Black Box Tests

Specification-Based Testing
Black-Box Testing

- To implement the behavioral testing, it is required to derive and select tests by analyzing the test basis.
- Test bases are the documents
- Test bases directly or indirectly explain
  - how the component or system under test should and shouldn't behave
  - what it is required to do
  - how it is required to do it.

Equivalence Partitioning

- This is the most basic of specification-based test design techniques
- The method is about testing various groups that are expected to be handled the same way by the system and exhibit similar behavior.
- Those groups can be inputs, outputs, internal values, calculations, time values, valid and invalid groups.

Equivalence Partitioning

- We select a single value from each equivalence partition
  - This allows us to reduce the number of tests.
- We can calculate coverage by dividing the number of equivalence partitions tested by the number identified
- The goal is to achieve 100% coverage by selecting at least one value from each partition.

Equivalence Partitioning

- This technique will find primarily functional defects
  - Where data is processed improperly in one or more partitions.
- The key to this technique is to take care that the values in each equivalence partition are handled the same way
  - Otherwise, it is possible to miss potentially important test values

Equivalent Classes—Equivalent Partitions

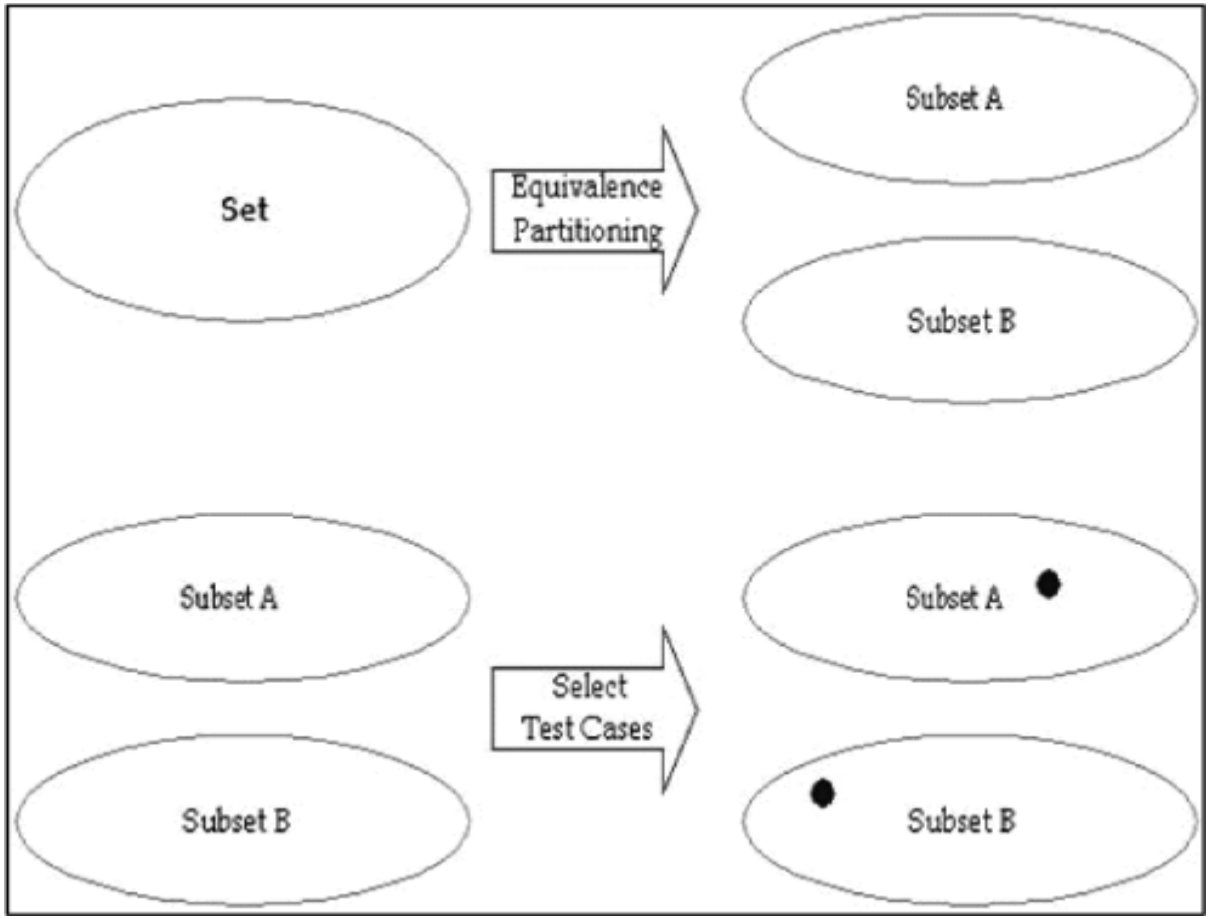
- inputs,
- outputs,
- internal values,
- time relationships,
- calculations,
- just about anything else of interest

- These classes or partitions are called equivalent
  - They should be handled the same way by the system

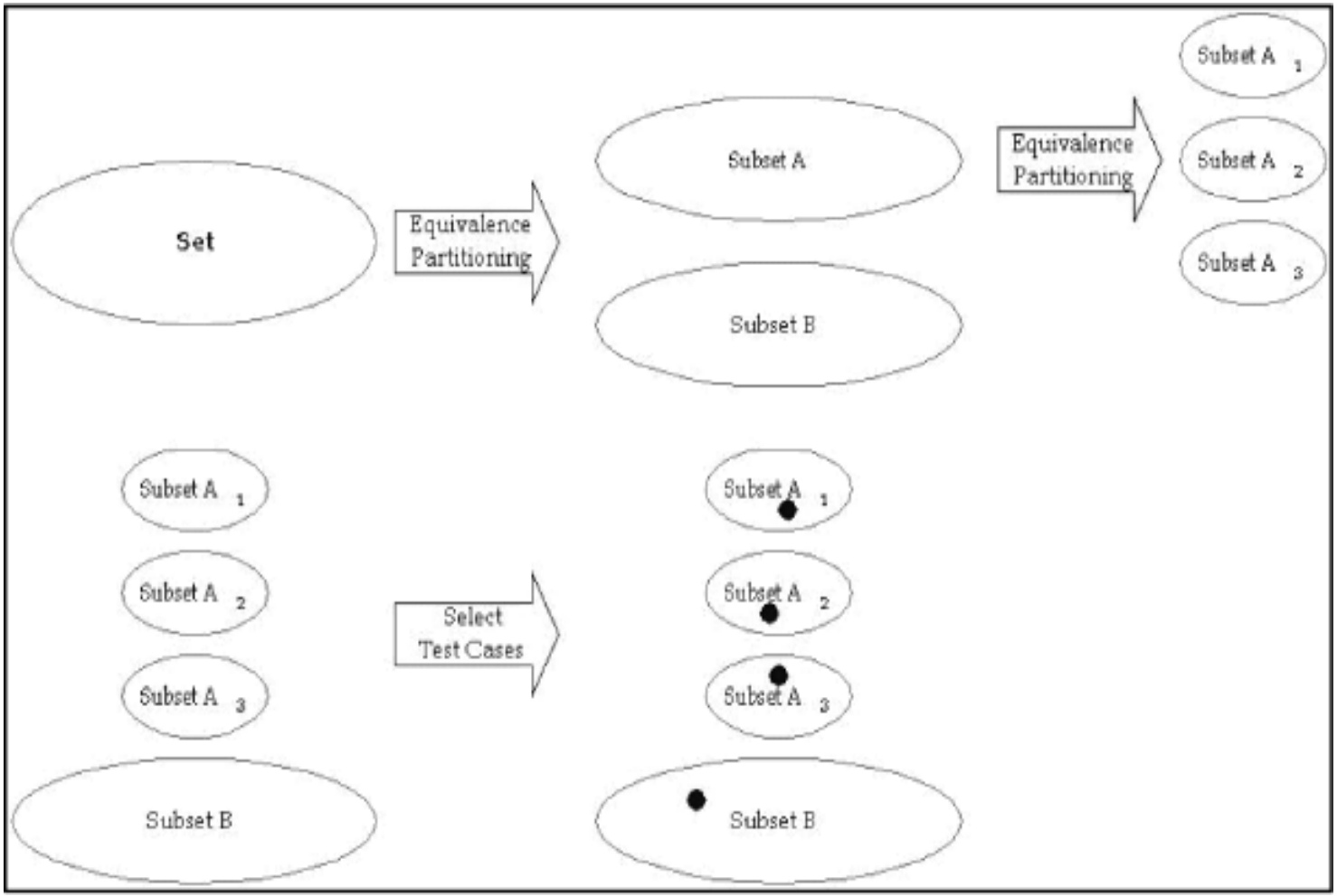
Valid and Invalid Equivalence Classes

- Some of the classes can be called valid equivalence classes
  - They describe valid situations that the system should handle normally.
- Other classes can be called invalid equivalence classes
  - They describe invalid situations
    - ✓The system should reject
    - ✓The system at least escalate to the user for correction or exception handling

Visualizing Equivalence Partitioning



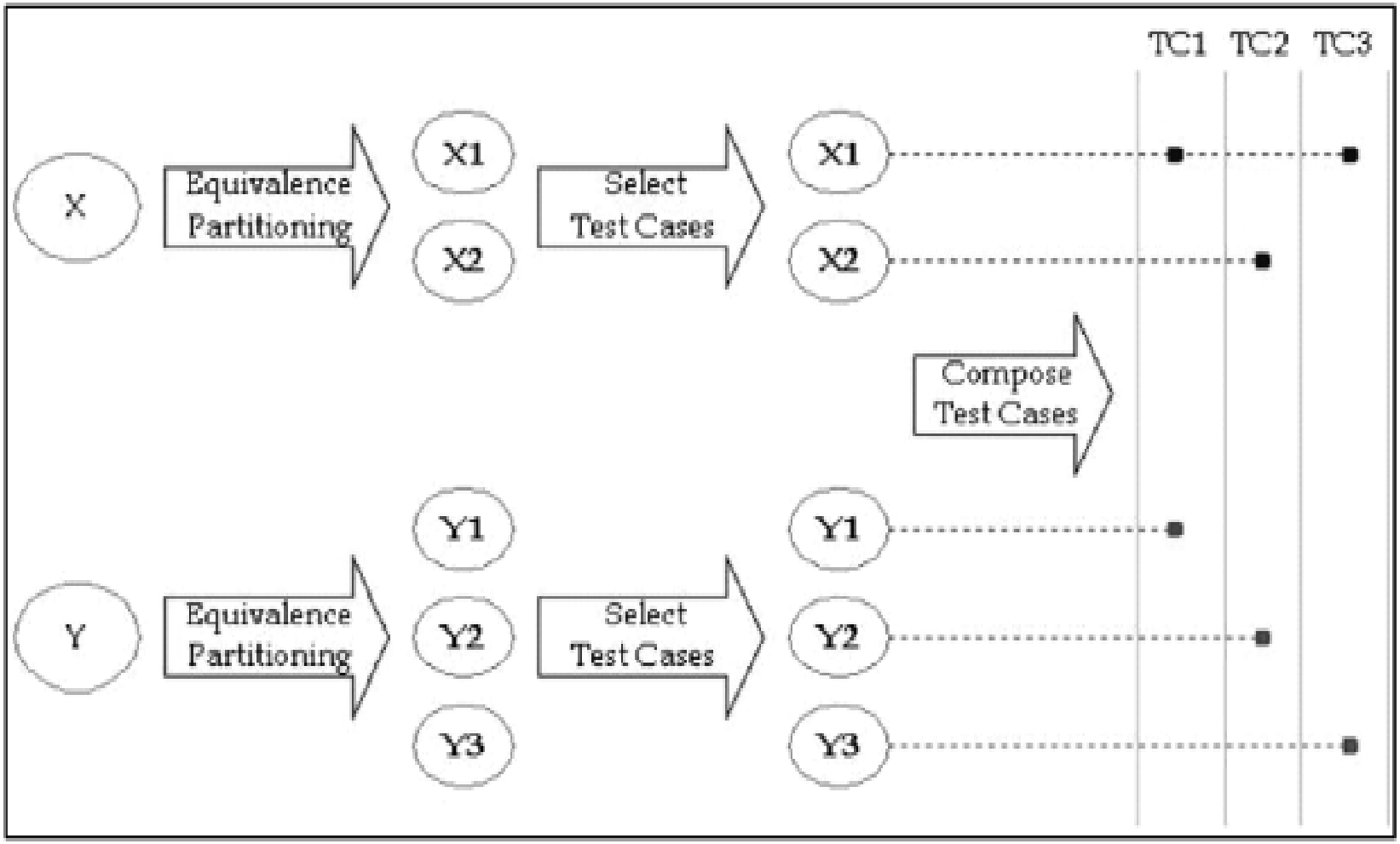
Equivalence Partitioning Applied Iteratively



An Important Factor for the Equivalence Partitioning

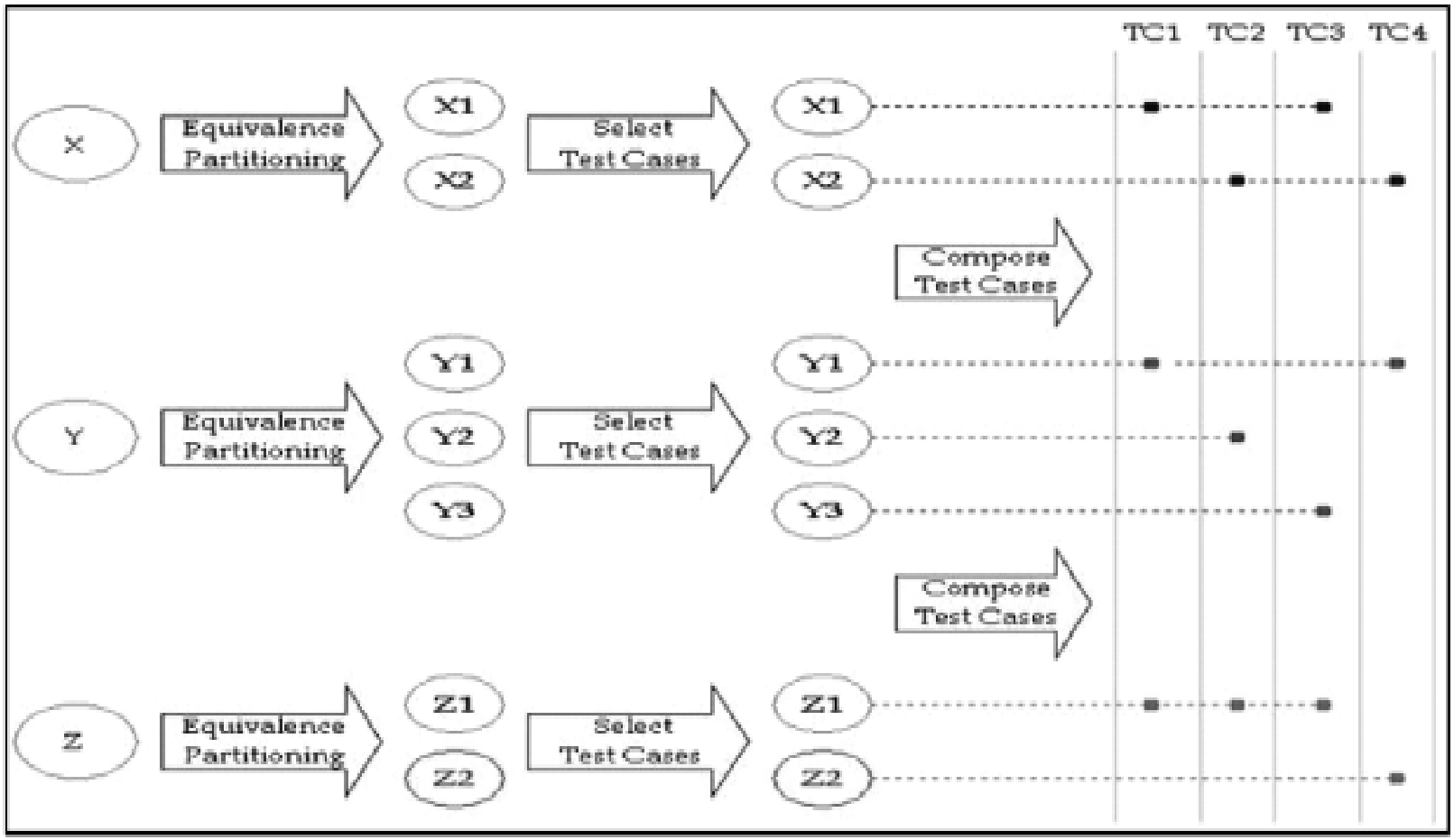
- The equivalence partitioning process does not subtract; it divides
- The union of the subsets must be the same as the original set that was equivalence partitioned
- Equivalence partitioning does not generate "spare" subsets
- If this is not true, failing may occcur
  - During the test of some important subset of inputs, outputs, configurations, …….

Composing Tests



The values may be independent and all valid,
or dependent and all valid

Composing Tests with Invalid Values



If we wanted to sub partition the invalid situation, we could also test with a start date in the present and one in the future, together with an end date before the start date, which would add two more tests.

Combinatorial Testing

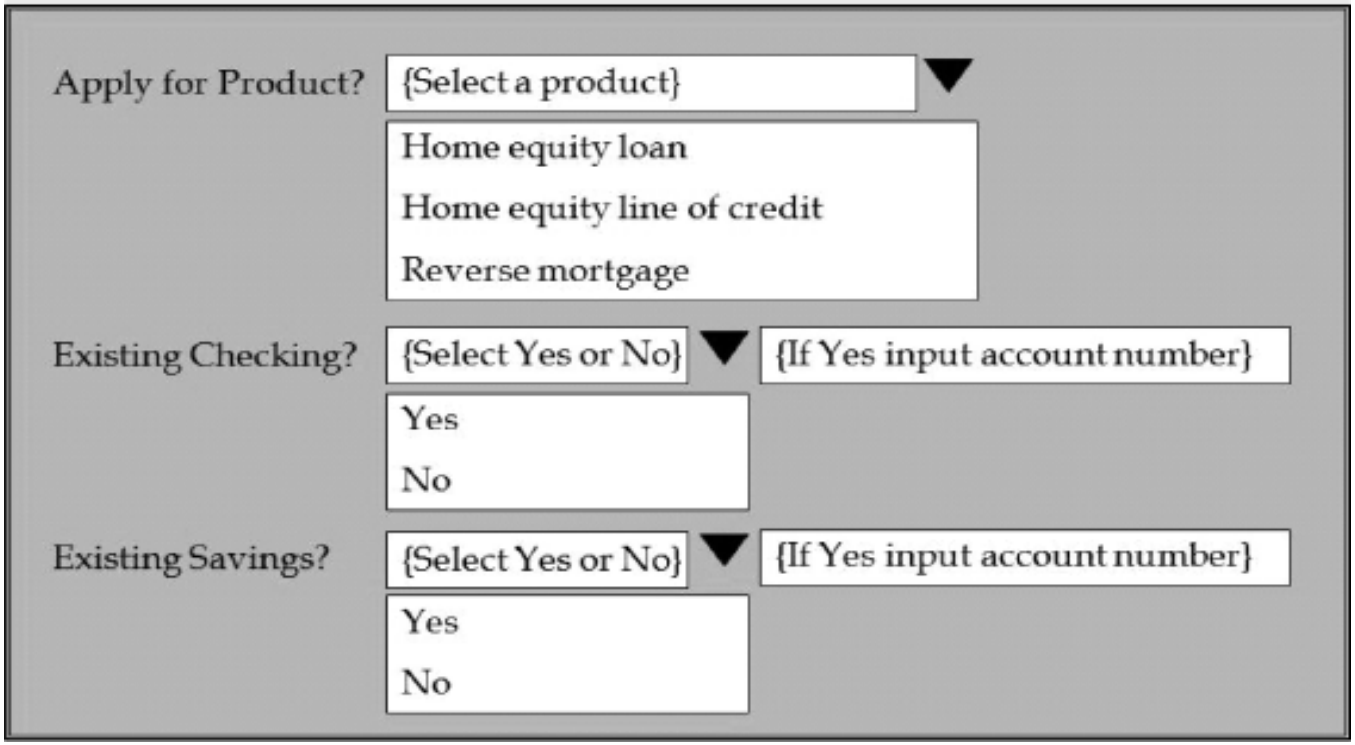
- Identify a suitable subset of test combinations
- Testing process achieves a predetermined level of coverage
  - when testing an object with multiple parameters
    - where parameters themselves each have several values
      - which gives rise to more combinations than are feasible to test in the time allowed.

For example

- Test combinations of tasks and milestones (Test Case TC1) with past, present, and future start dates (Test Case TC2) and valid end dates (Test Case TC3).
- We check that illegal end dates are correctly rejected (Test Case TC4).

Equivalence Partitioning Exercise

Screen prototype for one screen of the HELLOCARMS system



Equivalence Partitioning Debrief

For the application-product field, the equivalence partitions are:

#Partition

1Home equity loan

2Home equity line of credit

3Reverse mortgage

Equivalence Partitioning Debrief

- For each of two existing-account entries, the situation is best modeled as a single input field, which consists of two subfields.
- The first subfield is the Yes/No field.
  - This subfield determines the rule for checking the second subfield, which is the account number
  - If the first subfield is Yes, the second subfield must be a valid account number.
  - If the first subfield is No, the second subfield must be blank.

The existing checking account information partitions are:

#Partition

1Yes-Valid

2Yes-Invalid

3No-Blank

4No-NonblankAnd,

The existing savings account information partitions are:

#Partition

1Yes-Valid

2Yes-Invalid

3No-Blank

4No-Nonblank

For both of these, partitions 2 and 4 are invalid partitions, while partitions 1 and 3 are valid partitions.

Test cases for Equivalence Partitions

Inputs	1	2	3	4	5	6	7
Product	HEL	LOC	RM	HEL	LOC	RM	HEL
Existing Checking?	Yes	No	No	Yes	No	No	No
Checking Account	Valid	Blank	Blank	Invalid	Nonblank	Blank	Blank
Existing Savings?	No	Yes	No	No	No	Yes	No
Savings Account	Blank	Valid	Blank	Blank	Blank	Invalid	Nonblank
Outputs							
Accept?	Yes	Yes	Yes	No	No	No	No
Error?	No	No	No	Yes	Yes	Yes	Yes

Product

#	*Partition*	*Test Case*
1	Home equity loan (HEL)	1
2	Home equity line of credit (LOC)	2
3	Reverse mortgage (RM)	3

Checking

#	*Partition*	*Test Case*
1	Yes-Valid	1
2	Yes-Invalid	4
3	No-Blank	2
4	No-Nonblank	5

Boundary Value Analysis (BVA)

A black-box test design technique in which test cases are designed based on boundary values.

Boundary Value Analysis

- In boundary value analysis, we first use equivalence partitioning.
- For those partitions that are ordered values, we select the minimum and maximum values—the boundary values—as test values.
- This technique can be applied at any level of testing.
- It can even be used during white-box testing
  - when looking at loops
  - when looking at data structures such as lists and arrays,
  - during non-functional testing for boundaries in memory usage and performance.

Boundary Value Analysis

- Coverage is calculated as with equivalence partitioning
  - Counts the tested and identified boundary values
- It is important to apply this technique with ordered sets
- It is important to understand the precision of representation of the values.
- This technique will find situations
  - Where boundaries are in the wrong place
  - Boundaries are just plain missing
  - Undesired boundaries exist.
- In non-functional testing, performance and reliability problems under high levels of load are found with this technique.

Boundary Value Analysis

- Boundary value analysis is about testing the edges of equivalence classes.
- Instead of selecting one member of the class, the largest and smallest members of the class are selected and they are tested

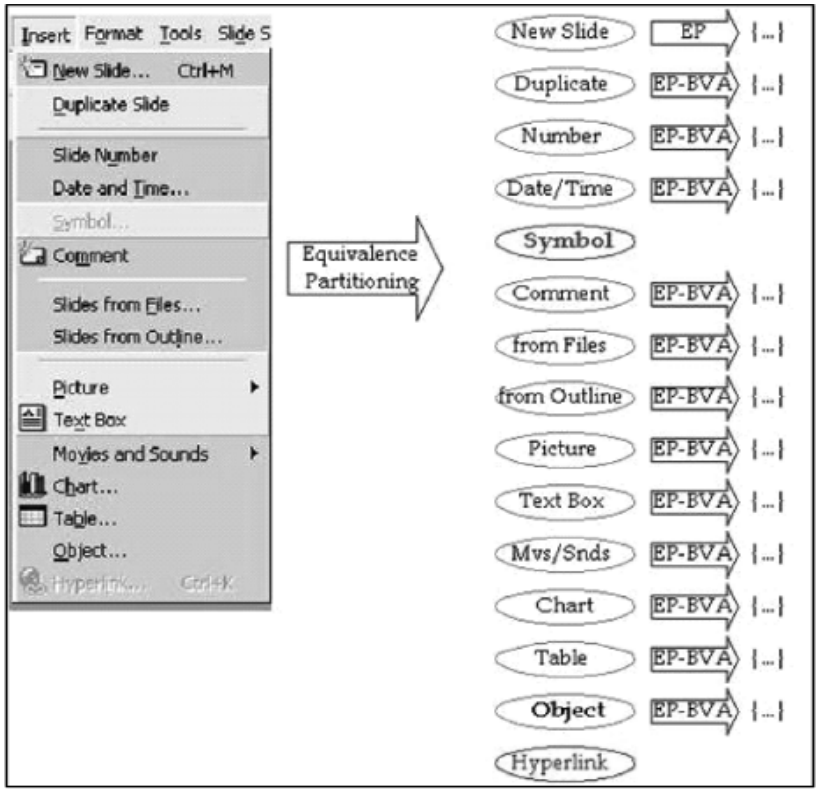
The Difference between Equivalence Partitioning and Boundary Value Analysis

- We have to represent each boundary value in a test, analogous to the equivalence partitioning situation.
- The coverage criterion is that every boundary value, both valid and invalid, must be represented in at least one test.
- The main difference is that there are at least two boundary values in each equivalence class.
- More tests are required about twice as many.

Pull-Down Menu

- Each selection is its own equivalence class.
- There is a first item and a last item on the menu,
- The ordering of the set is arbitrary and has no meaning below the user interface.
- Inside the software, there is no greater-than/less-than relationship between the items on this menu.
- We can use equivalence partitioning,and test each selection on the menu, not boundary values.
- There is no invalid selection possible here
  - The only selections we can't choose are grayed out.
  - Does that always work? Well, no.
  - It is required to test those things
    - ✓ should be grayed out
    - ✓ shouldn't be grayed out
    - ✓ grayed out are indeed inaccessible.

Example: Pull-Down Menu



- For each valid menu selection, we need further equivalence partitioning and, in most cases, boundary value analysis.
- For example, we might want to try to insert slides from a file that contains only one slide, as well as a file that contains the maximum number of slides.
- We will also want to try inserting a file that contains zero slides (if that's possible) and a file that contains one more than the maximum number of slides.
- For each valid menu selection, we'll need further equivalence partitioning and, in most cases, boundary value analysis.