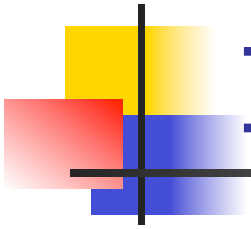
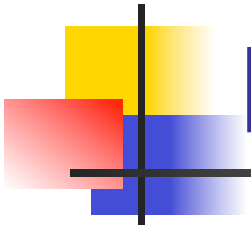
Boundary Value Testing

Chapter 5



Introduction

- Input domain testing is the most commonly taught (and perhaps the most commonly used) software testing technique
- We will see a number of approaches to input domain testing
- We will then study some of its limitations



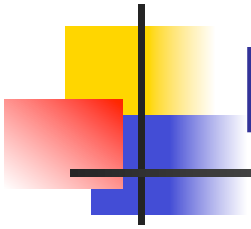
Boundary Value Analysis

- Many programs can be viewed as a function *F* that maps values from a set *A* (its domain) to values in another set *B* (its range)

$$F : A \longrightarrow B$$

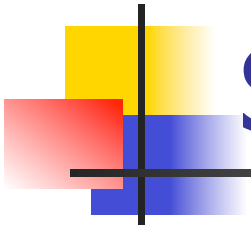
- The input variables of F will have some (possibly unstated) boundaries:

$$a \leq x_1 \leq b \qquad\qquad c \leq x_2 \leq d$$



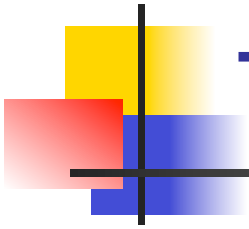
Boundary value analysis

- For each variable, select five values
  - Minimum
  - Just above the minimum
  - Nominal
  - Just below the maximum
  - Maximum



Single fault assumption

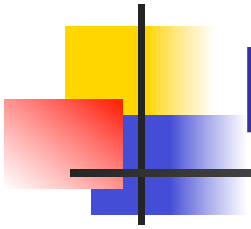
- Failures are only rarely the result of the simultaneous occurrence of two (or more) faults
- Generate test cases as such for all i
  - Values of all but one variable x_i at nominal
  - x_i assumes all 5 values from previous slide
- What is the number of test cases?



Two-variable function test cases

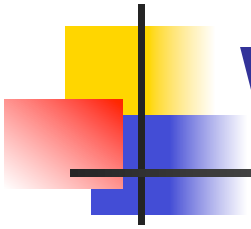
$<x_{1nom}, x_{2min}>$	$<x_{1min}, x_{2nom}>$
$<x_{1nom}, x_{2min+}>$	$<x_{1min+}, x_{2nom}>$
$<x_{1nom}, x_{2nom}>$	$<x_{1nom}, x_{2nom}>$
$<x_{1nom}, x_{2max-}>$	$<x_{1max-}, x_{2nom}>$
$<x_{1nom}, x_{2max}>$	$<x_{1max}, x_{2nom}>$

Let's apply this to the Triangle problem



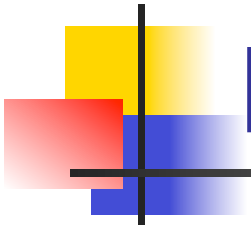
Robustness testing

- A simple extension to boundary value analysis
- Add two more values per variable
  - Slightly greater than the maximum
  - Slightly less than the minimum
- What is the expected output?
  - Hopefully error message, system recovers
- Implementing these test cases may not be possible



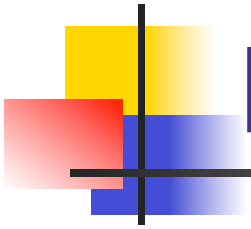
Worst-Case Testing

- Rejects the simple fault assumption and tests all combinations of values
- Instead of 4n+1 test cases, we have 5^n
- Often leads to a large number of test cases with low bug-finding power
- Usually better to apply Special Value Testing: test cases based on the tester's intuition



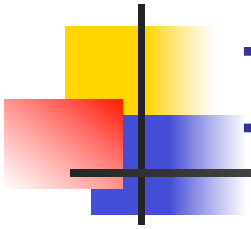
Robust Worst-Case Testing

- Add the values min– and max+ to the possible variable values
- Now take all combinations of variable values
- What is the number of test cases?



Limitations

- Does not work well for boolean variables
  - We will see a more suitable approach next week
- Does not work well for logical variables
  - PIN, transaction type
- Assumes independent variables



In class activity

- You are asked to test a software program that accepts a date as input and returns the next date
- Apply Boundary Value Analysis
- How satisfied are you with the results?