Software Development II
Unit 4: Blackbox Testing
Boundary Value Analysis (BVA)

Markus Roggenbach

February 2019

Recap

Complete testing is not feasible.

Main testing approaches:

Black box testing:

Test case selection based on specification.

White box testing:

Test case selection based on program code.

Literature

P C Jorgensen: *Software Testing: A Craftman's Approach*. 4th Edition, CRC Press, 2014.

You will learn

A first, simple, and effective blackboard testing technique

In particular, we look into

- how to select values for which to exercise an SUT
- how to combine these values to obtain a test suite

Boundary Value Analysis

Rationale of Boundary Value Analysis

Focus on the boundary of the input space!

Rationale:

- Loop conditions test often for $<$ instead of $\leq$ and vice versa
- Counters are often "off by one".
- US Army (CECOM) study: surprising portion of faults turn out to be boundary value faults (Jorgensen).

Where do the boundaries come from?

Applications often define boundaries of data:

- Array size, e.g., 720 * 1040 from the screen resolution.
- Physical limitations , e.g.,
  - no more than 31 days per month.
  - load capacity of an elevator (≥ 0, sensor has a maximum)
  - temperature (min and max of the sensor)

Step 1 in Boundary Value Analysis

For each input variable define

- minimum – *"min"*
- just above minimum – *"min+"*
- normal value – *"nom"*
- just below maximum – *"max-"*
- maximum – *"max"*

Example: Type “byte” of Java.

```
The values of the integral types are integers in the following ranges:
For byte, from -128 to 127, inclusive.
```

(from Gosling: Java language documentation)

Byte could be the temperature readings of a sensor.

min	=	-128
min+	=	-127
e.g. nom	=	-4
max-	=	126
max	=	127

Single Fault assumption (Reliability theory)

Failures are only rarely the result of the simultaneous occurrence of two (or more) faults.

This is an assumption on the SUT – that one ought to justify when applying Boundary Value Analysis (not in the context of this module).

Step 2 in Boundary Value Analysis

Make the single fault assumption.

Test cases generated by taking the following input values:

- keep all but one variable at their normal values
- let that variable range over all the values from the previous slide.

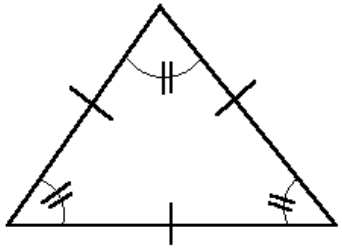
For 2 input variables: 9 test cases.
For n input variables: $4n + 1$ test cases.

Example – continued for type "byte"

name	input 1	input 2	expected result
T1	-128	-4	
T2	-127	-4	
T3	-4	- 4	
T4	126	-4	
T5	127	-4	
T6	-4	-128	
T7	-4	-127	
T8	-4	126	
T9	-4	127	

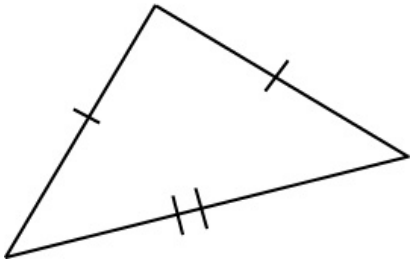
Example: BVA for the triangle problem

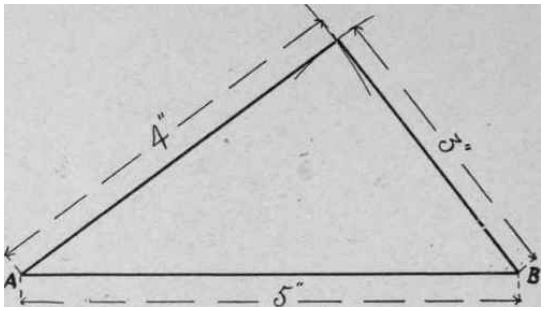
Different triangles



Equilateral Triangle

Isosceles: 2 Sides Are Equal





The computational problem

Triangle Problem:

Input: three integers a, b and c

Output: out of range, if c1, c2 or c3 fails

otherwise:

equilateral, if a=b=c

isosceles, if exactly two of the inputs are equal

scalene, if the inputs are pairwise different

not a triangle, if c4, c5, or c6 fails

c1	$1 \leq a \leq 200$	c4	$a < b + c$
c2	$1 \leq b \leq 200$	c5	$b < a + c$
c3	$1 \leq c \leq 200$	c6	$c < a + b$

Boundary Value Testing

Table 5.1 Boundary Value Analysis Test Cases

Case	*a*	*b*	*c*	*Expected Output*
1	100	100	1	Isosceles
2	100	100	2	Isosceles
3	100	100	100	Equilateral
4	100	100	199	Isosceles
5	100	100	200	Not a Triangle
6	100	1	100	Isosceles
7	100	2	100	Isosceles
8	100	100	100	Equilateral
9	100	199	100	Isosceles
10	100	200	100	Not a Triangle
11	1	100	100	Isosceles
12	2	100	100	Isosceles
13	100	100	100	Equilateral
14	199	100	100	Isosceles
15	200	100	100	Not a Triangle

Example: BVA for the modified multiplication problem

On the whiteboard

Multiplication:

Input: natural numbers a, b with $a \leq 10, b \leq 15$

Output: the natural number $a * b$

name	a	b	expected result
. . .	. . .	. . .	. . .

What you have learned in this unit

Definitions

- Whitebox testing
- Blackbox testing
- Process of deriving a BVA testsuite as UML Activity Diagram

You should be able to explain by example

- Where the boundaries of variable domains come from
- Why Boundary Value Analysis can be considered a 'good' testing method
- How to design a test suite for Boundary Value Analysis