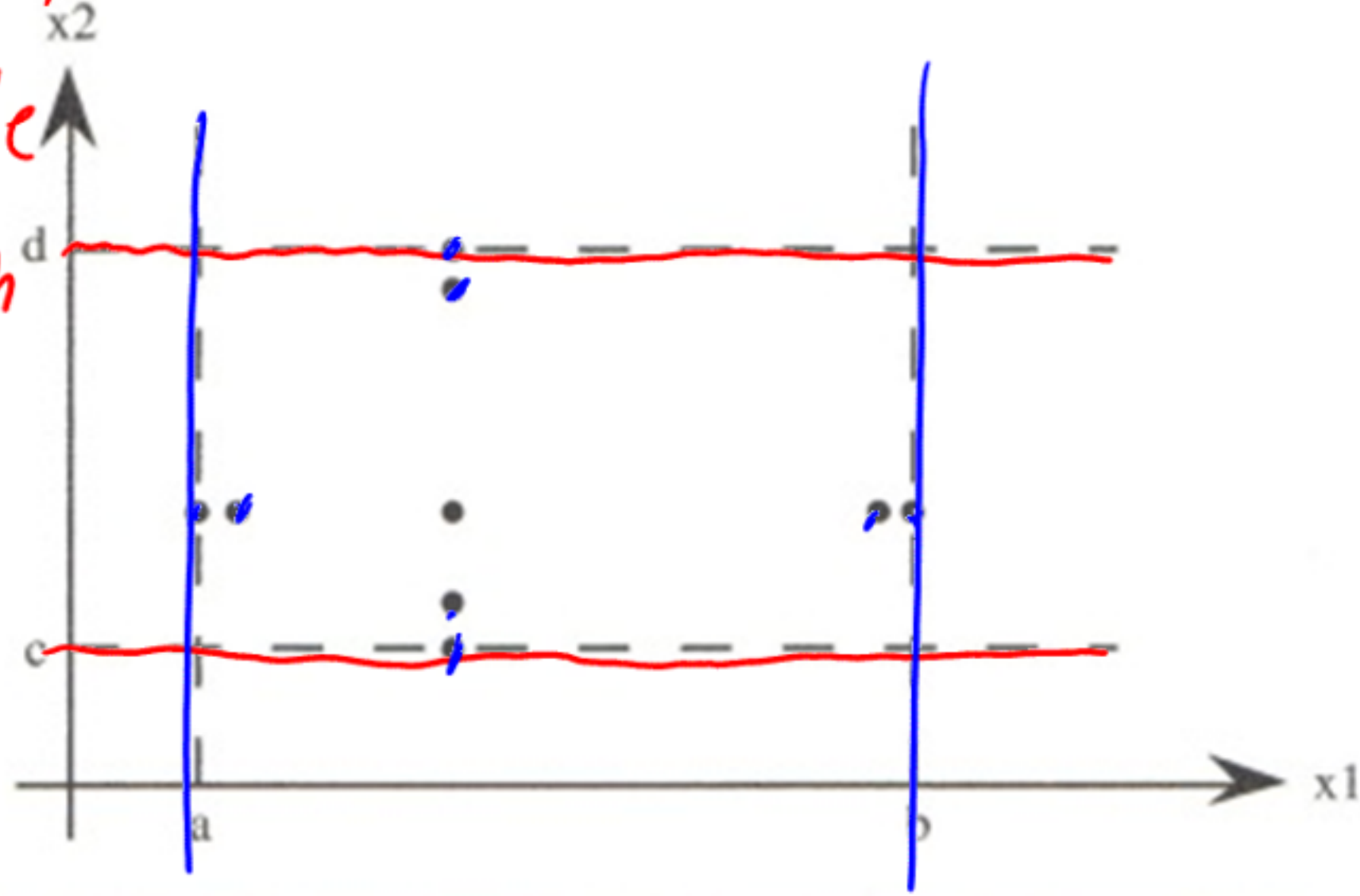
3 w/ BVA

Four general forms

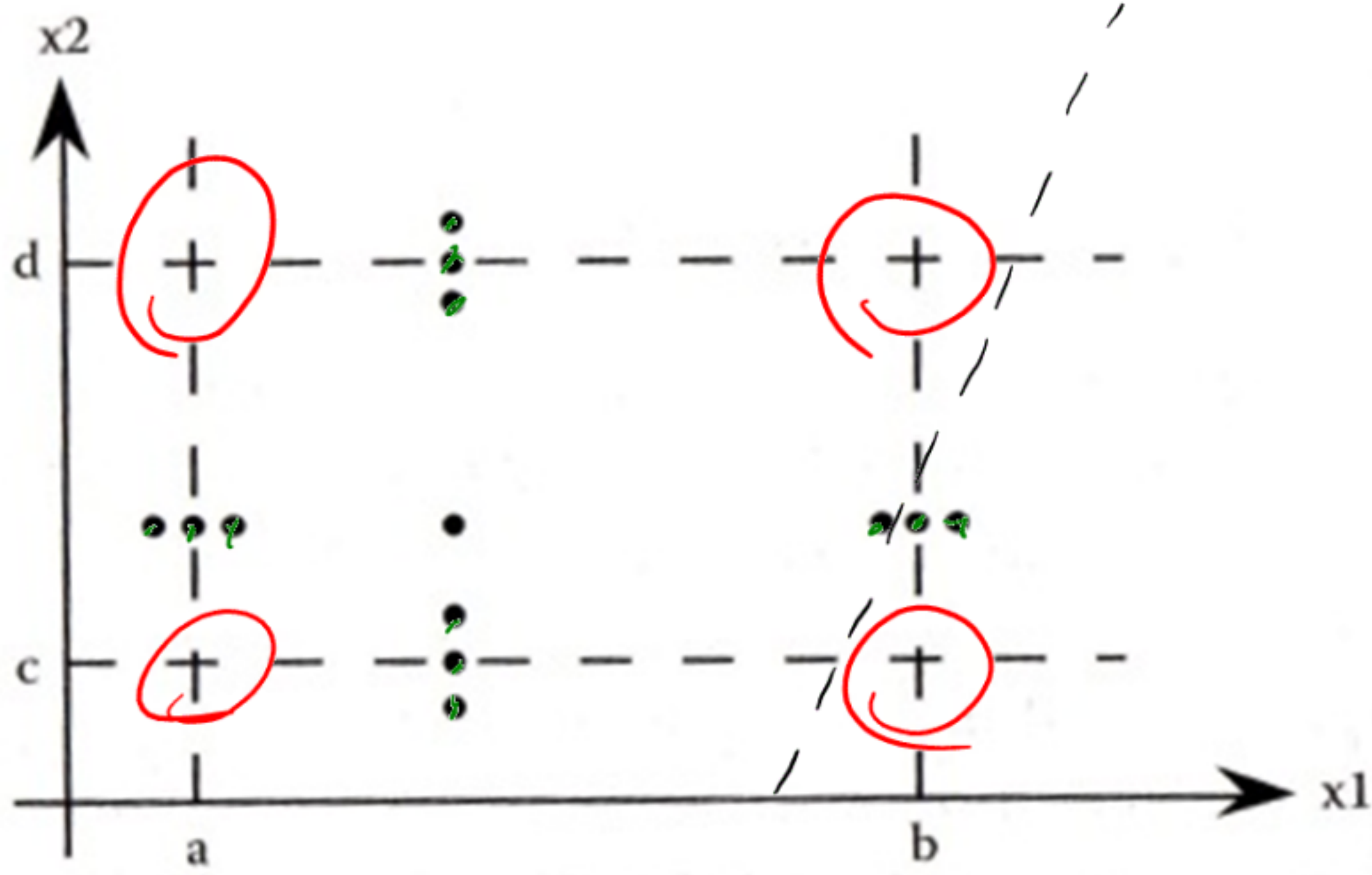
- Basic boundary value analysis
  - Tests at the boundary and one off
- Robustness testing
  - Tests at the boundary as well as one above and one below the boundary
- Worst case testing
  - More thorough than basic or robustness testing
- Robust worse case test of two variables
  - Extension of worst case testing

Basic Boundary Value Analysis

2 variables in the system

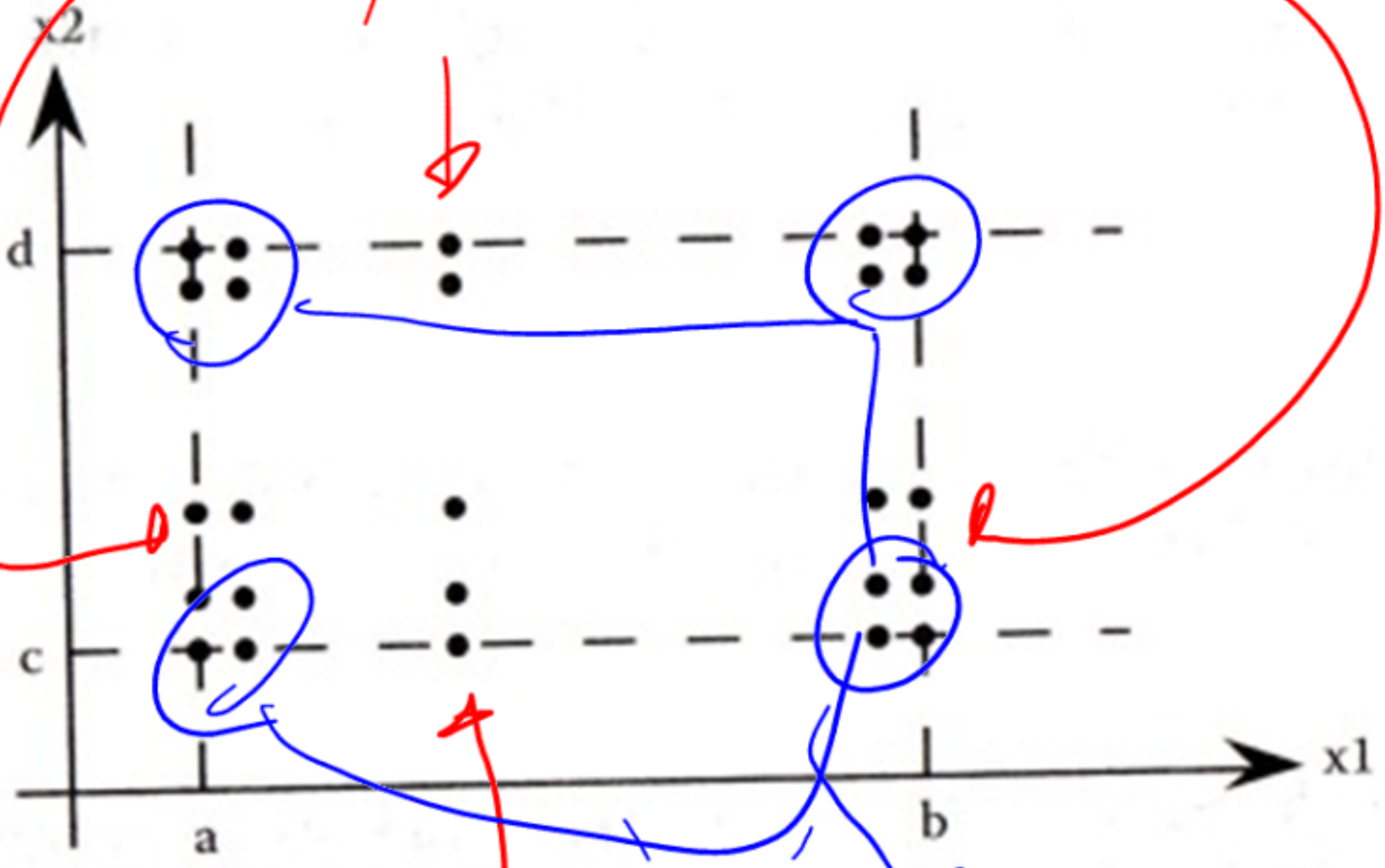


Robustness Testing



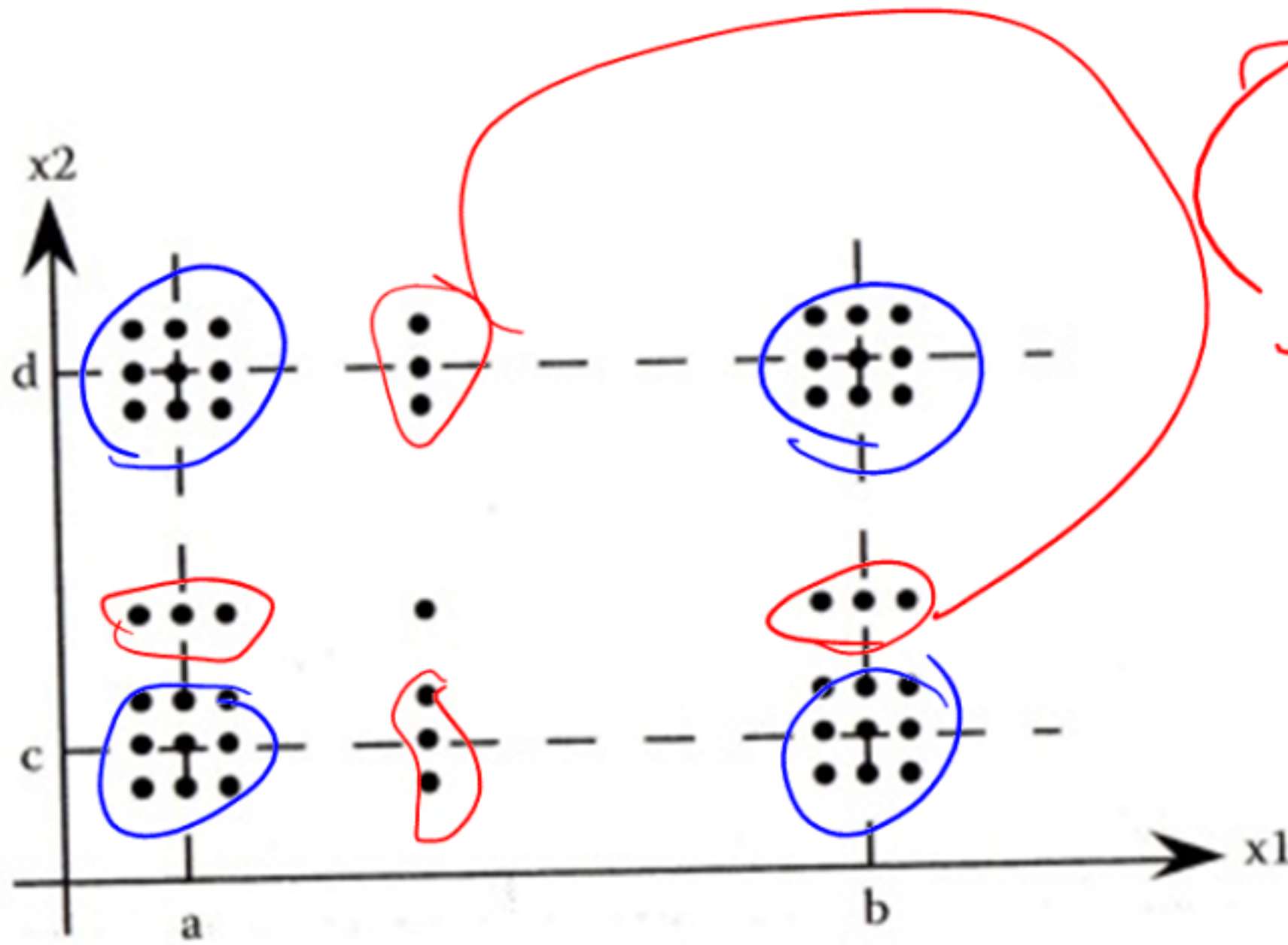
Worst Case Testing

middle tests



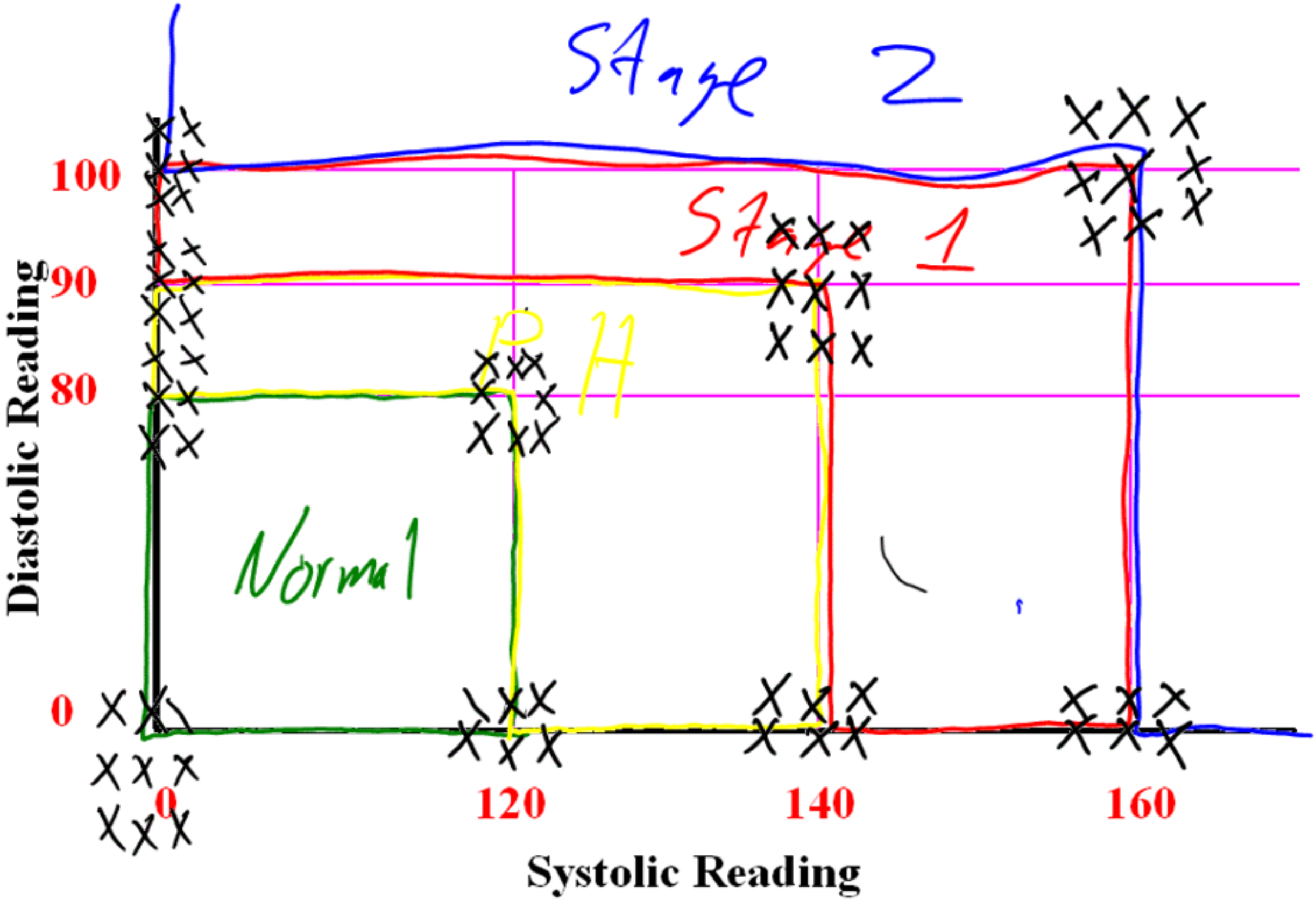
Corners

Robust worst case testing



Blood Pressure Example

Category	Systolic Reading		Diastolic Reading
Normal	less than 120	and	less than 80
Prehypertension	120 – 139	or	80 – 89
High Blood Pressure (Hypertension) Stage 1	140 – 159	or	90 – 99
High Blood Pressure (Hypertension) Stage 2	160 or higher	or	100 or higher


Stage 2
Stage 1
PH
Normal
100
90
80
0
0
120
140
160
Diastolic Reading
Systolic Reading