Equivalence Class Testing

Introduction

Use of Equivalence Classes as the basis for functional testing has 2 motivations:

- ❑ Have a sense of complete testing
- ❑ Avoid redundancy

Ex: In triangle problem (5,5,5) is a test case for equilateral triangle.

Testing again with (6,6,6) or (10,10,10) makes no sense!

Equivalence Class Test Cases for NextDate function

Based on valid values, the equivalence classes are:

M1= {month: 1 <= month <=12}

D1 = {day: 1 <=day<=31}

Y1= {year: 1812 <= year <= 2050}

And the invalid equivalence classes are:

M2 = {month : month < 1}

M3 = {month : month > 12}

D2 = {day : day < 1}

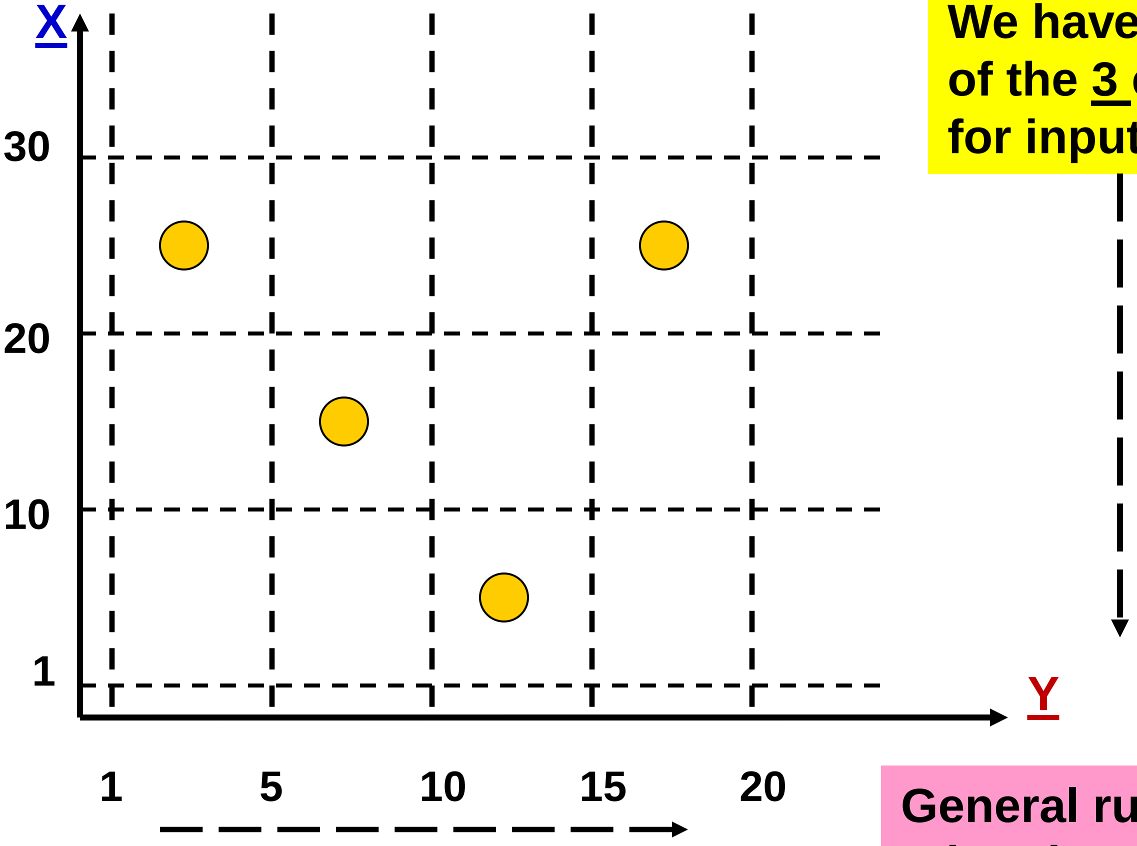
D3 = {day : day > 31}

Y2 = {year: year < 1812}

Y3 = {year : year > 2050}

Example of :Weak Normal Equivalence testing

Assume the equivalence partitioning of input X is: 1 to 10; 11 to 20, 21 to 30
and the equivalence partitioning of input Y is: 1 to 5; 6 to 10; 11 to15; and 16 to 20



We have covered everyone of the 3 equivalence classes for input X.

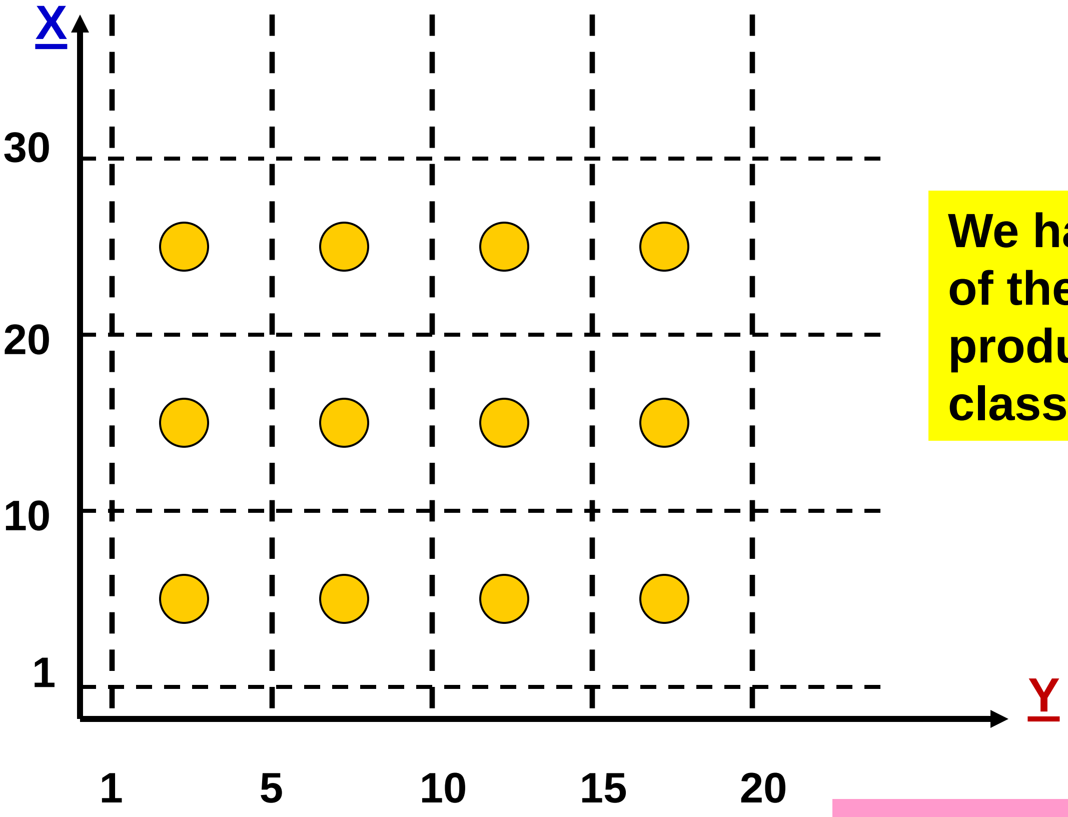
For (x, y)
we have:
(24, 2)
(15, 8)
(4, 13)
(23, 17)

We have covered each of the 4 equivalence classes for input Y.

General rule for # of test cases?
What do you think?
# of partitions of the largest set?

Example of : Strong Normal Equivalence testing

Assume the equivalence partitioning of input X is: 1 to 10; 11 to 20, 21 to 30 and the equivalence partitioning of input Y is: 1 to 5; 6 to 10; 11;15; and 16 to 20



We have covered everyone of the 3 x 4 Cartesian product of equivalence classes

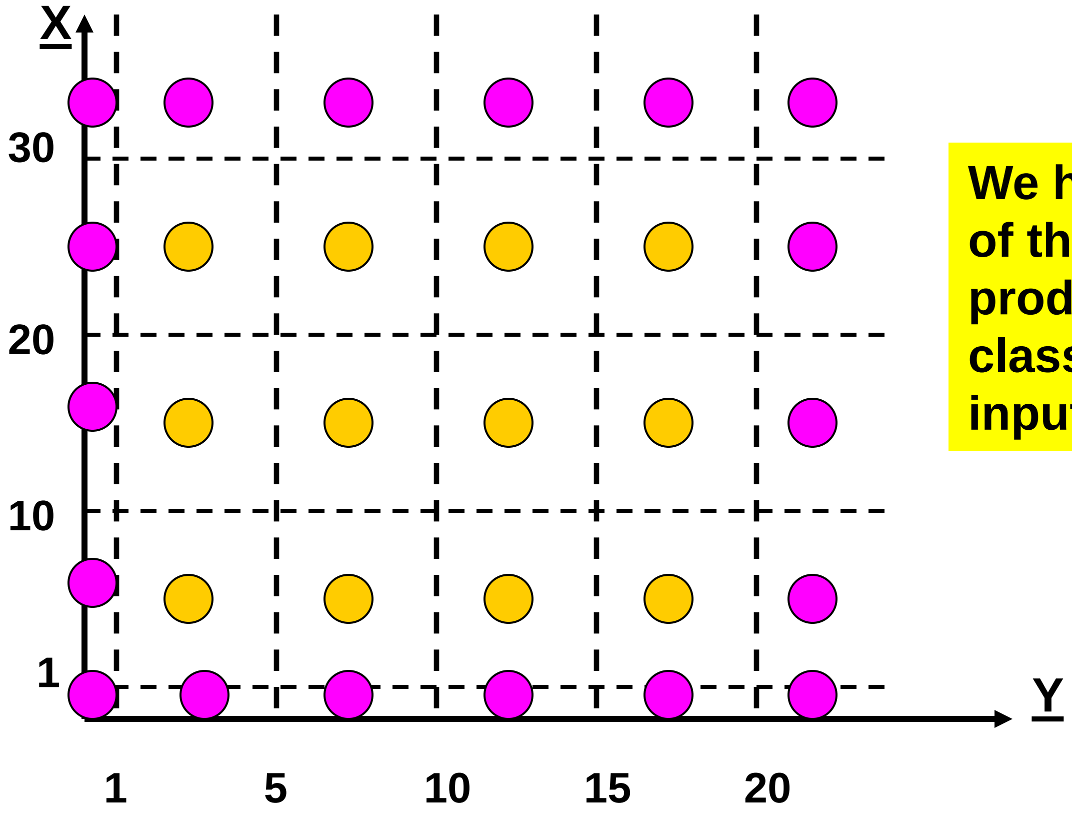
General rule for # of test cases? What do you think?

a. Weak Normal and Strong Normal EC test case

Test Case ID	Month (mm)	Day (dd)	Year (yyyy)	Expected Output
WN1, SN1	6	15	1912	6/16/1912

Example of : Strong Robust Equivalence testing

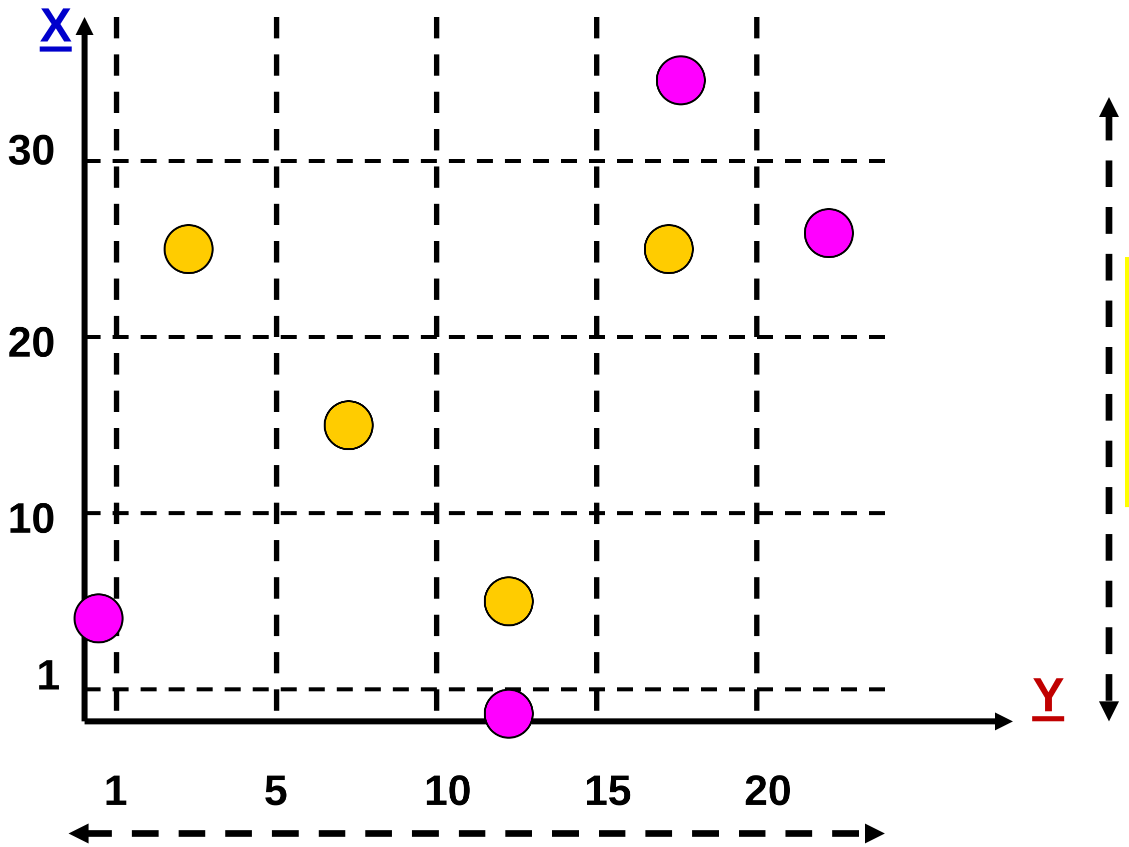
Assume the equivalence partitioning of input X is: 1 to 10; 11 to 20, 21 to 30 and the equivalence partitioning of input Y is: 1 to 5; 6 to 10; 11;15; and 16 to 20



We have covered everyone of the 5 x 6 Cartesian product of equivalence classes (including invalid inputs)

Example of :Weak Robust Equivalence testing

Assume the equivalence partitioning of input X is 1 to 10; 11 to 20, 21 to 30 and the equivalence partitioning of input Y is 1 to 5; 6 to 10; 11;15; and 16 to 20



We have covered everyone of the 5 equivalence classes for input X.

We have covered each of the 6 equivalence classes for input Y.

b. Weak Robust EC Test Cases

Test Case ID	Month (mm)	Day (dd)	Year (yyyy)	Expected Output
WR1	6	15	1912	6/16/1912
WR 2	-1	15	1912	Invalid Value of Month, as Month cannot be -ve
WR 3	13	15	1912	Invalid Value of Month, as Month is always < 12
WR 4	6	-1	1912	Invalid Value of Day, as Day cannot be -ve
WR 5	6	32	1912	Invalid Value of Day, as we cannot have 32 days in any month
WR 6	6	15	1811	Invalid Value of Year, as the rage is 1812 to 2012 only
WR 7	6	15	2013	Invalid Value of Year

c. Strong Robust EC test cases

- Multiple fault assumption.
- Robust- Test along with invalid inputs.
- Since there are 3 partitions/ Classes for each input variables. The total number of test cases is

  (M1,M2,M3)x(D1,D2,D3)x(Y1,Y2,Y3)

  i.e. 3x3x3= 27 test cases.

Strong Robust EC test cases (contd..)

Test case Id	Day	Month	Year
SR1	D1	M1	Y1
SR2	D1	M1	Y2
SR3	D1	M1	Y3
SR4	D1	M2	Y1
SR5	D1	M2	y2
SR6	D1	M2	y3
SR7	D1	M3	Y1
SR8	D1	M3	Y2
SR9	D1	M3	Y3
SR10	D2	M1	Y1
SR11	D2	M1	Y2
SR12	D2	M1	Y3
SR13	D2	M2	Y1
SR14	D2	M2	Y2
SR15	D2	M2	Y3
SR16	D2	M3	Y1
SR17	D2	M3	Y2
SR18	D2	M3	Y3
SR19	D3	M1	Y1
SR20	D3	M1	Y2
SR21	D3	M1	Y3
SR22	D3	M2	Y1
SR23	D3	M2	Y2
SR24	D3	M2	Y3
SR25	D3	M3	Y1
SR26	D3	M3	Y2
SR27	D3	M3	Y3

Modified Equivalence Class for NextDate Problem

We need the modified classes as we know that at the end of a month

- The next day is 1 and the month is incremented.
- At the end of a year, both the day and the month are reset to 1 and the year is also incremented.
- Finally, the problem of leap year makes determining the last day of a month interesting.

With all the above in mind, we describe the following 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 = 2000}
Y2 = {year : year is a leap year}
Y3 = {year : year is a common year}

Weak Normal Equivalence Class

- The inputs are mechanically selected from the approximate middle of the corresponding class
- Based on single fault assumption.

Test Case ID	Month (mm)	Day (dd)	Year (yyyy)	Expected Output
WN1	6	14	2000	6/15/2000
WN2	7	29	1996	7/30/1996
WN3	2	30	2002	2/31/2002 (Impossible)
WN4	6	31	2000	7/1/2000 (Impossible)

Strong Normal Equivalence Class

- Strong- Multiple fault assumption
- Normal- Testing for valid set of data/ valid inputs
- There will be 36 test cases.

(M1,M2,M3)X(D1,D2,D3,D4)X(Y1,Y2,Y3)

i.e. 3x4x3 =36 test cases.

Strong Normal Equivalence Class

Test Case ID	Month (mm)	Day (dd)	Year (yyyy)
SN1	6	14	2000
SN2	6	14	1996
SN3	6	14	2002
SN4	6	29	2000
SN5	6	29	1996
SN6	6	29	2002
SN7	6	30	2000
SN8	6	30	1996
SN9	6	30	2002
SN10	6	31	2000
SN11	6	31	1996
SN12	6	31	2002
SN13	7	14	2000
SN14	7	14	1996
SN15	7	14	2002
SN16	7	29	2000
SN17	7	29	1990
SN18	7	29	2002
SN19	7	30	2000
SN20	7	30	1996
SN21	7	30	2002
SN22	7	31	2000
SN23	7	31	1996
SN24	7	31	2002
SN25	2	14	2000
SN26	2	14	1996
SN27	2	14	2002
SN28	2	29	2000
SN29	2	29	1996
SN30	2	29	2002
SN31	2	30	2000
SN32	2	30	1996
SN33	2	30	2002
SN34	2	31	2000
SN35	2	31	1996
SN36	2	31	2002

Weak Robust Equivalence Class

- For valid inputs use one value from each valid class
- For invalid inputs, a test case will have one invalid value (from input domain) and the remaining value will all be valid.

Strong Robust Equivalence Class

- Multiple Fault assumption
- Tests along with invalid inputs.
- We will have
  - Day- 6 classes
  - Month – 5 Classes
  - Year -5 classes
- Therefore, Total no. of test Cases will be

5x6x5 = 150

Triangle Problem

- **Simple version**: The triangle program accepts three integers, a, b, and c, as input. These are taken to be sides of a triangle. The output of the program is the type of triangle determined by the three sides: Equilateral, Isosceles, Scalene, or Not A Triangle.
- **Improved version:** “Simple version” plus better definition of inputs:

The integers a, b, and c must satisfy the following conditions:

- c1. $1 \le a \le 200$ c4. $a < b + c$
- c2. $1 \le b \le 200$ c5. $b < a + c$
- c3. $1 \le c \le 200$ c6. $c < a + b$

(contd..)

Final Version: "Improved version" plus better definition of outputs:

If an input value fails any of conditions c1, c2, or c3, the program notes this with an output message, for example, "Value of b is not in the range of permitted values. "If values of a, b, and c satisfy conditions c1, c2, and c3, one of four mutually exclusive outputs is given:

□ If all three sides are equal, the program output is Equilateral.

□ If exactly one pair of sides is equal, the program output is Isosceles.

□ If no pair of sides is equal, the program output is Scalene.

□ If any of conditions c4, c5, and c6 is not met, the program output is NotATriangle.

Write all EC test cases for Triangle Problem

- Use output equivalence class
- 4 possible outputs : Equilateral, Isosceles, Scalene, and Not A Triangle
- 4 equivalence class :
    - 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 scalene}
    - R4 ={<a,b,c> : the triangle with sides a, b, and c do not form a triangle}

Weak normal

Test case id	a	b	c	Expected
1	5	5	5	Equilateral
2	2	2	3	Isosceles
3	3	4	5	Scalene
4	4	1	2	Not a triangle

Weak Robust Equivalence Class

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