Equivalence Class Testing

Chapter 6

Boundary value problems

- **What problems does boundary value testing have?**

Boundary value problems – 2

- Boundary Value Testing derives test cases with
  - Serious gaps
  - Massive redundancy

Motivation for equivalence class testing

- **What are the motivations for equivalence class testing?**

Motivation for equivalence class testing – 2

- Avoid redundancy
  - Have fewer test cases
- Complete testing
  - Remove gaps

Addressing the motivation

- **How do equivalence classes meet the motivations of complete testing and avoiding redundancy?**

Addressing the motivation– 2

- The variable domain is partitioned into disjoint sub-sets

Assumptions

- **What assumptions are made?**

Assumptions – 2

- Program is a function from input to output
- Input and/or output variables have well defined intervals
  - For a two-variable function F(x1,x2)

    $a \le x_1 \le d$, with intervals [a,b), [b,c), [c,d]

    $e \le x_2 \le g$, with intervals [e,f), [f,g]

Assumptions – 3

- Completeness
  - The entire set is represented by the union of the sub-sets
- Redundancy
  - The disjointness of the sets assures a form of non-redundancy
- Choose one test case from each sub-set

Variations

- **What variations are used for equivalence class testing?**

Variations – 2

- Use the same two orthogonal dimensions as in boundary value analysis
  - Robustness
    - Robust-normal distinguishes valid data from invalid data
  - Single/Multiple Fault Assumption
    - Weak-strong distinguishes single from multiple fault
- Combinations give four variations.

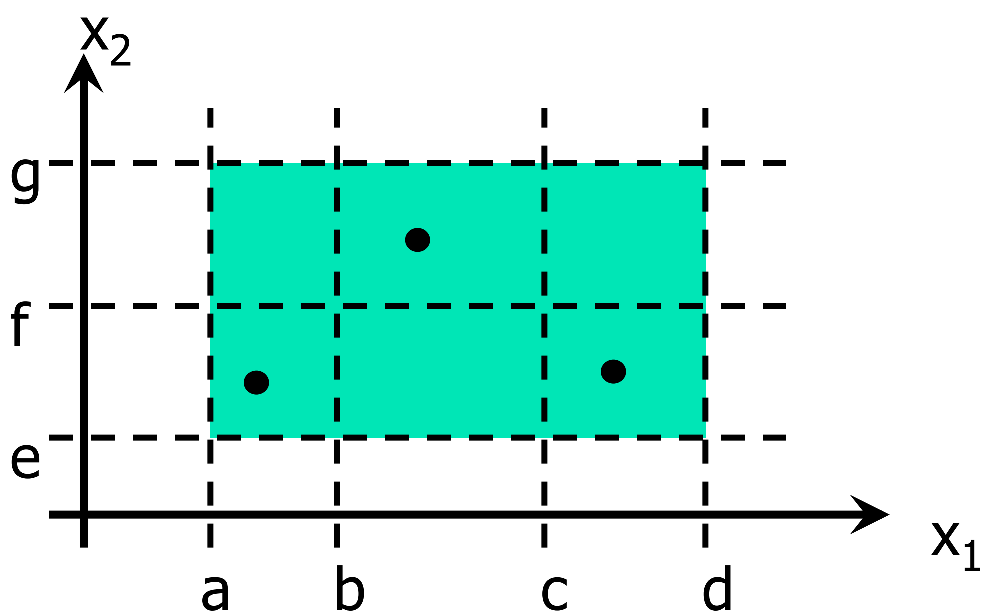
Weak-Normal ECT

- **What is the number of test cases for weak-normal testing?**

Weak-Normal ECT – 2

Number of test cases =

max / [[v : 1 .. #variables • number_equivalence_classes ($variable_v$)]]



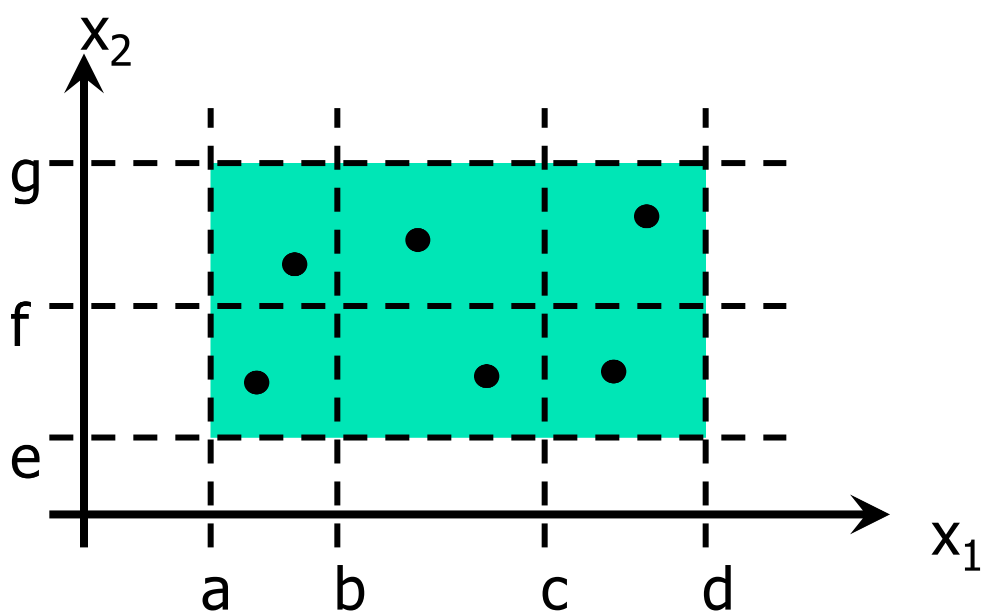
Strong-Normal ECT

- **What is the number of test cases for strong-normal testing?**

Strong-Normal ECT – 2

Number of test cases =

$\times$ / [[v : 1 .. #variables • number_equivalence_classes ($variable_v$)]]



Weak-Robust ECT

- **What is the number of test cases for weak-robust testing?**

Weak-Robust ECT – 2

Number of test cases =

max / [[v : 1 .. #variables • number_equivalence_classes (variable$_v$)]]
+
+/ [[v : 1 .. #variables • number_invalid_bounds (variable$_v$)]]

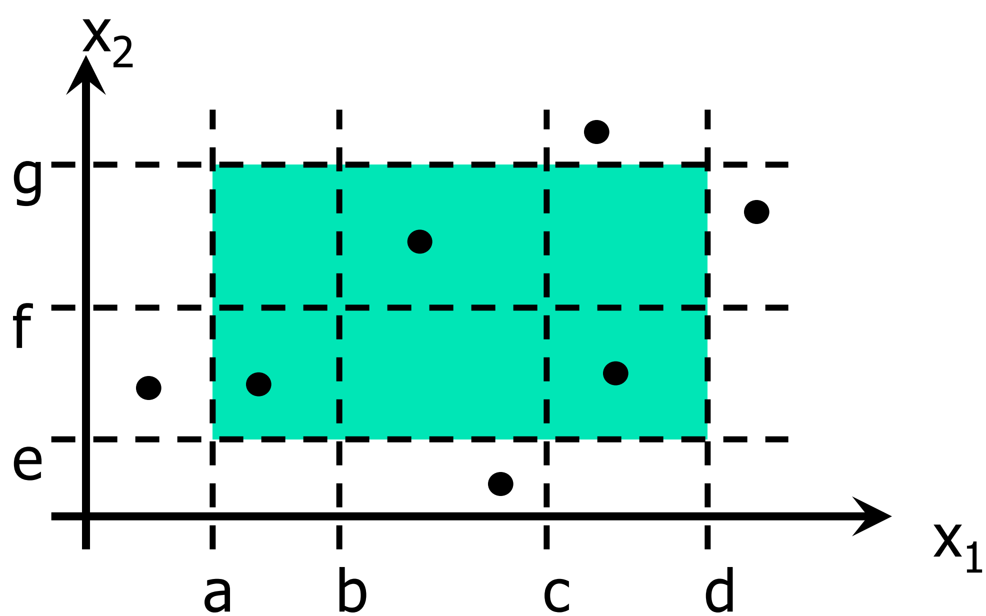


Figure 6.3 in the textbook is incorrect

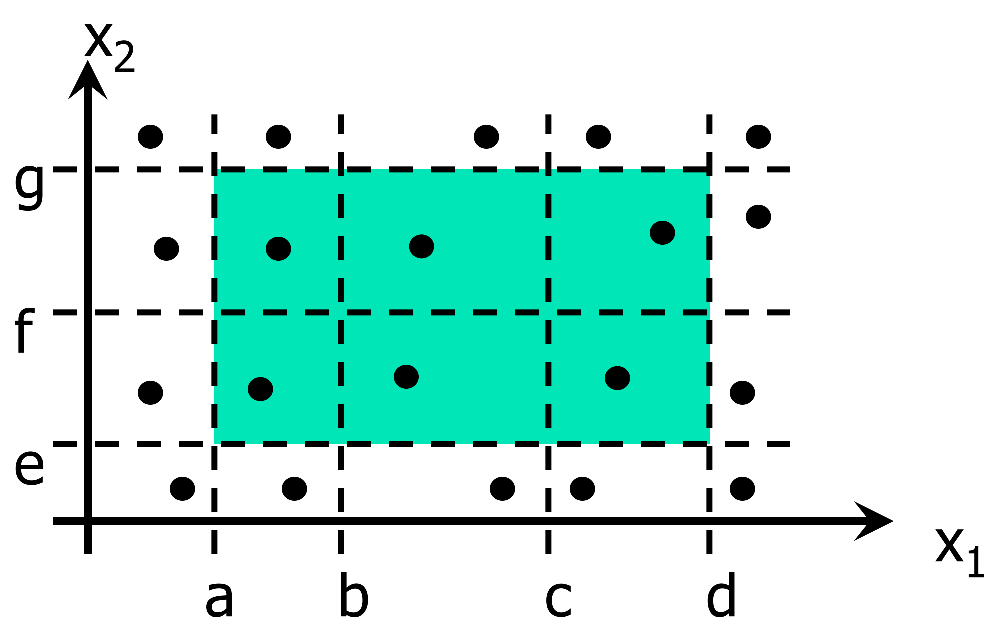
Strong-Robust ECT

- **What is the number of test cases for strong-robust testing?**

Strong-Robust ECT – 2

Number of test cases =

$\times$ / [[v : 1 .. #variables • number_equivalence_classes (variable$_v$)
+ number_invalid_bounds (variable$_v$)]]



Limitations of ECT

- **What are the limitations of equivalence class testing?**

Limitations of ECT – 2

- The same as those for boundary value testing
  - Does not work well for Boolean variables
  - Does not work well for logical variables
  - When variables are not independent – i.e. are dependent
  - Not that useful for strongly-typed languages
- For robust variations same as for boundary value testing
  - Difficult or impossible to determine expected values for invalid variable values

Triangle Output Equivalence Classes

- Four possible outputs
  - Not a Triangle
  - Isosceles
  - Equilateral
  - Scalene

Triangle Output Equivalence Classes – 2

- Output (range) equivalence classes

O1 = {a, b, c : 0 .. 200 | equilateral_triangle (<a,b,c>) }

O2 = {a, b, c : 0 .. 200 | isoceles_triangle (<a,b,c>) }

O3 = {a, b, c : 0 .. 200 | scalene_triangle (<a,b,c>) }

O4 = {a, b, c : 0 .. 200 | not_a_triangle (<a,b,c>) }

What are the number of test cases for

- **weak-normal?**
- **strong-normal?**
- **weak-robust?**
- **strong-robust?**

Why don't the previous formulas work?

Triangle – Weak Normal Test Cases

Test Case	a	b	c	Expected Output
WN1	5	5	5	Equilateral
WN2	2	2	3	Isosceles
WN3	3	4	5	Scalene
WN4	4	1	2	Not a Triangle

Triangle – Weak Robust Test Cases

Weak-normal cases + following error cases

Test Case	a	b	c	Expected Output
WR1	-1	5	5	a not in range
WR2	5	-1	5	b not in range
WR3	5	5	-1	c not in range
WR4	201	5	5	a not in range
WR5	5	201	5	b not in range
WR6	5	5	201	c not in range

Triangle – input equivalence classes

$D1 = \{ a,b,c : 1..200 \mid a = b = c \bullet \langle a,b,c \rangle \}$

$D2 = \{ a,b,c : 1..200 \mid a = b, a \neq c \bullet \langle a,b,c \rangle \}$

$D3 = \{ a,b,c : 1..200 \mid a = c, a \neq b \bullet \langle a,b,c \rangle \}$

$D4 = \{ a,b,c : 1..200 \mid b = c, a \neq b \bullet \langle a,b,c \rangle \}$

$D5 = \{ a,b,c : 1..200 \mid a \neq b, a \neq c, b \neq c \bullet \langle a,b,c \rangle \}$

$D6 = \{ a,b,c : 1..200 \mid a \geq b+c \bullet \langle a,b,c \rangle \}$

$D7 = \{ a,b,c : 1..200 \mid b \geq a+c \bullet \langle a,b,c \rangle \}$

$D8 = \{ a,b,c : 1..200 \mid c \geq a+b \bullet \langle a,b,c \rangle \}$

What are the number of test cases for

- **weak-normal?**
- **strong-normal?**
- **weak-robust?**
- **strong-robust?**

Is this a good set of equivalence classes to use or is there a problem?

NextDate – naive equivalence classes

M1 = { month : 1 .. 12 }

D1 = { day : 1 .. 31 }

Y1 = { year : 1812 .. 2012 }

Invalid data

M2 = { month : Integer | month < 1 }

M3 = { month : Integer | month > 12 }

D2 = { day : Integer | day < 1 }

D3 = { day : Integer | day > 31 }

Y2 = { year : Integer | year < 1812 }

Y3 = { year : Integer | year > 2012 }

What are the number of test cases for

- **weak-normal?**
- **strong-normal?**
- **weak-robust?**
- **strong-robust?**

What is the problem with using these equivalence classes?

NextDate – improved equivalence classes

M1 = {month : 1 .. 12 | days(month) = 30 }

M2 = {month : 1 .. 12 | days(month) = 31 }

M3 = {month : {2} }

D1 = {day : 1 .. 28}

D2 = {day : {29} }

D3 = {day : {30} }

D4 = {day : {31} }

Y1 = {year : {2000} }

Y2 = {year : 1812 .. 2012 | leap_year (year) ∧ year ≠ 2000 }

Y3 = {year : 1812 .. 2012 | common_year (year) }

What is good and bad with using these equivalence classes?

Weak Normal Test Cases

Test Case	Month	Day	Year	Expected Output
WN1	6	14	1900	6/15/1900
WN2	7	29	1996	7/30/1996
WN3	2	30	2002	Invalid input date
WN4	6	31	1900	Invalid input date

NextDate strong test cases

- **What are the number of test cases for strong-normal testing?**
- **What are the number of test cases for strong-robust testing?**

NextDate strong test cases – 2

- There are 36 strong-normal test cases (3 x 4 x 3)
- Some redundancy creeps in
  - Testing February 30 and 31 for three different types of years seems unlikely to reveal errors
- There are 150 strong-robust test cases (5 x 6 x 5)

Commission problem – input classes

L1 = {locks : 1 .. 70 }

L2 = {locks : { -1 } }

S1 = {stocks : 1 .. 80 }

B1 = {barrels : 1 .. 90}

Invalid data

L3 = {locks : Integer | locks ≤ 0 ∧ locks ≠ -1}

L4 = {locks : Integer | locks > 70 }

S2 = {stocks : Integer | stocks < 1 }

S3 = {stocks : Integer | stocks > 80 }

B2 = {barrels : Integer | barrels < 1 }

B3 = {barrels : Integer | barrels > 90 }

What are the number of test cases for

- **weak-normal?**
- **strong-normal?**
- **weak-robust?**
- **strong-robust?**

What is good and not good about using these classes?

Commission problem – output classes

Sales = 45 × locks + 30 × stocks + 25 × barrels

S1 = {sales : 0 .. 1000 }

S2 = {sales : 1001 .. 1800 }

S3 = {sales : Integer | sales > 1800 }

Invalid data

S4 = {sales : Integer | sales < 0}

What are the number of test cases for

- **weak-normal?**
- **strong-normal?**
- **weak-robust?**
- **strong-robust?**

What is good and not good about using these classes?

Figure 5.6, page 84 shows the classes pictorially

Guidelines and observations

- Equivalence Class Testing is appropriate when input data is defined in terms of intervals and sets of discrete values.
- Equivalence Class Testing is strengthened when combined with Boundary Value Testing
- Strong equivalence makes the presumption that variables are independent.
  - If that is not the case, redundant test cases may be generated

Guidelines and observations – 2

- Complex functions, such as the NextDate program, are well-suited for Equivalence Class Testing

- Several tries may be required before the "right" equivalence relation is discovered
  - If the equivalence classes are chosen wisely, the potential redundancy among test cases is greatly reduced.
  - The key point in equivalence class testing is the choice of the equivalence relation that determines the classes.