Software Testing & Quality Assurance

Boundary Value Testing

- Boundary value analysis
- Single fault assumption
- Robustness testing
- Worst case testing
- Special value testing
- Random testing

Credits & Readings

- The material included in these slides are mostly adopted from the following books:
  - *Software Testing: A Craftsman's Approach*, by Paul Jorgensen, CRC PRESS, third edition, ISBN: 0-8493-7475-8
  - Cem Kaner, Jack Falk, Hung Q. Nguyen, *"Testing Computer Software"* Wiley (see also http://www.testingeducation.org/)
  - Cem Kaner, James Bach, Bret Pettichord, *"Lessons Learned in Software Testing",* Wiley
  - Paul Ammann and Jeff Offutt, *"Introduction to Software Testing"*, Cambridge University Press
  - Kent Beck, *"Test-driven Development by Example"* Addison-Wesley
  - Robert Binder, *"Testing Object-Oriented Systems: Models, Patterns, and Tools"* Addison-Wesley
  - Glen Myers, *"The Art of Software Testing"*

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 *boundary value analysis*
- We will then study some of the limitations of domain testing

What is Boundary Value Analysis?

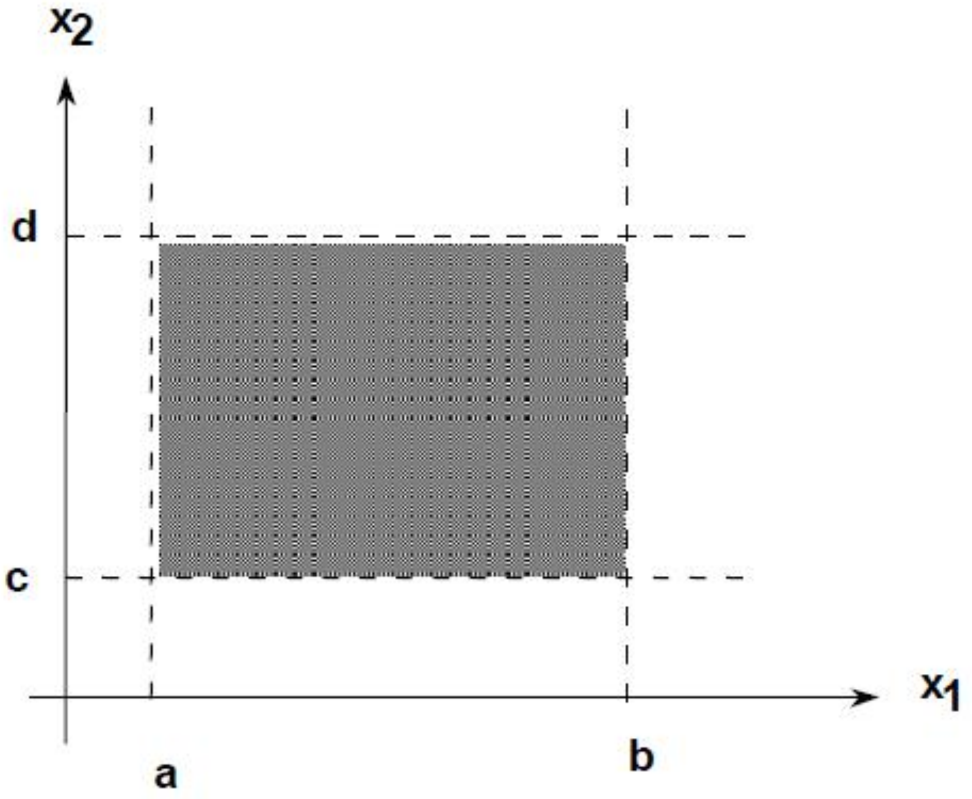
- Many programs can be viewed as a function *F* that maps values from a set *A* (its domain) to values in another set *B* (its range)

$$F : A \to B$$

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

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

Input domain for variables x_1 and x_2



What does BVA focus on?

- It focuses on the boundary of the input space to identify test cases
- The rationale behind this is that errors tend to occur near the extreme values of an input variable
  - Many times when a system is tested the function result is “off by one”

What is the basic idea?

- To use input variable values at their…
  - – Minimum
  - – Just above the minimum
  - – A nominal value
  - – Just below their maximum
  - – At their maximum

The “single fault” assumption

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

How does BVA work?

- Boundary value analysis test cases are obtained by
  - holding the values of <u>all but one</u> variable at their nominal values, and
  - letting that variable <u>assume its extreme values</u> i.e.
    - Minimum
    - Just above the minimum
    - Nominal
    - Just below the maximum
    - Maximum

Example: BVA test cases for a two-variable function

$<x_{1nom}, x_{2min}>$

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

$<x_{1nom}, x_{2nom}>$

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

$<x_{1nom}, x_{2max}>$

$<x_{1min}, x_{2nom}>$

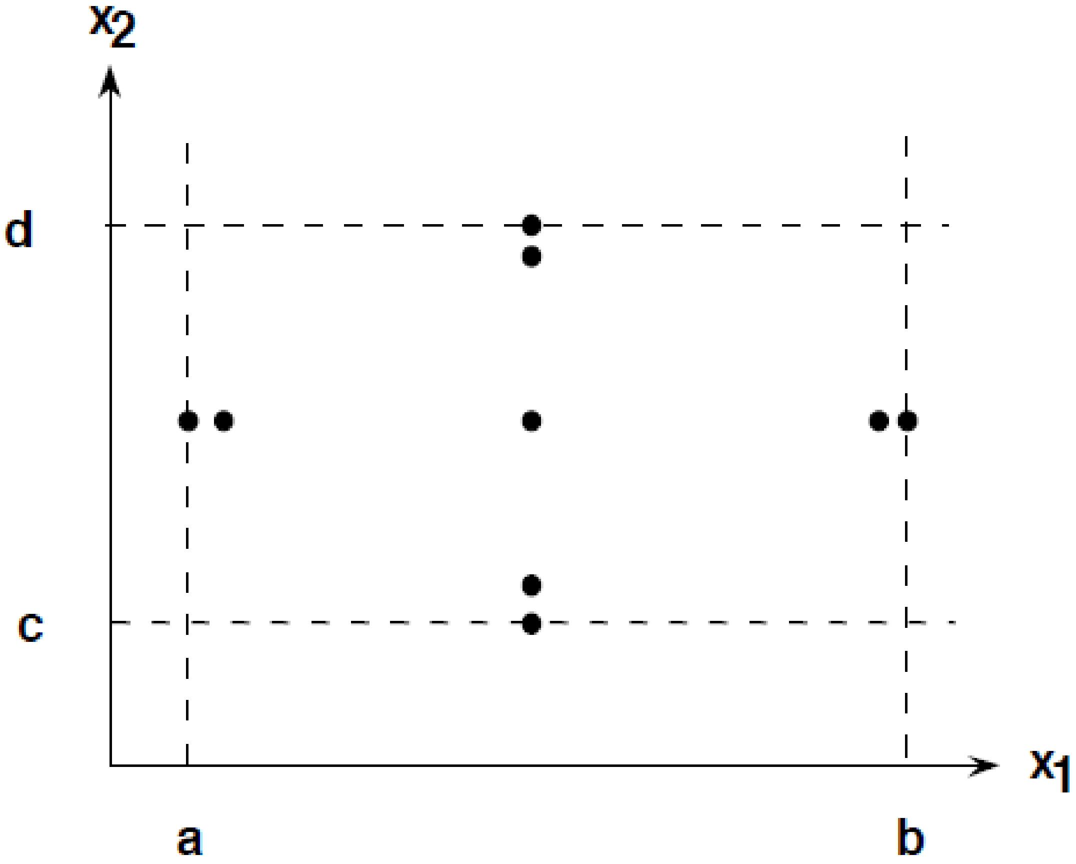
$<x_{1min+}, x_{2nom}>$

$<x_{1nom}, x_{2nom}>$

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

$<x_{1max}, x_{2nom}>$

BVA test cases for x_1 and x_2



In-class activity

- Let’s apply BVA to the
  - Adder program
    - Input domain: 2 integers (2-digits each)
  - Triangle problem
    - Input domain: 3 sides (values 1-200 each)

Limitations

- Boundary value analysis works well when the program is a function of several *independent variables* that represent bounded physical quantities
  - Physical boundaries can be extremely important
    - Airport in Phoenix shut down
      - Air temperature was 122 degrees
      - Instruments could only accept up to 120 degrees
- It won't work for the *NextDate* program because
  - dependencies exist among the *month, day* and *year* variables
- It does not work well for logical variables
  - Customer's PIN in ATM, transaction type
- It does not work well for Boolean variables
  - Boolean variables lend themselves to *decision table-based* testing (we will discuss later)

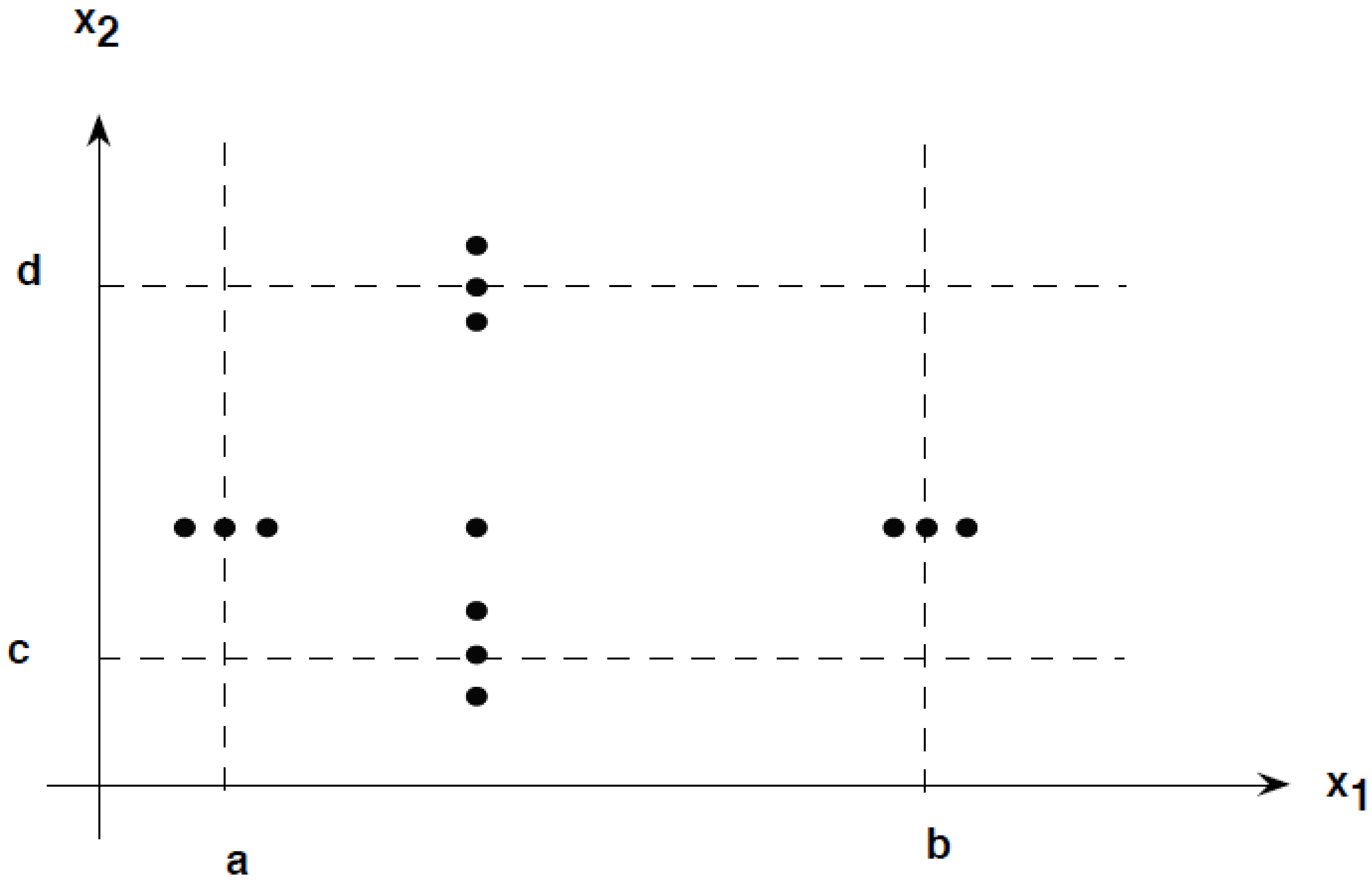
How can BVA be generalized?

- It can be generalized in 2 ways:
  - By the number of variables
  - By the number of ranges
- Generalizing the number of variables is easy
  - Function of *n* variables, we hold all but one at the nominal values and let remaining variable assume min, min+, nom, max-, and max
    - It makes *4n+1* test cases

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

Robust testing: test cases for x_1 and x_2



In-class activity

- Let’s apply robust testing to the
  - Adder program
    - Input domain: 2 integers (2-digits each)
  - Triangle problem
    - Input domain: 3 sides (values 1-200 each)

Worst case testing

- Rejects the single fault assumption and tests all combinations of values
- Instead of *5n* 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* (*i.e.* test cases based on the tester's intuition)
- Best application for worst-case testing when
  - physical variables have numerous interactions
  - failure of the function is extremely costly
  - Use of automated testing tools

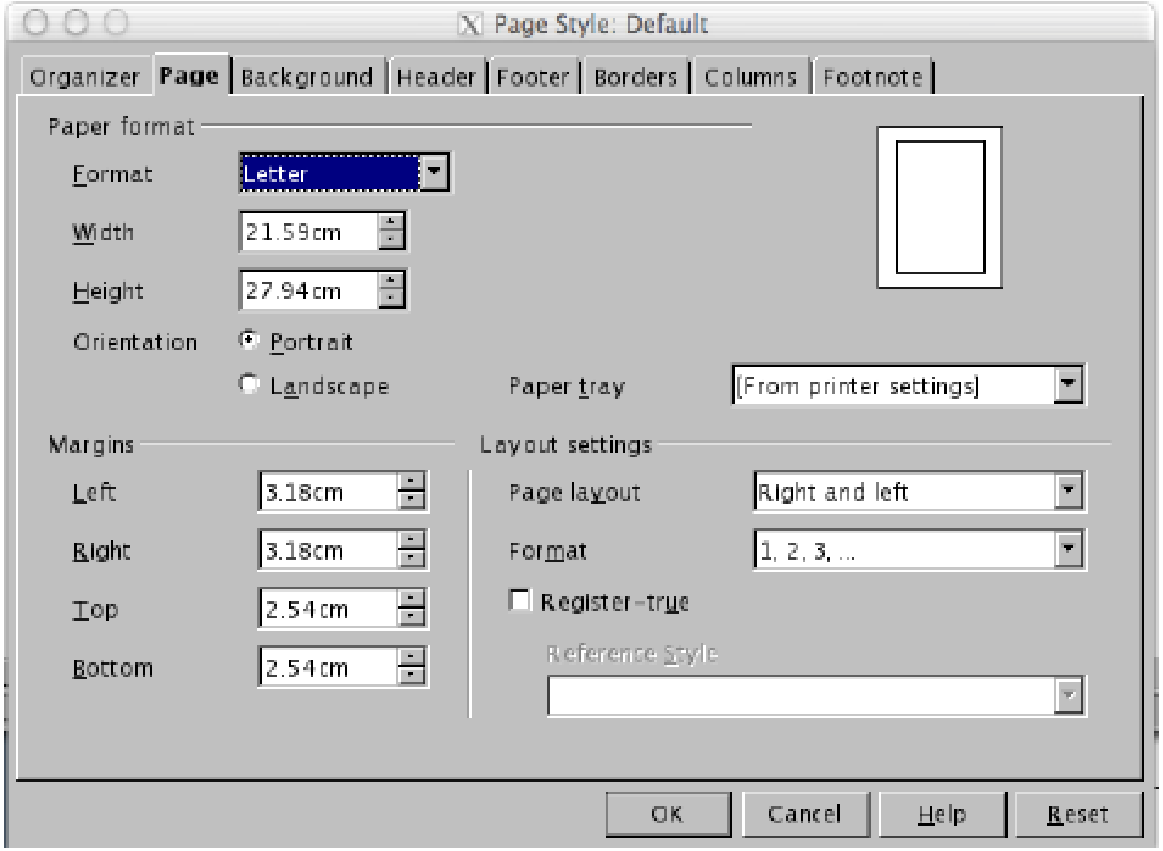
Special value testing

- Most widely practiced form of functional testing
- Most intuitive and least uniform
- Occurs when a tester uses his/her domain knowledge, experience with similar programs, and information about "soft spots" to devise test cases
  - Also called "ad hoc testing" or "seat-of-the-pants" testing
  - No guidelines used other then best engineering judgment
  - Can be very useful, often more effective in revealing error results

Random testing

- Besides always choosing min, max, min+….
  - Use a random number generator to pick test case values
  - Avoids biases in testing

In class activity



Paper format

Format: Letter

Width: 21.59cm

Height: 27.94cm

Orientation: ● Portrait ○ Landscape

Paper tray: [From printer settings]

Margins

Left: 3.18cm

Right: 3.18cm

Top: 2.54cm

Bottom: 2.54cm

Layout settings

Page layout: Right and left

Format: 1, 2, 3, ...

□ Register-true

Reference Style

- Do a domain analysis on <u>page width and height</u>
  - BVA
  - Robust testing
  - Special value
- Assume the spec mentions that
  - Width values between 10cm and 60cm should be handled
  - Height values between 20cm and 100cm
- Can you identify any weaknesses of BVA?