Introduction to Software Testing
Chapter 6
Input Space Partition Testing

Paul Ammann & Jeff Offutt

http://www.cs.gmu.edu/~offutt/softwaretest/

Boundary Value Analysis

- When presented with a boundary condition, test:
  - the valid data just inside the boundary
  - the last possible valid data
  - the invalid data just outside the boundary
- For example:
  - Min – 1 / Max +1
  - First – 1/ Last + 1
  - Highest +1/Lowest – 1
- Let’s see some examples.

Boundary Value Analysis (Examples)

- A text entry field allows 1 to 255 characters.
- A program reads and writes to a CD-R.

Boundary Value Analysis

- How does that technique differ from input space partitioning?
  - Instead of selecting an element in a block as a representative, this technique requires that one or more elements be selected such than each edge of the block is tested.
  - Instead of focusing attention on the input conditions (input space), test cases are derived by considering the result space (output equivalence class).
- Let’s see an example.

Boundary Value Analysis – Exercise!

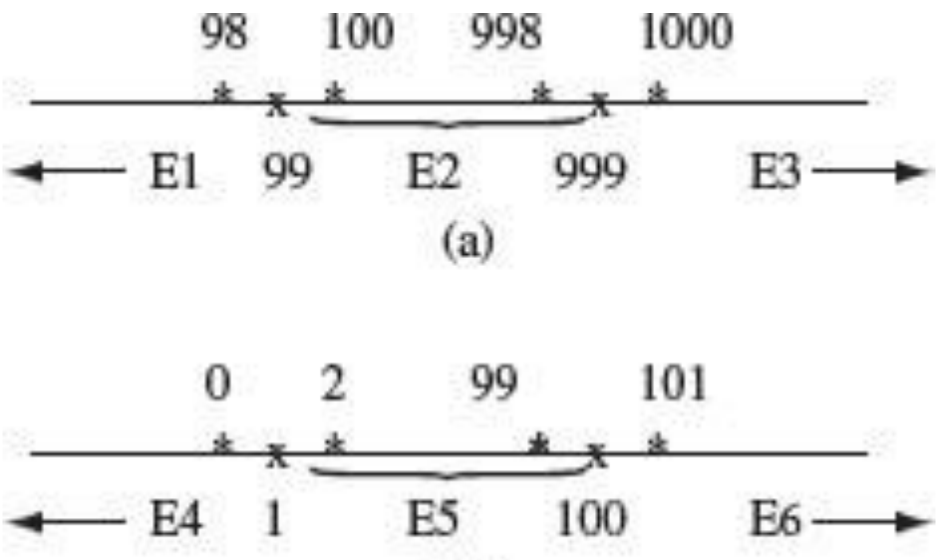
- Consider a method findPrice that takes two inputs code (integer) and quantity (integer).
- Code represents an item's code, and quantity represents the quantity purchased from that item.
- findPrice uses the inputs to retrieve the unit price, description, and the total price of the item corresponding to that code.
- findPrice should display an error message if either of the two inputs is incorrect.
- Code must be in the range [99, 999].
- Quantity must be in the range [1,100].

Boundary Value Analysis - Example

1. Identify the equivalence partitions.

- Code E1: values < 99; E2: 99 <= values <=999
  E3: values > 999
- Quantity E4: values < 1; E5: 1<=values<=100
  E6: values > 100.

2. How many boundaries do we have?



(a)

(b)

Remember!
"at and around values"

Boundary Value Analysis - Example

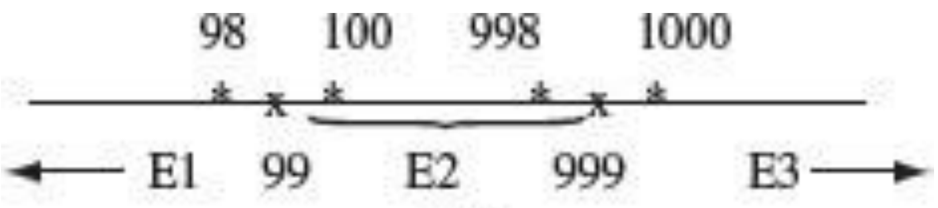
1. Identify the equivalence partitions.

- Code E1: values < 99; E2: 99 <= values <=999

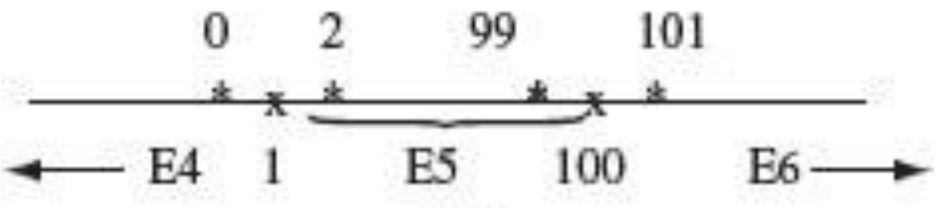
  E3: values > 999
- Quantity E4: values < 1; E5: 1<=values<=100

  E6: values > 100.

2. How many boundaries do we have?



(a)



(b)

Boundary Value Analysis - Example

1. Identify the equivalence partitions.

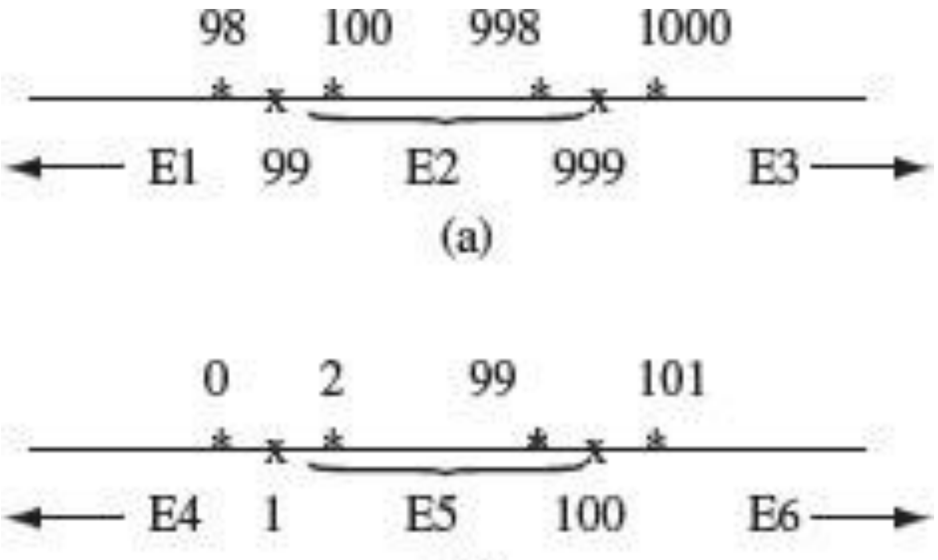
- Code E1: values < 99; E2: 99 <= values <=999

  E3: values > 999

- Quantity E4: values < 1; E5: 1<=values<=100

  E6: values > 100.

2. How many boundaries do we have?



(a)

(b)

Any concerns about your created TCs?

Boundary Value Analysis - Example

- Consider a method textSearch to search for a string s in text txt.
- Position of characters in txt begins with 0 for the first character, 1 for the second character, and so on.
- Both s and txt are input to textSearch method
- The method returns an integer p.
- If p >=0, it denotes the starting position of s within txt.
- If p <0, it implies that s was not found within txt.

Boundary Value Analysis - Example

1. Identify the equivalence partitions.

- s E1: empty string; E2: non-empty string
- txt E3: empty string; E4: non-empty string

Q: What are the boundaries for s and txt?

- There are no upper bounds for s and txt (i.e., no upper limit on their length)
- The lower bound on their length is 0.
- Hence, we have one boundary case for s (covered by E1), and one boundary case for txt (covered by E3).

Boundary Value Analysis - Example

2. Partition the output space into equivalence classes.

- p: E5: p <0 ; E6: p >= 0.

Q: What are the boundary conditions for p?

- We have only one boundary p=0

- To obtain an output within E5, s must not be in txt
- How can we cause the output p to be 0?
  1. s must be in txt
  2. s must appear at the start of txt.
  - Those two conditions force us to add a new test based on boundary value analysis:
    - S= “This” , txt = “This is a good day”.
- Note: NONE of the identified equivalence partitions suggested this specific case!

Boundary Value Analysis - Example

- We have six equivalence classes (E1 to E6)
- s E1: empty string; E2: non-empty string
- txt E3: empty string; E4: non-empty string
- p: E5: p <0 ; E6: p >= 0.
- ALSO, we have boundary for p = 0, and boundaries for s and t to be empty strings.

Hence, we have 4 tests so far:

- T1: s= empty, txt = “This is a good day”(covers E1, E4, E5)
- T2: s = “This”, txt = empty (covers E2, E3, E5)
- T3: s = “a good”, txt = “This is a good day”(covers E6)
- T4: s= “This” , txt = “This is a good day”(imposed by boundary value analysis for p)

Boundary Value Analysis - Example

- But… we need to ensure that textSearch works properly when s occurs at the boundaries of txt.
- T4 ensured that it works at the start of txt.
- We need another test to ensure that it works at the end of txt.
  - T5: s= “day”, txt = “This is a good day”.
- Any more boundaries?
  - Yes!
  - We need to examine test inputs that are NEAR the boundaries.

Boundary Value Analysis - Example

Near the boundaries?

1. s starts at position 1 in txt (near the lower boundary of txt)

- T_6 : (*s* = “aughter”, *txt* = “Laughter is good for the heart.”)

2. s ends at one character before the last character in txt. (near the end of txt)

- T_7 : (*s* = “heart”, *txt* = “Laughter is good for the heart.”)

Boundary Value Analysis - Example

Near the boundaries?

3. All but the first character of *s* occur in *txt* starting at position 0. Expected output: *p* = –1.

- T_8 : (*s* = “gLaughter”, *txt* = “Laughter is good for the heart.”)

4. All but the last character of *s* occur in *txt* at the end. Expected output: *p* = –1.

- T_9 : (*s* = “heart.d”, *txt* = “Laughter is good for the heart.”).

Sub-Boundary Conditions

- Are boundary conditions that rely on your domain knowledge about a topic.
- Example: ASCII table

References

- “The Art of Software Design”, Third edition, chapter 4