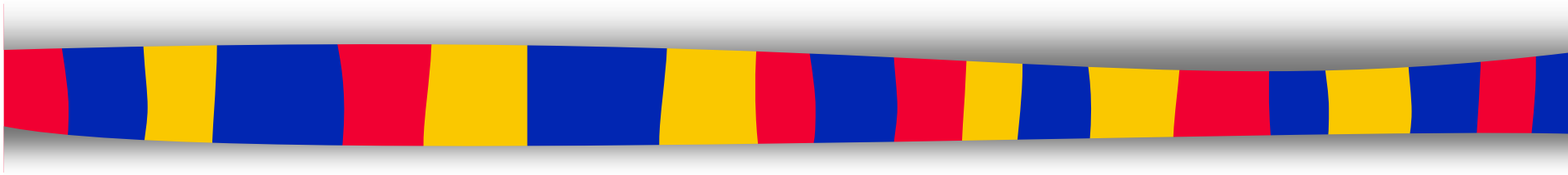
EECS 4313

Software Engineering Testing



Topic 04:

Boundary Value Testing

Zhen Ming (Jack) Jiang

Relevant Reading

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

$$F : A \rightarrow B$$

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

Boundary value analysis

- For each variable, select five values
  - Minimum (min)
  - Just above the minimum (min+)
  - Nominal (nom)
  - Just below the maximum (max-)
  - Maximum (max)

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 the previous slide (min, min+, nom, max-, max)
- What is the number of test cases?

Two-variable function test cases

$\langle x_{1nom}, x_{2min} \rangle$

$\langle x_{1nom}, x_{2min+} \rangle$

$\langle x_{1nom}, x_{2nom} \rangle$

$\langle x_{1nom}, x_{2max-} \rangle$

$\langle x_{1nom}, x_{2max} \rangle$

$\langle x_{1min}, x_{2nom} \rangle$

$\langle x_{1min+}, x_{2nom} \rangle$

$\langle x_{1nom}, x_{2nom} \rangle$ **duplicate**

$\langle x_{1max-}, x_{2nom} \rangle$

$\langle x_{1max}, x_{2nom} \rangle$

Boundary value analysis (BVA) - Advantages

- Independent variables
  - Single fault assumption
- Physical quantities

Boundary value analysis (BVA) - Limitations

- Does not work well for Boolean variables
  - **Why are these not suitable?**
- Does not work well for logical variables
  - PIN, transaction type
  - **Why are these not suitable?**
- When variables are not independent – i.e., are dependent

Extensions to the (normal) boundary value testing

- Robustness testing
- Worst case testing
- Robust worst case testing
- Special value testing
- Random testing

Robustness testing

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

Worst-Case Testing

- Rejects the single 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?*

Other extensions to the boundary value testing

- Special value testing
  - Use best engineering judgment
    - Intuition
    - Domain knowledge
    - Experience
    - Soft spots
- Random testing
  - Select random values for each variable