Aidas Kasperavičius
IFM-0/2

Equivalence Partitioning

Equivalence partitioning

- Equivalence partitioning is a software testing technique that divides the input and/or output data of a software unit into partitions of data from which test cases can be derived.
- The equivalence partitions are usually derived from the requirements specification for input attributes that influence the processing of the test object.
- Test cases are designed to cover each partition at least once.

What can be found using equivalence partitioning?

- Equivalence partitioning technique uncovers classes of errors.
- Testing uncovers sets of inputs that causes errors or failures, not just individual inputs.

What can be partitioned?

- Usually it is the input data that is partitioned.
- However, depending on the software unit to be tested, output data can be partitioned as well.
- Each partition shall contain a set or range of values, chosen such that all the values can reasonably be expected to be treated by the component in the same way (i.e. they may be considered 'equivalent').

Recommendations on defining partitions

A number of items must be considered:

- All valid input data for a given condition are likely to go through the same process.
- Invalid data can go through various processes and need to be evaluated more carefully. For example:
  - a blank entry may be treated differently than an incorrect entry,
  - a value that is less than a range of values may be treated differently than a value that is greater,
  - if there is more than one error condition within a particular function, one error may override the other, which means the subordinate error does not get tested unless the other value is valid.

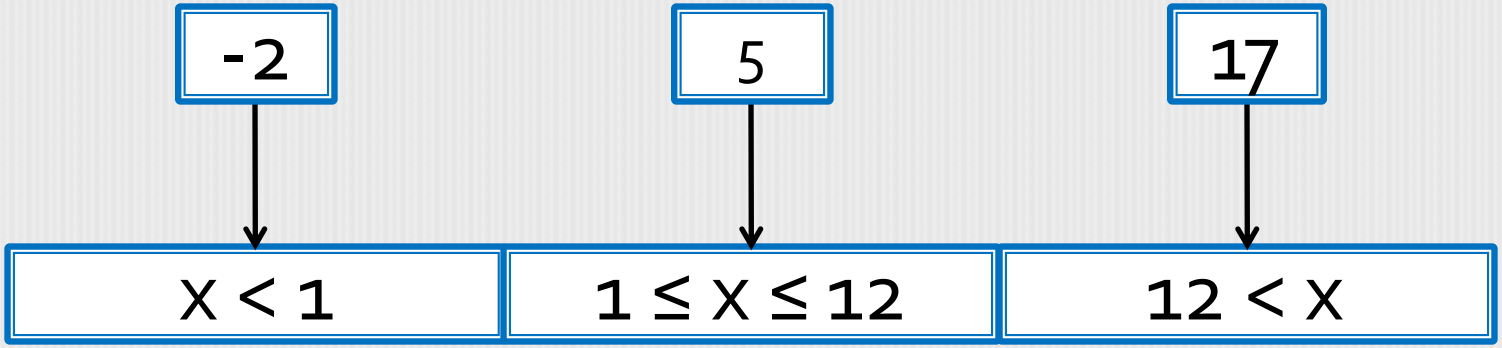
Equivalence partitioning example

- Example of a function which takes a parameter "month".
- The valid range for the month is 1 to 12, representing January to December. This valid range is called a partition.
- In this example there are two further partitions of invalid ranges.

$X < 1$	$1 \leq X \leq 12$	$12 < X$

Equivalence partitioning example

- Test cases are chosen so that each partition would be tested.

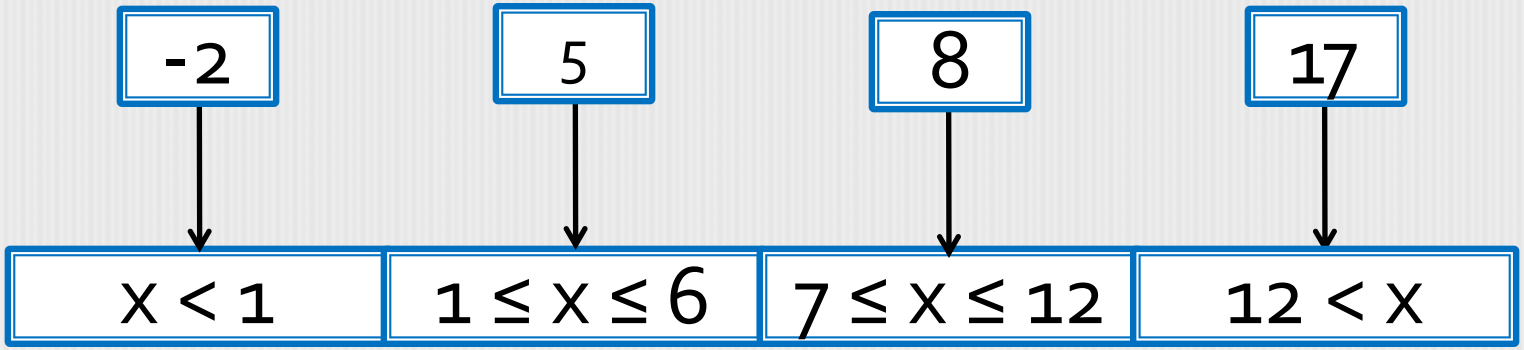


So, is it black box?

- The tendency is to relate equivalence partitioning to so called ***black box*** testing.
- However ***grey box*** technique might be applied as well.

Equivalence partitioning using grey-box technique

- Depending upon the input the software internally will run through ***different paths*** to perform slightly different actions according to the part of the year.
- Therefore middle partition is divided into two subpartitions.

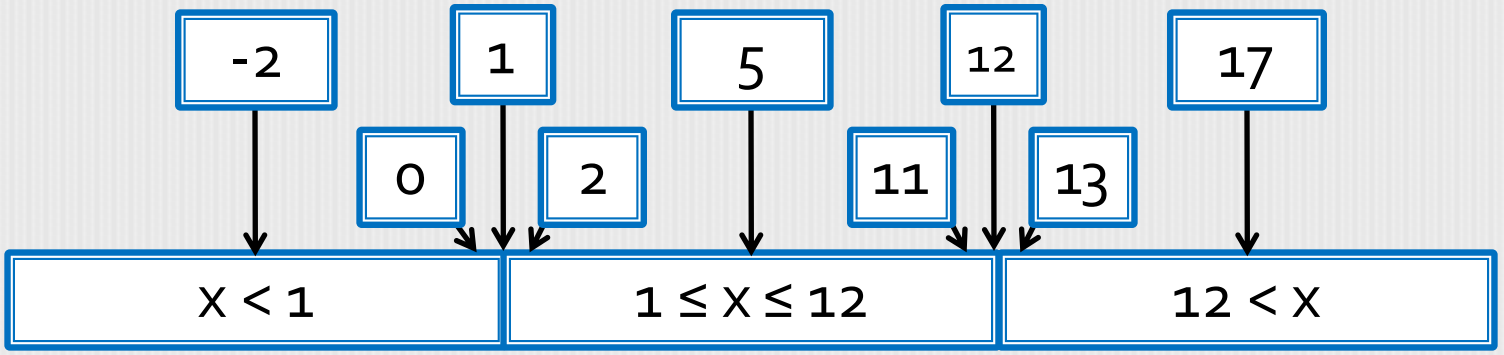


Boundary value analysis

- Equivalence partitioning is not a stand alone method to determine test cases. It is usually supplemented by ***boundary value analysis***.
- ***Boundary value analysis*** focuses on values on the edge of an equivalence partition or at the smallest value on either side of an edge.

Equivalence partitioning with boundary value analysis

- We use the same example as before.
- Test cases are supplemented with ***boundary values***.



Summary

Pros:

- optimum test case size, therefore time-saving;
- uncovers a class of error, not just an error with specific data input.

Cons:

- possible mistakes when defining partitions;
- does not test all inputs.

Question #1

Is equivalence partitioning testing a black-box method?

Answer

Equivalence partitioning testing is usually considered as black-box method, however grey-box technique can be applied too.

Question #2

How does boundary value analysis supplement equivalence partitioning?

Answer

Boundary value analysis supplements equivalence partitioning by adding boundary value test cases.

References

- http://en.wikipedia.org/wiki/Equivalence_partitioning
- http://en.wikipedia.org/wiki/Boundary_value_analysis
- http://it.toolbox.com/blogs/enterprise-solutions/testing-via-equivalence-partitioning-17116
- http://www.softwaretestinghelp.com/what-is-boundary-value-analysis-and-equivalence-partitioning/
- http://www.scribd.com/doc/49845951/6/Equivalence-Partitioning

Thank You