EECS 4313

Software Engineering Testing

Topic 05:

Equivalence Class Testing

Zhen Ming (Jack) Jiang

Relevant Readings

- [Jorgensen] chapter 6

Introduction

- Boundary Value Testing derives test cases with
  - Massive redundancy
  - Serious gaps
- Equivalence Class Testing attempts to alleviate these problems
- Two orthogonal dimensions
  - Robustness
  - Single/Multiple Fault Assumption

Equivalence Class Testing

- Partition the set of all test cases into mutually disjoint subsets whose union is the entire set
- Choose one test case from each subset
- Two important implications for testing:
  1. The fact that the entire set is represented provides a form of completeness
  2. The disjointness assures a form of non-redundancy

Equivalence Class Selection

- 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.
- We will differentiate below, between four different types of equivalence class testing (ECT).
  - Weak normal ECT
  - Strong normal ECT
  - Weak robust ECT
  - Strong robust ECT

Assumptions regarding ECT

- Equivalence Class Testing is appropriate when the system under test can be expressed as a function of one or more variables, whose domains have well defined intervals
- Input and/or output variables have well-defined intervals
  - For a two-variable function $F(x_1,x_2)$

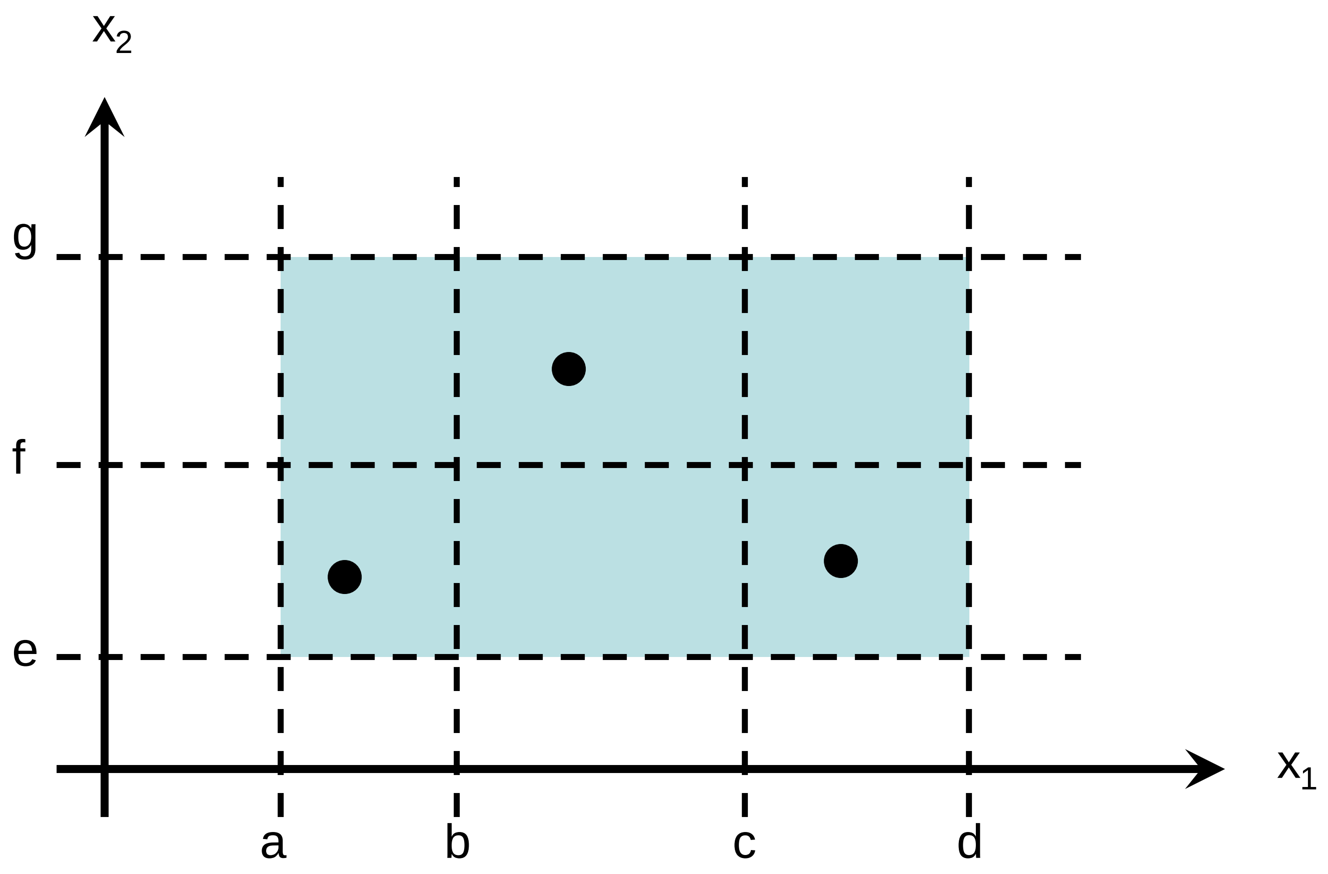
    $a \leq x_1 \leq d$, with intervals $[a,b)$, $[b,c)$, $[c,d]$

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

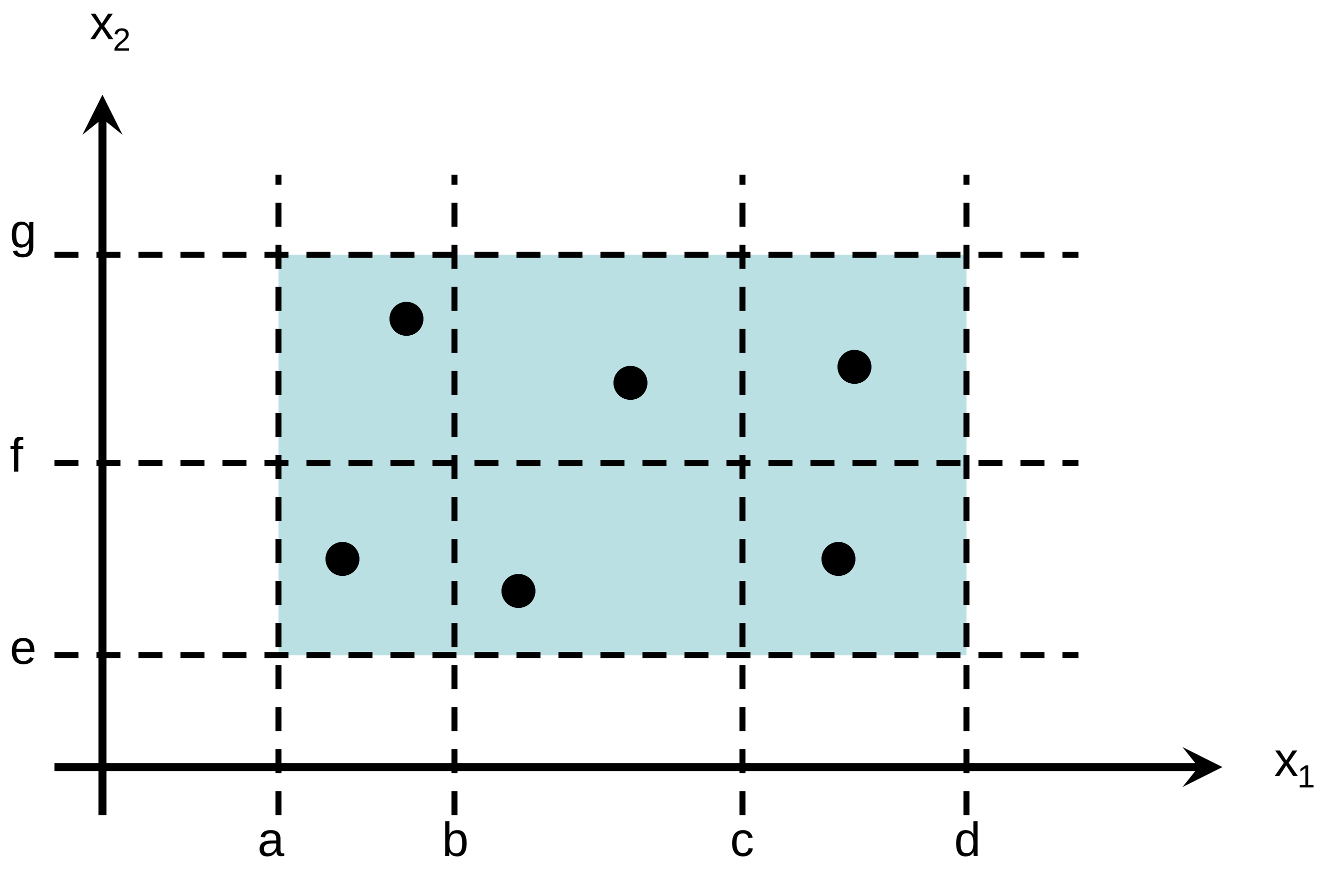
More assumptions regarding ECT

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

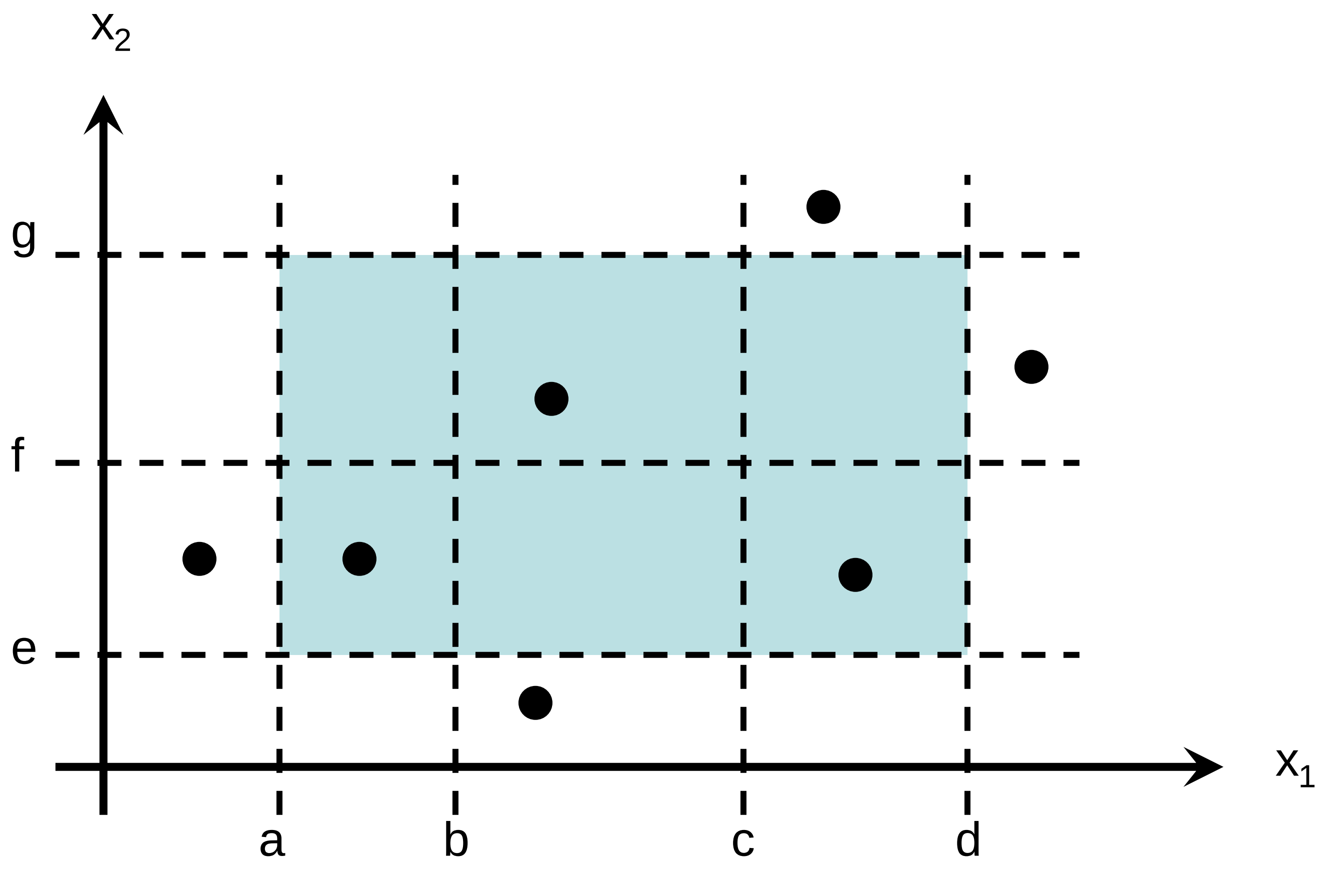
Weak Normal ECT



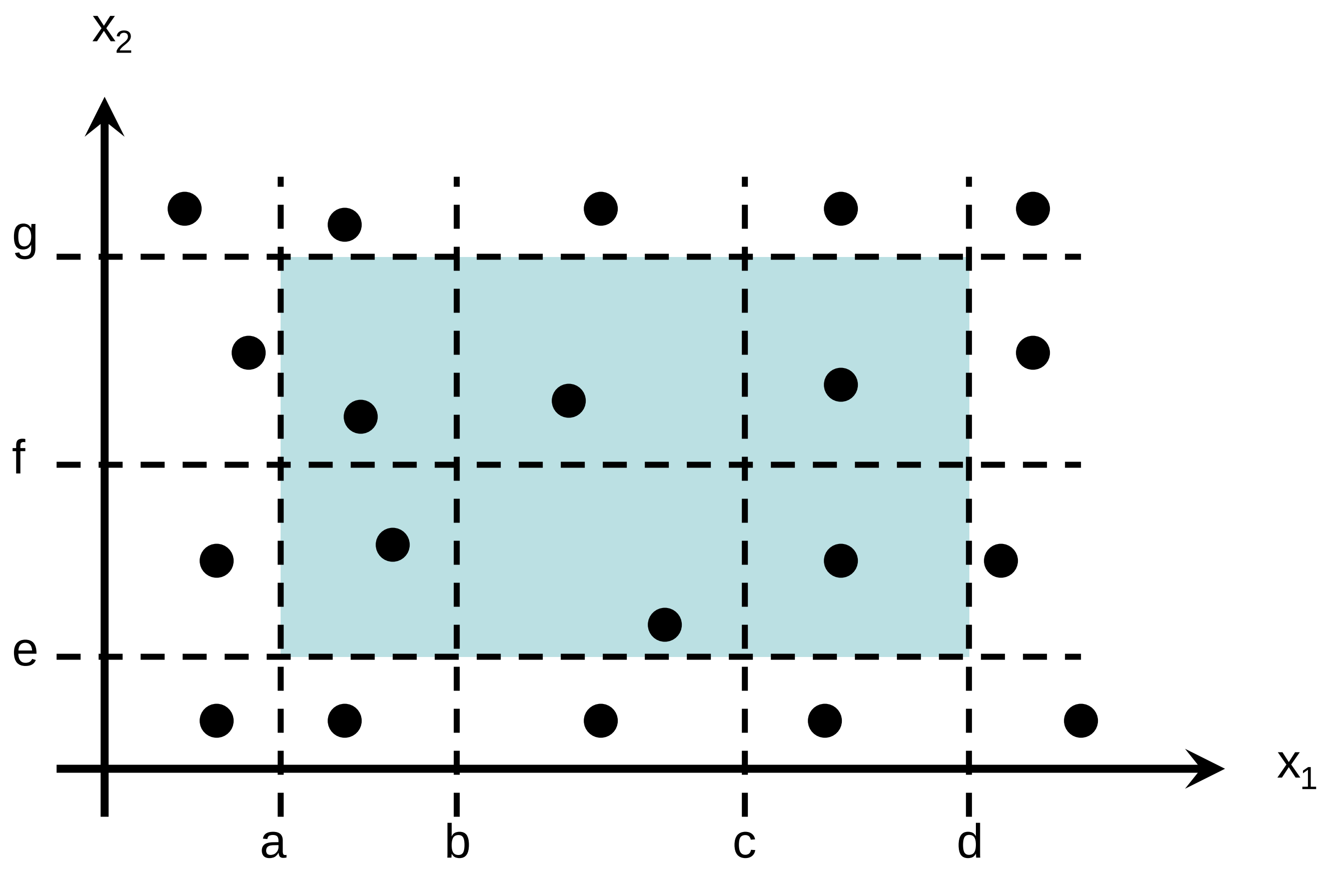
Strong Normal ECT



Weak Robust ECT



Strong Robust ECT



Limitations of ECT

- 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

Equivalence Class Test Cases for the Triangle Problem

- Four possible outputs:
  - Not a Triangle, Isosceles, Equilateral, Scalene
- We can use these to identify output (range) equivalence classes:

  R1= {the triangle with sides a, b, c, is equilateral}

  R2= {the triangle with sides a, b, c, is isosceles}

  R3= {the triangle with sides a, b, c, is scalene}

  R4= {sides a, b, c do not form a 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

Since there are no valid subintervals of variables a, b and c exist, the strong normal equivalence class test cases are identical

Weak Robust ECT - Additional Test 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

a, b and c are all in [1, 200]

Take home exercise

- What is the strong robust ECT for the triangle problem?

Input equivalence classes

$D1 = \{\langle a,b,c \rangle \mid a = b = c\}$

$D2 = \{\langle a,b,c \rangle \mid a = b,\ a \neq c\}$

$D3 = \{\langle a,b,c \rangle \mid a = c,\ a \neq b\}$

$D4 = \{\langle a,b,c \rangle \mid b = c,\ a \neq b\}$

$D5 = \{\langle a,b,c \rangle \mid a \neq b,\ a \neq c,\ b \neq c\}$

$D6 = \{\langle a,b,c \rangle \mid a \geq b+c\}$

$D7 = \{\langle a,b,c \rangle \mid b \geq a+c\}$

$D8 = \{\langle a,b,c \rangle \mid c \geq a+b\}$

The NextDate Problem

- NextDate is a function of three variables: month, date, and year. It returns the date of the day after the input date. The month, date, and year variables have integer values subject to these conditions (the year range ending in 2012 is arbitrary):

  (C1) $1 \le \text{month} \le 12$

  (C2) $1 \le \text{day} \le 31$

  (C3) $1812 \le \text{year} \le 2012$

- As we did with the triangle program, we can make our problem statement more specific. This entails defining responses for invalid values of the input values for the day, month, and year. We can also define responses for invalid combinations of inputs, such as June 31 of any year. If any of conditions c1, c2, or c3 fails, NextDate produces an output indicating the corresponding variable has an out-of-range value—for example, “Value of month not in the range 1...12.” Because numerous invalid day–month–year combinations exist, NextDate collapses these into one message: “Invalid Input Date.”

NextDate Equivalence Classes

M1= {month | month has 30 days}
M2= {month | month has 31 days}
M3= {month | month is February}
D1= {day | 1 ≤ day ≤ 28}
D2= {day | day = 29}
D3= {day | day = 30}
D4= {day | day=31}
Y1= {year | year = 1900}
Y2= {year | year is a leap year}
Y3= {year | year is a common year}

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 discussion

- 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

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 takes the presumption that variables are independent. If that is not the case, redundant test cases may be generated
  - e.g., February 30 and 31 for three different types of years can be redundant test cases

Guidelines and observations

- 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