Topic #2 –Equivalence Partitioning Test Method

Instructor: Jerry Gao, Ph.D., Professor
San Jose State University

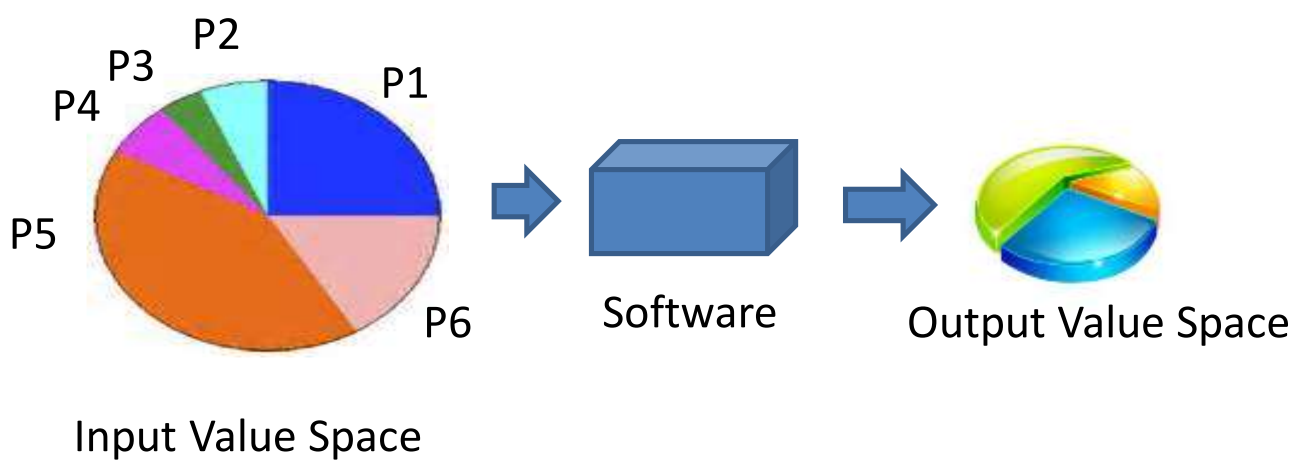
TOPIC #2 – EQUIVALENCE PARTITIONING TEST METHOD

- What is Equivalence Partitioning(EP) Testing?
- Why Do We Need the EP Test Method?
- How to Use the EP Test Method?
- Equivalence Partitioning Test Examples
- Equivalence Partitioning Summary

What is software equivalence partitioning (EP) method?

Definition: Equivalence Partitioning (EP) is a specification-based test technique. The idea behind this is to divide the input value space for a under-test program into a set of partitions (or classes). Each partition only needs one test input vector to cover it even though there are many others.

Equivalence partitions are also known as equivalence classes.

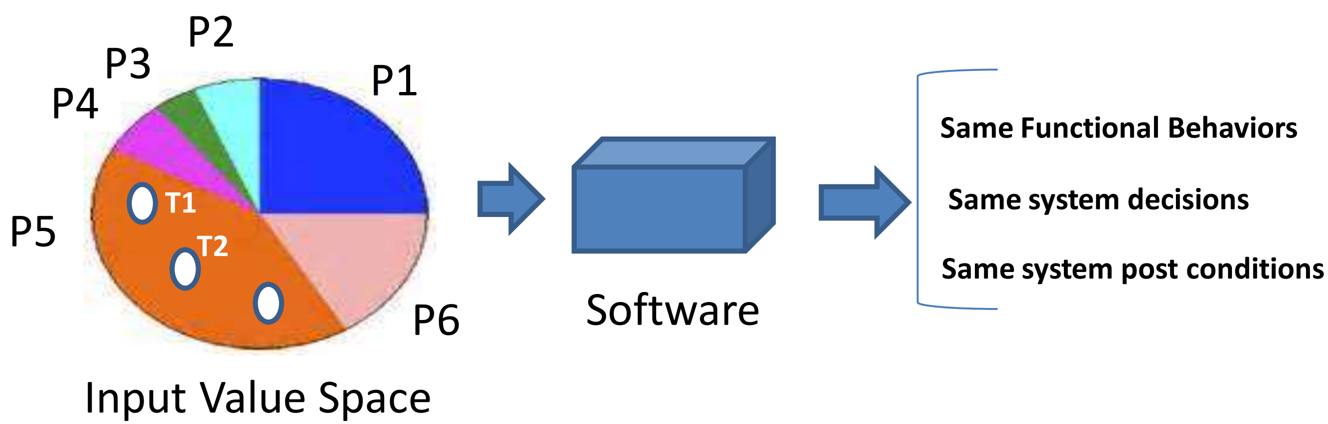


What is software equivalence partitioning (EP) method?

Assumptions of the EP method:

- An under-test software will demonstrate the same functional behavior for all tests of an EQ partition. Therefore, only one test is needed for each partition.

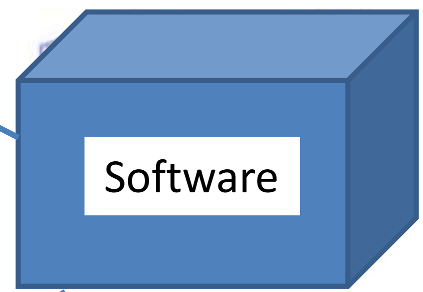
- For any test of an EQ partition (known as P), if an under-test software will demonstrate a failure (or an error), then the other tests of P will lead the software to produce an similar failure.



Why Do We Need the EP Method?

Answer: - Testing cost reduction by reducing tests for each input data.
- Achieve adequate test coverage criteria for EP classes

Exhaustive testing for every input Value is not possible.



Too many possible Input values for each Input data

Equivalence Partitioning Class Examples

- Different data value ranges of input parameters
- Option values for input data, such as dates, times, country names, ete.
- Different groups of invalid inputs for parameters
- Different groups of equivalent input/output events, such as phone signals
- Equivalent operating environments for an under-test system
- Equivalent classes of a third-party system's outputs as inputs
- Number of records in a database (or other equivalent objects
- Equivalent classes of input data values from users
- Equivalent resource types of a system

How to Use Equivalence Partitioning

- For the input value space of an under-test system, please do the following:

Step #1: Identify equivalent input value classes

- All EP classes must be independent, and not overlapping
- All EP classes must cover all of input value space
- EP classes must cover both invalid input values and valid input values

Step #2: Select one test from each EP class, and identify its expected output

An Equivalence Partitioning Example – A Banking Account

For a savings Bank account,

- 3% rate of interest is credited if its balance is in the range of $0 to $100
- 5% rate of interest is credited if its balance is in the range of over $100 to $1000
- 7% rate of interest is credited if its balance is is over $1000 and above

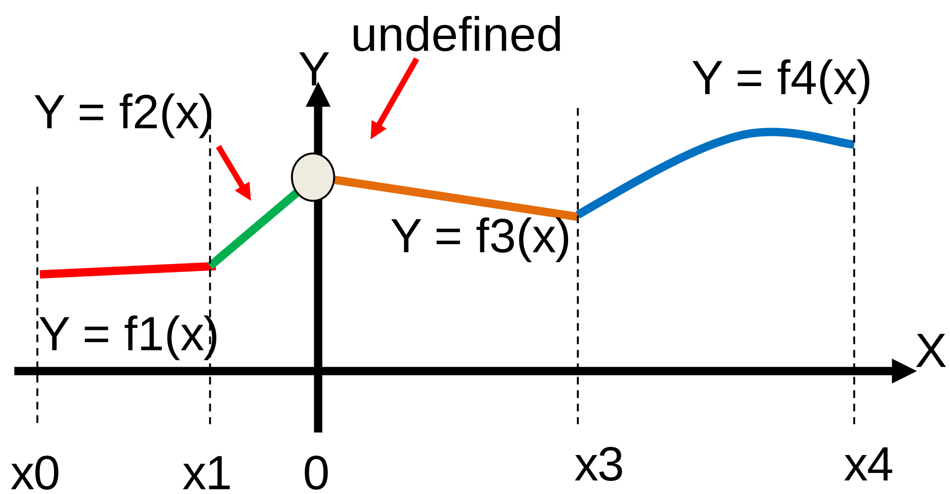
Please identify your EQ Partition classes?

	Partition 1	Partition 2		Partition 3		Partition 4
Account Balance	Invalid partition	Valid (for 3% interest)		Valid (for 5%)		Valid (for 7%)
	-$0.01	$0.00	$100.00	$100.01	$999.99	$1000.00

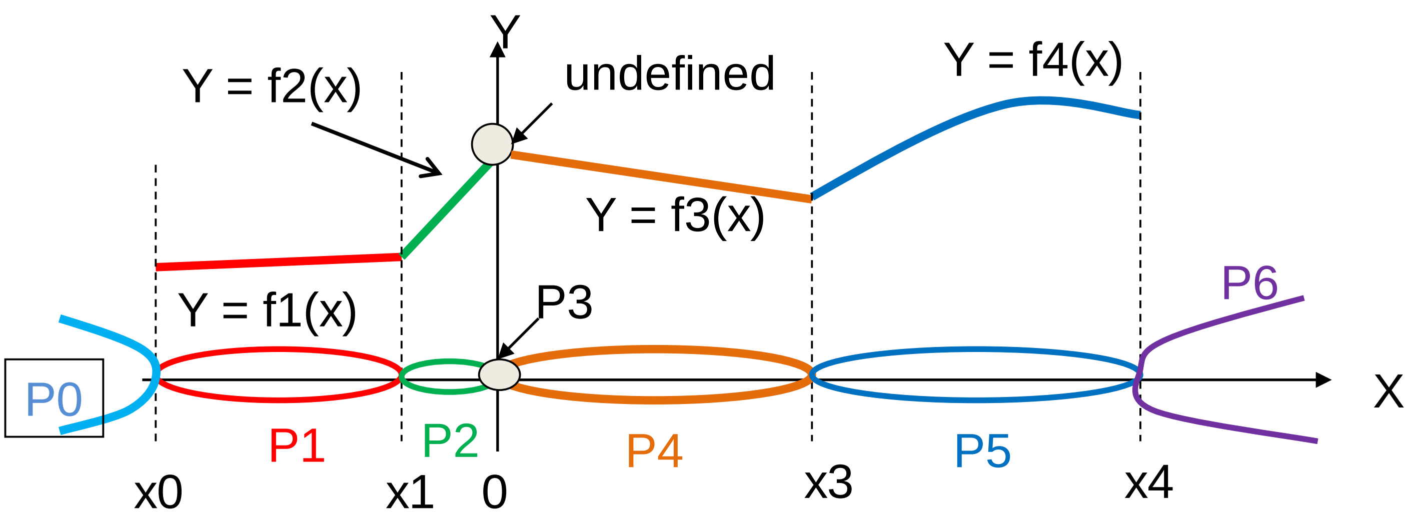
An Equivalence Partitioning Example

A software component provides a set of application functions.
These functions generate a corresponding Y value based on the value of input data X.
Each function is called and executed when X value is fall into a specific data range.

$$Y(x) = \begin{cases} f1(x) & x \text{ in } [x0, x1] \\ f2(x) & x \text{ in } (x1, 0) \\ \text{Undefined} & x = 0 \\ f3(x) & x \text{ in } (0, x3] \\ f4(x) & x \text{ in } (x3, x4] \end{cases}$$



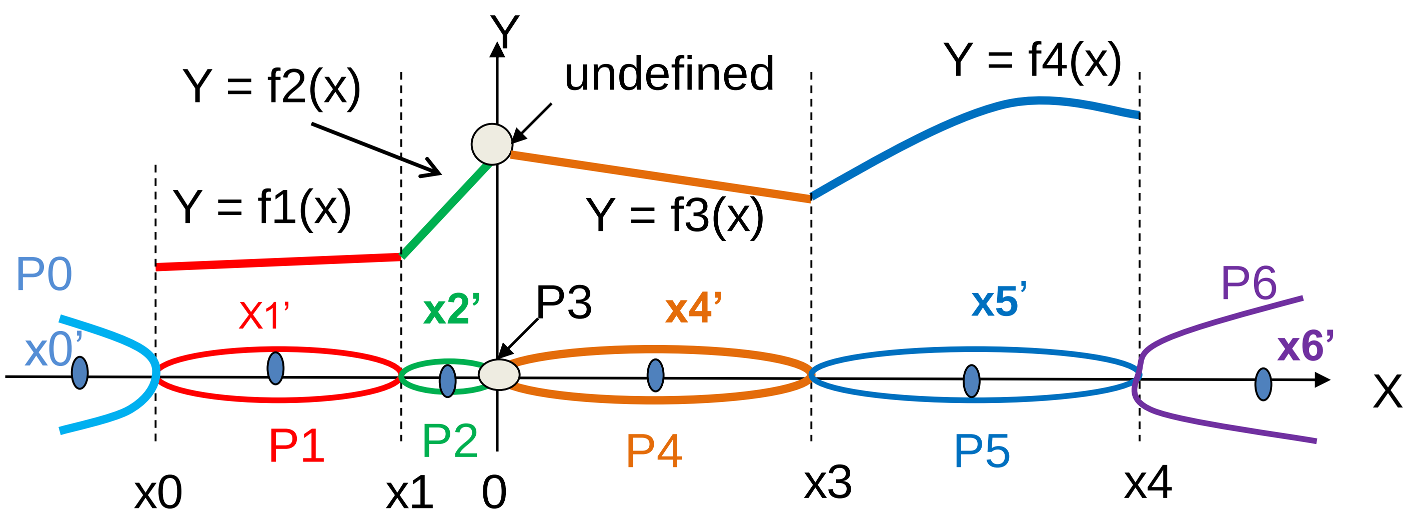
An Equivalence Partitioning Example



EQ Partitions:

P0: x < x0 or x in (x0, Very Small No.)
P1: x in [x0, x1]
P2: x in (x1, 0)
P3: x = 0
P4: x in (0, x3]
P5: x in (x3, x4]
P6: x > x4or x in (x4, Very Larger No.)

An Equivalence Partitioning Example



Test Cases for EQ Partitions:

x= x1',	y = f1(x1') = y1'
x = x2',	y = f2(x2') = y2'
x = 0,	y = undefined
x = x4',	y = f3(x4') = y3'
x = x5',	y = f4(x5') = y4'
x = x0',	y = Outside value boundary
x = x6',	y = Outside value boundary

Equivalence Partitioning Summary

Advantage:

- Simple to use
- Cost reduction for software testing by selecting one test for each EP class

Test Coverage:

Using Equivalence Partitioning, we can achieve EP class test coverage:

For each EP class (or partition), there must be at least one test.

Limitations:

Very difficult to apply when under-test software has many Input data parameters, and multi-dimension input value spaces.

Challenges:

- Lack of step-by-step tips to identify EP classes for a software
- Difficult to deal with a complicated input data set