Equivalence Class Testing

Garreth Davies

CS 339 Advanced Topics In Computer Science – Testing

(Prof. Schlingloff & Dr. M Roggenbach)

Contents

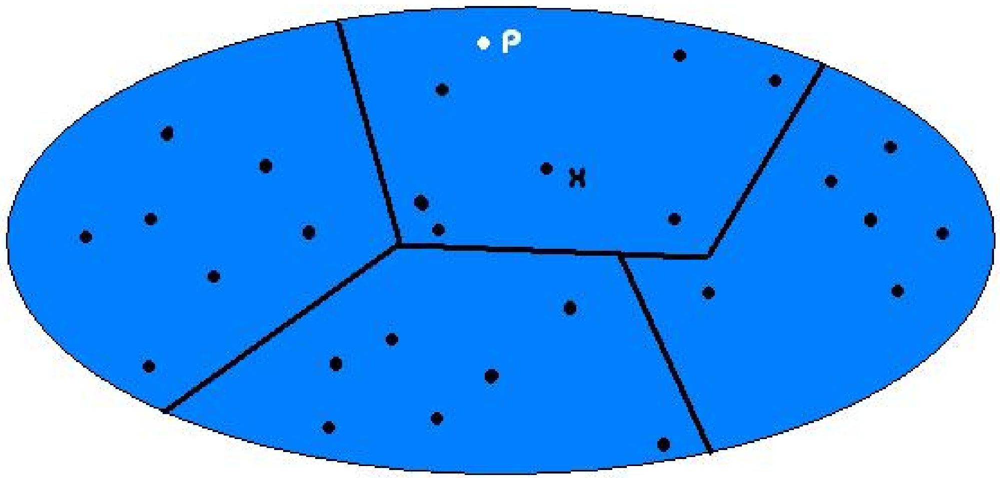
- Equivalence Class Testing
  - What is it?
  - Different forms of equivalence class testing
- Examples
  - Triangle Problem
  - Next Date Function Problem
- Testing Properties
  - Testing Effort
  - Guidelines & Observations

Equivalence Class Testing

What is Equivalence Class Testing?

- The next step from boundary value testing
- Motivation of Equivalence class testing
- Robustness
- Single/Multiple fault assumption

What is Equivalence Class Testing?



- What is an equivalence class?
- Completeness and Non-redundancy
- Equivalence Relation
- Greatly Reduces Redundancy

Weak & Normal Equivalence Class Testing

- A function F, of two variables x_1 and x_2
- x_1 and x_2 have the following boundaries and intervals within boundaries:

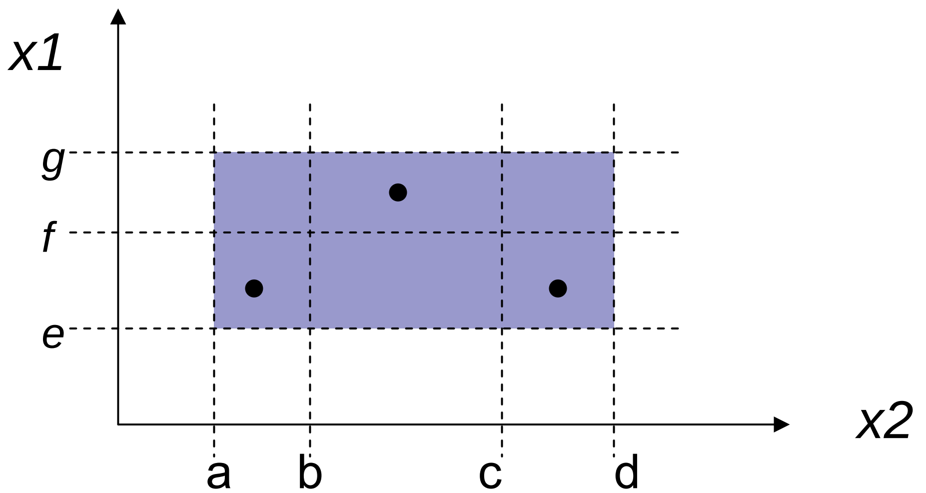
  $a \leq x_1 \leq d$, with intervals [a, b) [b, c), [c, d)

  $e \leq x_2 \leq g$, with intervals [e, f) [f, g)

- [= closed interval, (= open interval

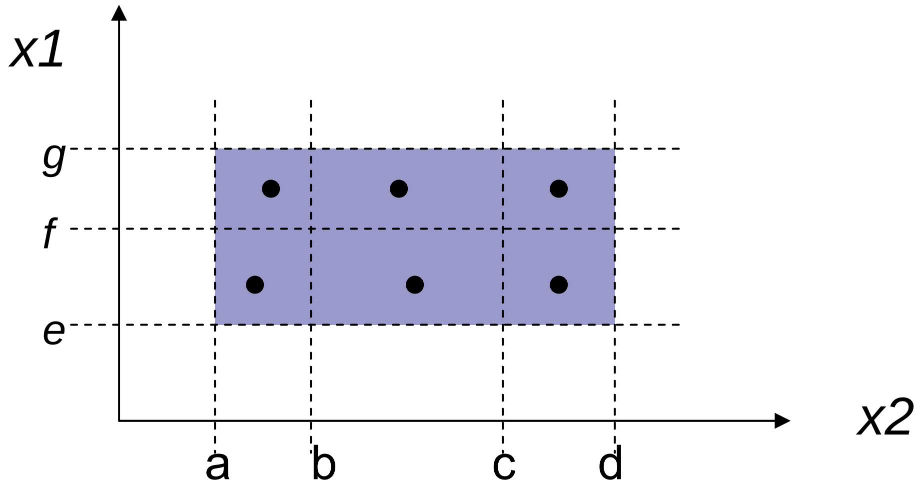
Weak Normal Equivalence Class Testing

- One variable from each equivalence class
- Values identified in systematic way



Strong Normal Equivalence Class Testing

- Test cases taken from each element of Cartesian product of the equivalence classes.
- Cartesian product guarantees notion of completeness.

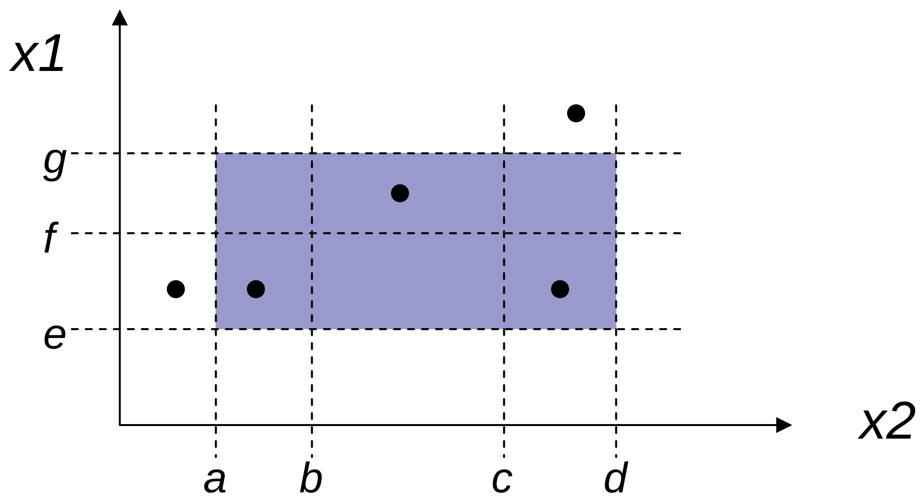


Robust Equivalence Class Testing

- Robust - consideration of invalid values.
- Two problems with robust ECT
  - Specification (expected output for invalid TC?)
  - Strongly typed languages (eliminate need), Traditional equivalence class testing (FORTRAN, COBAL) – errors common

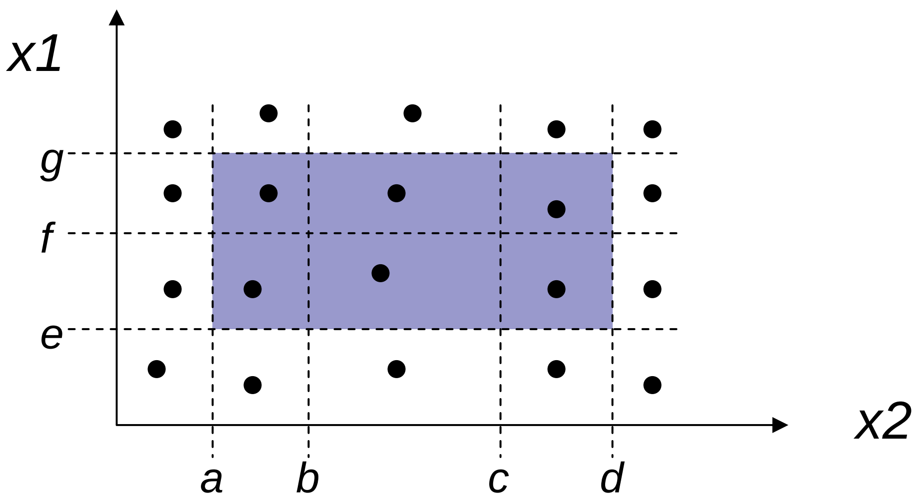
Weak Robust Equivalence Class Testing

- Valid inputs – weak normal ECT
- Invalid inputs – each TC has one invalid value, single fault should cause failure.



Strong Robust Equivalence Class Testing

- Combination of both robust and strong



Examples
Triangle Problem

Triangle Problem

- Four possible outputs – NotA-Triangle, Scalene, Isosceles and Equilateral.

R1 = {<a,b,c> : the triangle with sides a.b and c is equilateral}
R2 = {<a,b,c> : the triangle with sides a,b and c is isosceles}
R3 = {<a,b,c> : the triangle with sides a,b and c is isosceles}
R4 = {<a,b,c> : sides a,b and c do not form a triangle}

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 Problem

- Weak robust Equivalence Class Test Cases

Test Case	a	b	c	Expected Output
WR1	-1	5	5	Value of a is not in the range of permitted values
WR2	5	-1	5	Value of b is not in the range of permitted values
WR3	5	5	-1	Value of c is not in the range of permitted values
WR4	201	5	5	Value of a is not in the range of permitted values
WR5	5	201	5	Value of b is not in the range of permitted values
WR6	5	5	201	Value of c is not in the range of permitted values

Triangle Problem

- Here is one “corner” of the cube in 3-space of the additional strong robust equivalence class test cases.

Test Case	a	b	c	Expected Output
SR1	-1	5	5	Value of a is not in the range of permitted values
SR2	5	-1	5	Value of b is not in the range of permitted values
SR3	5	5	-1	Value of c is not in the range of permitted values
SR4	-1	-1	5	Values of a, b are not in the range of permitted values
SR5	5	-1	-1	Values of b, c are not in the range of permitted values
SR6	-1	5	-1	Values of a, c are not in the range of permitted values
SR7	-1	-1	-1	Values of a, b, c are not in the range of permitted values

Triangle Problem

- $D1 = \{\langle a, b, c \rangle : a = b = c\}$
- $D2 = \{\langle a, b, c \rangle : a = b, a \neq c\}$
- $D3 = \{\langle a, b, c \rangle : a = c, a \neq b\}$
- $D4 = \{\langle a, b, c \rangle : b = c, a \neq b\}$
- $D5 = \{\langle a, b, c \rangle : a \neq b, a \neq c, b \neq c\}$

- As separate question, constitute triangle? $\langle 1, 4, 1 \rangle$
- $D6 = \{\langle a, b, c \rangle : a \geq b + c\}$
- $D7 = \{\langle a, b, c \rangle : b \geq a + c\}$
- $D8 = \{\langle a, b, c \rangle : c \geq a + b\}$

Examples
Next Date Function Problem

Next Date Function Problem

- Valid Equivalence Classes

M1 = { month : $1 \leq$ month ≤ 12 }
D1 = { day: $1 \leq$ day ≤ 31 }
Y1 = { year: $1812 \leq$ year ≤ 2012 }

- Invalid Equivalence Classes

M2 = { month : month < 1 }
M3 = { month : month > 12 }
D2 = { day: day < 1 }
D3 = { day: day > 31 }
Y2 = { year: year < 1812 }
Y3 = { year: year > 2012 }

Next Date Function Problem

- Valid classes = Independent variables
- One weak and strong normal ECT.

Day	Month	Year	Expected Output
15	6	1912	16/6/1912

Next Date Function Problem

- Weak Robust Test Cases

Day	Month	Year	Expected Output
15	6	1912	16/6/1912
-1	6	1912	day not in range
32	6	1912	day not in range
15	-1	1912	month not in range
15	13	1912	month not in range
15	6	1811	year not in range
15	6	2013	year not in range

Next Date Function Problem

- Strong robust ECT

Test Case	Month	Day	Year	Expected Output
SR1	-1	15	1912	Value of month not in the range 1..12
SR2	6	-1	1912	Value of day not in the range 1..31
SR3	6	15	1811	Value of year not in the range 1812..2012
SR4	-1	-1	1912	Value of month not in the range 1..12
				Value of day not in the range 1..31
SR5	6	-1	1811	Value of day not in the range 1..31
				Value of year not in the range 1812..2012
SR6	-1	15	1811	Value of month not in the range 1..12
				Value of year not in the range 1812..2012
SR7	-1	-1	1811	Value of month not in the range 1..12
				Value of day not in the range 1..31
				Value of year not in the range 1812..2012

Next Date Function Problem

- Previous test cases were poor.
- Focus on Equivalence Relation.
- What must be done to an input date?
- We produce a new set of Equivalence Classes.

Next Date Function Problem

- M1 = { month: month has 30 days }
- M2 = { month: month has 31 days }
- M3 = { month: month is February }
- D1 = { day: $1 \leq$ day ≤ 28 }
- D2 = { day: day = 29 }
- D3 = { day: day = 30 }
- D4 = { day: day = 31 }
- Y1 = { year: year = 2000 }
- Y2 = { year: year is a leap year }
- Y3 = { year: year is a common year }

- Simplify the question of the last day of the month.

Next Date Function Problem

- Weak normal ECT
- Mechanical selection & automatic test Generation

Day	Month	Year	Expected Output
14	6	2000	15/6/2000
29	7	1996	30/7/1996
30	2	2002	impossible date
31	6	2000	impossible input date

Next Date Function Problem

- Strong normal ECT

Day	Month	Year	Expected Output
14	6	2000	15/6/2000
14	6	1996	15/6/1996
14	6	2002	14/6/2002
29	6	2000	30/6/2000
29	6	1996	30/6/1996
29	6	2002	30/6/2002
30	6	2000	1/7/2000
30	6	1996	1/7/1996
...	...	...	...
30	2	2002	impossible date
31	2	2000	impossible date
31	2	1996	impossible date
31	6	2002	impossible date

3 month classes*4 day classes*3 year classes = 36 ECT.

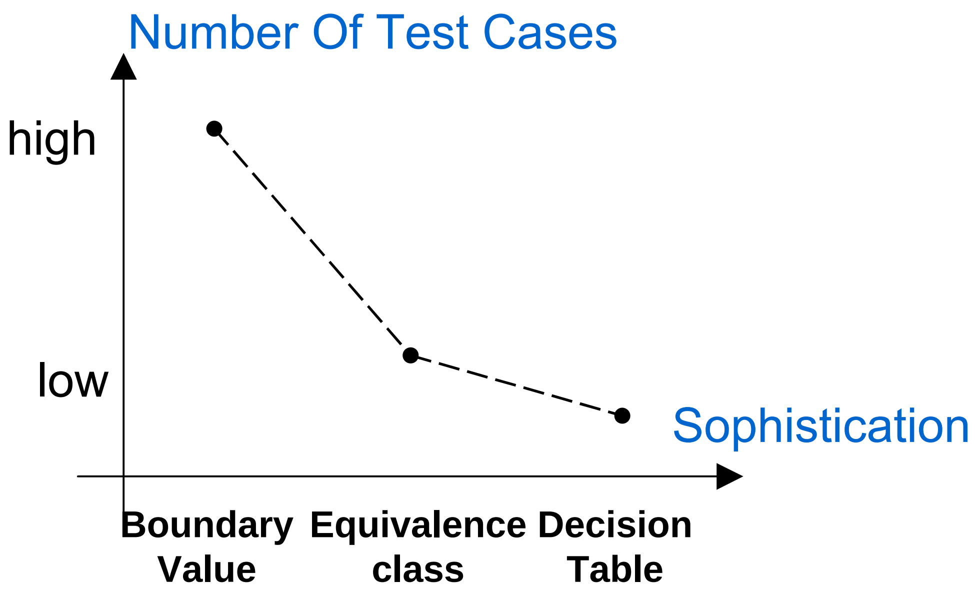
Next Date Function Problem

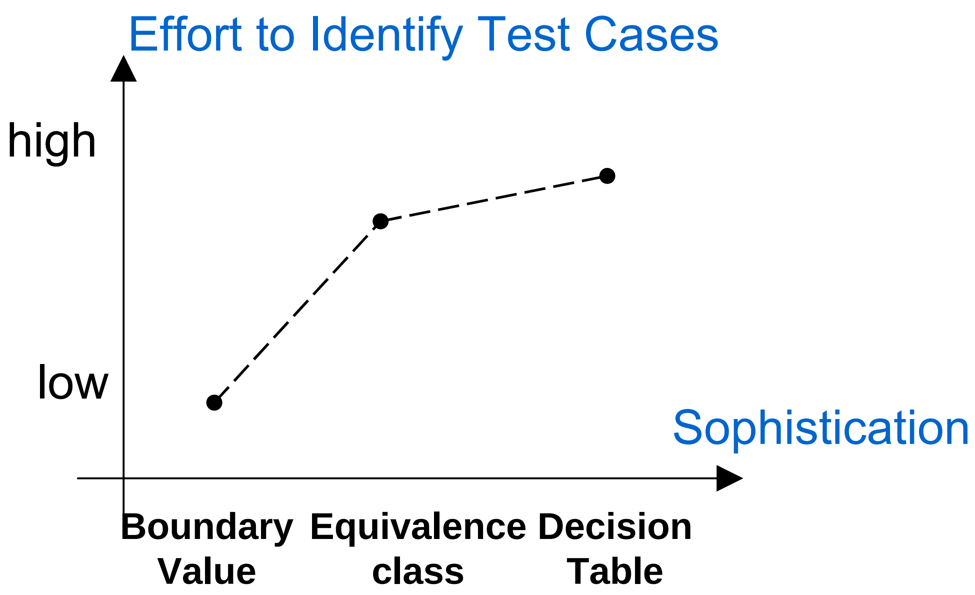
- Moving from weak to strong ECT.
- Independence resulting in the cross product.
- Adding two invalid classes for each variable = 150 robust equivalence class test cases!

Testing Properties

Testing Effort

- EC techniques pay attention to the function itself. More thought required.
- Trade off between test identification effort and test execution effort.





Guidelines & Observations

- Implementation language strongly typed, no need for robust forms.
- ECT is appropriate to certain data input.
- Complex functions help identify useful EC, illustrated by next date function problem.
- Several attempts may be needed.

Summary

- Equivalence Class Testing improves on boundary value testing
- Equivalence Relation is key to producing useful test cases
- Equivalence Class Testing can be succeeded

Questions?

Case Study

- This example lets us compare functional testing methods.
- Insurance company computes the semi-annual car insurance premium based on:

Premium = BaseRate*ageMultiplier – safeDrivingReduction

- Safe driving reduction is given when the current points on driver’s license are below an age related cut-off

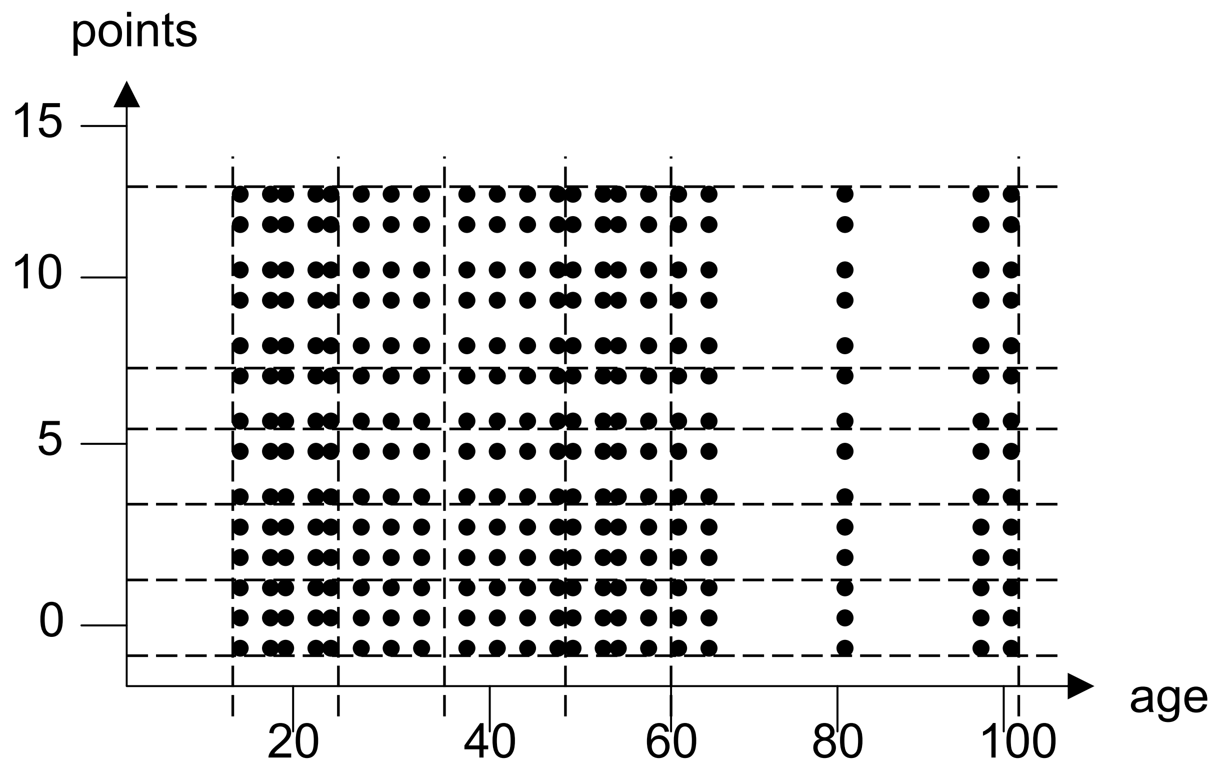
Case Study

- Using the following intervals we can show the difference between worst case boundary-value testing and equivalence class testing.

$A1 = \{ age : 16 \leq age < 25 \}$
$A2 = \{ age : 25 \leq age < 35 \}$
$A3 = \{ age : 35 \leq age < 45 \}$
$A4 = \{ age : 45 \leq age < 60 \}$
$A5 = \{ age : 60 \leq age < 100 \}$
$P1 = \{ points = 0, 1 \}$
$P2 = \{ points = 2, 3 \}$
$P3 = \{ points = 4, 5 \}$
$P4 = \{ points = 6, 7 \}$
$P5 = \{ points = 8, 9, 10, 11, 12 \}$

Case Study

- Worst-case boundary value testing.
- 273 worst-case boundary value test cases! Redundancy.



Case Study

- Equivalence class testing clearly reduces redundancy.
- Why test point classes P2-P5 for A1?
- Decision table testing improves on this.