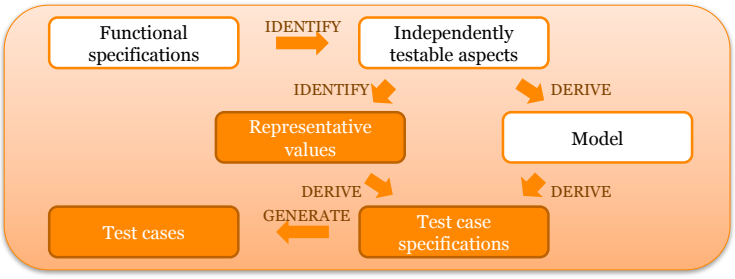
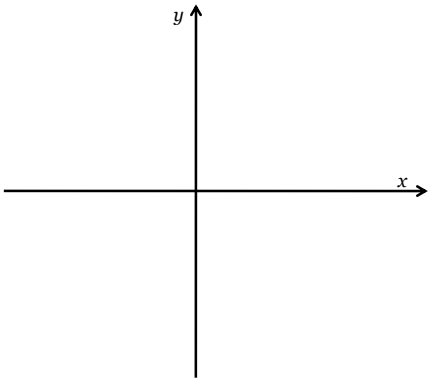
Equivalence Partitioning

Techniques based on partitioning of the input space

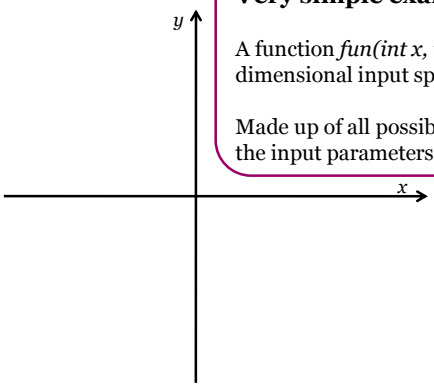
Identifying representative values



Input Space



Input Space

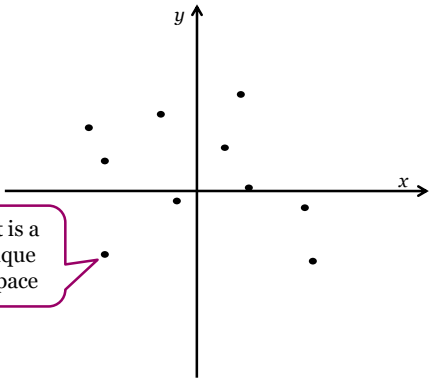


Very simple example:

A function *fun(int x, int y)* has a two-dimensional input space.

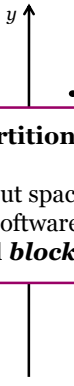
Made up of all possible combinations of the input parameters *x* and *y*.

Input Space



Each test case input is a (x,y) pair, and a unique point in the input space

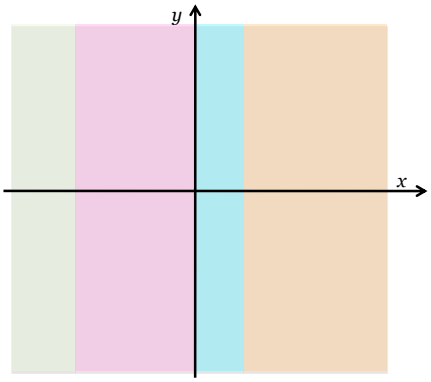
Input Space



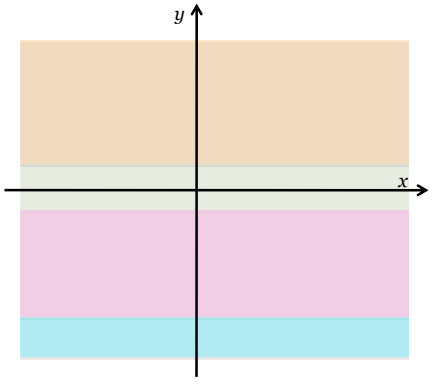
Equivalence Partitioning – basic idea:

1. Divide input space into ***blocks***.
2. Within the blocks, the software should behave ***similarly***.
3. Then just cover all ***blocks*** instead of all inputs.

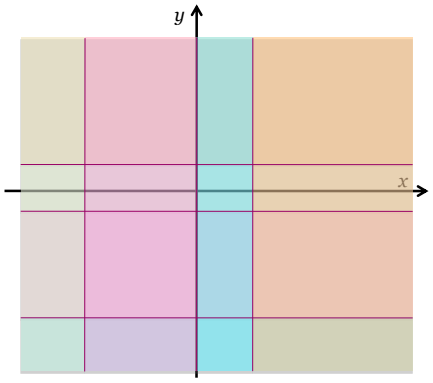
Equivalence Partitioning



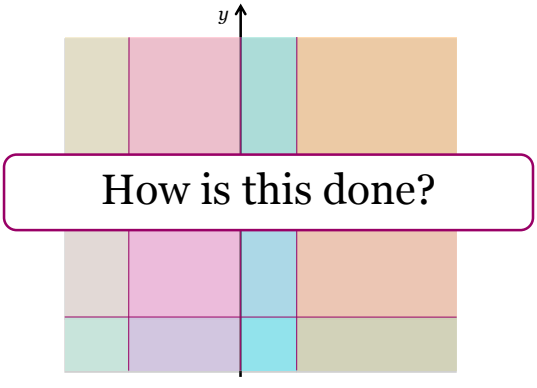
Equivalence Partitioning



Equivalence Partitioning

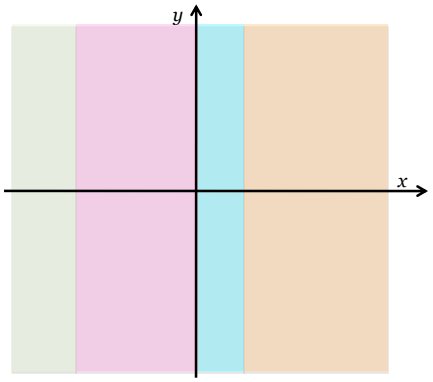


Equivalence Partitioning



Equivalence Partitioning

Basic Concepts



Equivalence Partitioning

Basic Concepts

Equivalence Partitioning

Basic Concepts

- **Characteristic**
  - A rule that states how different groups of input values should be treated (e.g., valid versus invalid input)

Equivalence Partitioning

Basic Concepts

- **Characteristic**
  - A rule that states how different groups of input values should be treated (e.g., valid versus invalid input).
- **Partition**
  - A division of the input space into equivalence classes based on a certain characteristic.

Equivalence Partitioning

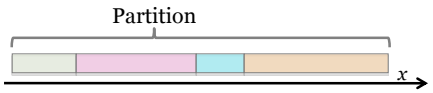
Basic Concepts

- **Characteristic**
  - A rule that states how different groups of input values should be treated (e.g., valid versus invalid input).
- **Partition**
  - A division of the input space into equivalence classes based on a certain characteristic.
- **Block**
  - A part of a partition within which all values should be treated equally with respect to a certain characteristic.

Equivalence Partitioning

Basic Concepts

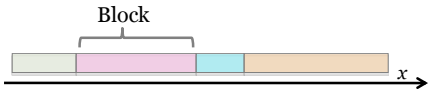
- **Characteristic**
  - A rule that states how different groups of input values should be treated (e.g., valid versus invalid input).
- **Partition**
  - A division of the input space into equivalence classes based on a certain characteristic.
- **Block**
  - A part of a partition within which all values should be treated equally with respect to a certain characteristic.



Equivalence Partitioning

Basic Concepts

- **Characteristic**
  - A rule that states how different groups of input values should be treated (e.g., valid versus invalid input).
- **Partition**
  - A division of the input space into equivalence classes based on a certain characteristic.
- **Block**
  - A part of a partition within which all values should be treated equally with respect to a certain characteristic.



Equivalence Partitioning

Basic Concepts

- **Characteristic**
  - A rule that states how different groups of input values should be treated (e.g., valid versus invalid input).
- **Partition**
  - A division of the input space into equivalence classes based on a certain characteristic.
- **Block**
  - A part of a partition within which all values should be treated equally with respect to a certain characteristic.

Partition properties:

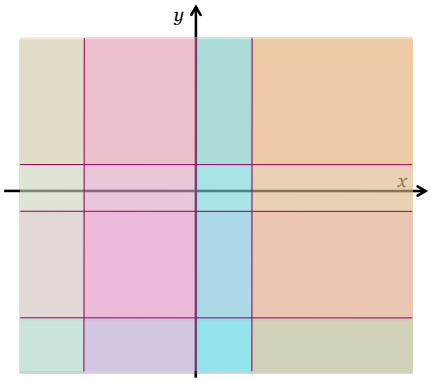
All blocks within a partition must be **disjoint** (there can be no overlap).

The union of all blocks in a partition must make up the **complete** input space.

Equivalence Partitioning

Basic Concepts

- **Characteristic**
  - A rule that states how different groups of input values should be treated (e.g., valid versus invalid input).
- **Partition**
  - A division of the input space into equivalence classes based on a certain characteristic.
- **Block**
  - A part of a partition within which all values should be treated equally with respect to a certain characteristic.
- **Input domain model (IDM)**
  - The combination of a set of partitions for the same input domain, including the definition of incompatible combinations of blocks.



Equivalence Partitioning

General Process for any given ITA

1. Identify the input space (all parameters that may affect the behavior).
2. Partition based on characteristics. Strategies:
   - Individual partitioning of each input parameter
   - Functional (output-based) modeling of the input space
3. Combine partitions into an input domain model.
   - Here, identify and remove invalid combinations
4. Identify representative values
5. Derive test cases by complementing with expected output

Equivalence Partitioning

General Process for any given ITA

1. Identify the input space (all parameters that may affect the
   - Individual partitioning of each input parameter
   - Functional (output-based) modeling of the input space
3. Combine partitions into an input domain model.
   - Here, identify and remove invalid combinations
4. Identify representative values
5. Derive test cases by complementing with expected output

Let's look at how this can be applied to the ***days between dates*** problem

Equivalence Partitioning

Example

1. Identify the input space (all parameters that may affect the behavior).
2. Partition based on characteristics. Strategies:
   - Individual partitioning of each input parameter
   - Functional (output-based) modeling of the input space
3. Combine partitions into an input domain model.
   - Here, identify and remove invalid combinations
4. Identify representative values
5. Derive test cases by complementing with expected output

For the days between dates problem, these are the two input strings representing the dates.

Let's call them a and b.

Equivalence Partitioning
Example

1. Identify the input space (all parameters that may affect the behavior).
2. Partition based on characteristics. Strategies:
   - Individual partitioning of each input parameter
   - Functional (output-based) modeling of the input space
3. Combine partitions into an input domain model.
   - Here, identify and remove invalid combinations
4. Identify representative values
5. Derive test cases by complementing with expected output

Characteristic	B_1	B_2	B_3
Valid input a?	null	Not a valid date	A valid date
Valid input b?	null	Not a valid date	A valid date

Equivalence Partitioning
Example

1. Identify the input space (all parameters that may affect the behavior).
2. Partition based on characteristics. Strategies:
   - Individual partitioning of each input parameter
   - Functional (output-based) modeling of the input space
3. Combine partitions into an input domain model.
   - Here, identify and remove invalid combinations
4. Identify representative values
5. Derive test cases by complementing with expected output

Characteristic	B_1	B_2	B_3
Leap year	The interval between date a and b includes at least one valid 29th of February	The interval between date a and b does not include any valid 29th of February	a and b do not define a valid interval

Equivalence Partitioning
Example

1. Identify the input space (all parameters that may affect the behavior).
2. Partition based on characteristics. Strategies:
   - Individual partitioning of each input parameter
   - Functional (output-based) modeling of the input space
3. Combine partitions into an input domain model.
   - Here, identify and remove invalid combinations
4. Identify representative values
5. Derive test cases by complementing with expected output

Equivalence Partitioning
Example

Characteristic									
Leap year	The interval between date a and b includes at least one valid 29th of February			The interval between date a and b does not include any valid 29th of February			a and b do not define a valid interval		
Valid input a?	null	Not a valid date	A valid date	null	Not a valid date	A valid date	null	Not a valid date	A valid date
Valid input b?	null	Not a valid date	A valid date	null	Not a valid date	A valid date	null	Not a valid date	A valid date

Equivalence Partitioning

Example

Characteristic									
Leap year	The interval between date a and b includes at least one valid 29th of February			The interval between date a and b does not include any valid 29th of February			a and b do not define a valid interval		
Valid input a?	null	Not a valid date	A valid date	null	Not a valid date	A valid date	null	Not a valid date	A valid date
Valid input b?	null	Not a valid date	A valid date	null	Not a valid date	A valid date	null	Not a valid date	A valid date

Question: How many combinations we have to test?

9 18 27 36

Equivalence Partitioning

Example

Characteristic									
Leap year	The interval between date a and b includes at least one valid 29th of February			The interval between date a and b does not include any valid 29th of February			a and b do not define a valid interval		
Valid input a?	null	Not a valid date	A valid date	null	Not a valid date	A valid date	null	Not a valid date	A valid date
Valid input b?	null	Not a valid	A valid date	null	Not a valid	A valid date	null	Not a valid date	A valid date

On a first glance, this gives us 3X3X3=27 possible combinations...

...now, let's see if we have any invalid combinations of blocks from different partitions...

Equivalence Partitioning

Example

Characteristic									
Leap year	The interval between date a and b includes at least one valid 29th of February			The interval between date a and b does not include any valid 29th of February			a and b do not define a valid interval		
Valid input a?	null	Not a valid date	A valid date	null	Not a valid date	A valid date	null	Not a valid date	A valid date
Valid input b?	null	Not a valid date	A valid date	null	Not a valid date	A valid date	null	Not a valid date	A valid date

Equivalence Partitioning

Example

Characteristic									
Leap year	The interval between date a and b includes at least one valid 29th of February			The interval between date a and b does not include any valid 29th of February			a and b do not define a valid interval		
Valid input a?	null	Not a valid date	A valid date	null	Not a valid date	A valid date	null	Not a valid date	A valid date
Valid input b?	null	Not a valid date	A valid date	null	Not a valid date	A valid date	null	Not a valid date	A valid date

Equivalence Partitioning
Example

Characteristic									
Leap year	The interval between date a and b includes at least one valid 29th of February			The interval between date a and b does not include any valid 29th of February			a and b do not define a valid interval		
Valid input a?	null	Not a valid date	A valid date	null	Not a valid date	A valid date	null	Not a valid date	A valid date
Valid input b?	null	Not a valid date	A valid date	null	Not a valid date	A valid date	null	Not a valid date	A valid date

Question: How many combinations now we have to test?

9 10 11 27

Equivalence Partitioning
Example

Characteristic									
Leap year	The interval between date a and b includes at least one valid 29th of February			The interval between date a and b does not include any valid 29th of February			a and b do not define a valid interval		
Valid input a?	null	Not a valid date	A valid date	null	Not a valid date	A valid date	null	Not a valid date	A valid date
Valid input b?	null	Not a valid date	A valid date	null	Not a valid date	A valid date	null	Not a valid date	A valid date

In this very simple example, if we feel like testing inputs from all partition combinations, we still have 11 (or 10?) different test cases to worry about...

Equivalence Partitioning

Example

1. For each ITA, identify all parameters that affect its behavior. These make up the input space.
2. Partition the input space based on a set of characteristics. Strategies:
   - Individual partitioning of each input parameter
   - Functional (or even output-based) modeling of the input space
3. Combine the partitions into an input domain model.
   - In this step, identify and remove invalid combinations
4. Identify representative values
5. Derive test cases by complementing with expected output

We leave these for you to think about…