

V&V Activities

- Reviews, Inspections, and Walkthroughs
- Testing
 - Formal and informal methods
 - Dynamic (run tests) and static (reviews, formal verification)
 - Levels: Unit, Integration, System, Regression
 - Techniques: Functional (black-box), Structural (white/clear-box), Stress, ...

Testing Glossary

- Error: mistake, bug
- Fault: result of an error, defect
- Failure: when a fault executes
- Incident: symptom associated with a failure
- Test Case: set of inputs and expected output
- Clean Tests: show something works
- Dirty Tests: show something doesn't work

Testing

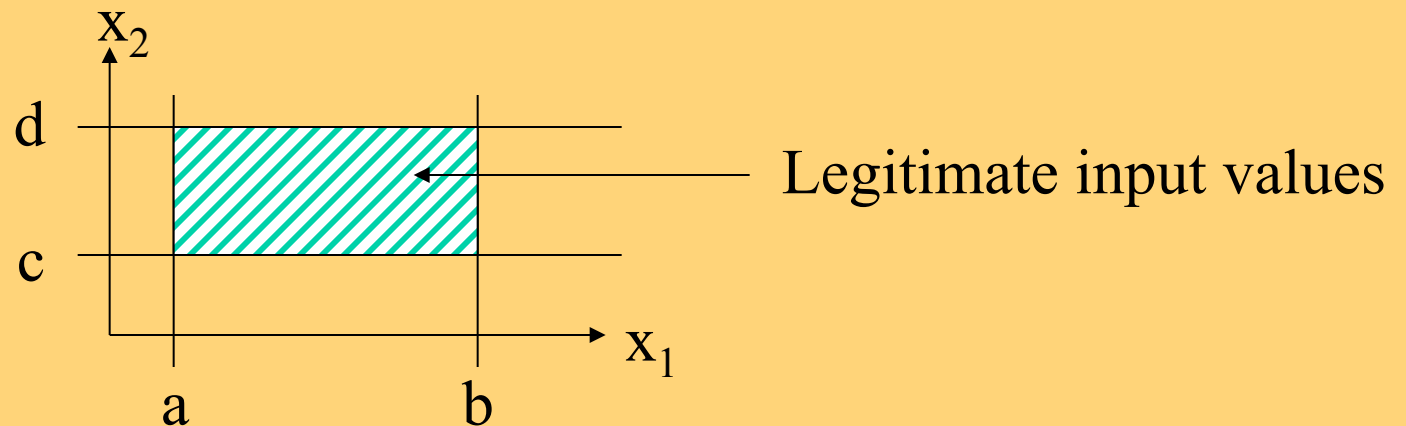
- A process of executing a program with the intent of finding errors
- Objective: to find defects
- Can detect the presence of defects, but not their absence

Testing Approaches

- Functional Testing
 - Boundary Value Analysis
 - Equivalence Class
 - Decision Tables
 - Cause and Effect
- Structural Testing (white/clear-box)
 - Program graphs
 - Define-use paths
 - Program slicing

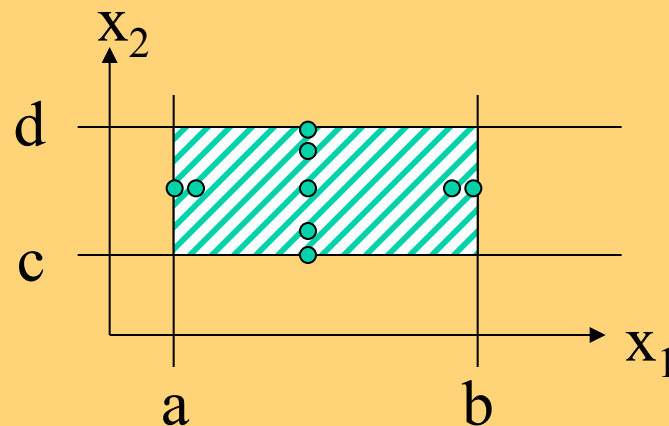
Boundary Value Analysis

- Think of a program as a function
 - $f(x_1, x_2)$
 - x_1 and x_2 have some boundaries
 - $a \leq x_1 \leq b$ (range of legitimate values)
 - $c \leq x_2 \leq d$ (a, b, c, d are boundary values)



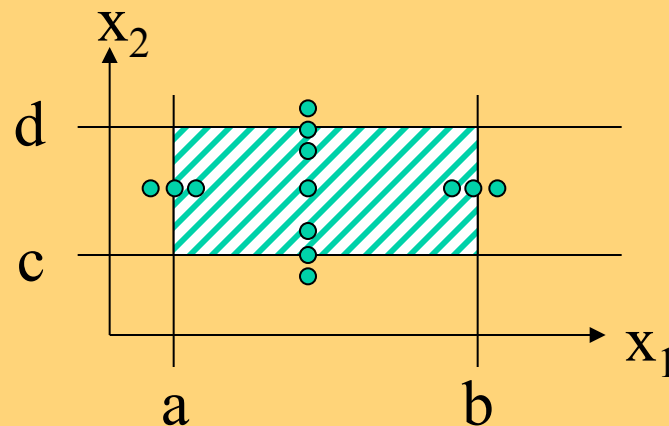
Boundary Value Analysis

- Premise: Bugs tend to lurk around the edges
- Single fault assumption
 - Hold all variables but one constant
 - Vary one to min, min+1, nominal, max-1, max
 - n variables yields $4n + 1$ test cases



BVA Variation

- Also test beyond boundaries
 - min-1, max+1
 - n variables yields $6n + 1$ test cases



Worst-case BVA

- Reject single fault assumption
 - Allow multiple variables to vary
 - n variables yields 5^n test cases

