Pseudo code for Testing Algorithm:

```
1. Start Testing
2. Import Libraries (numpy)
3. Define Dataset (X, y)
4. Load Trained Weights and Bias (weights, bias)
5. Testing Loop:
    6. For each Data Point in Dataset:
        7. Forward Pass using Trained Weights/Bias
        8. Make Prediction
        9. Print Input and Predicted Output
10. End Testing
```

here's a brief explanation of the pseudocode:

Training Pseudocode:

1. **Start Training**: Indicates the beginning of the training process.
2. **Import Libraries**: This step imports the necessary libraries, particularly NumPy for numerical operations.
3. **Define Dataset**: Sets up the input-output pairs for the AND gate dataset.
4. **Initialize Weights and Bias**: Randomly initializes the weights and bias for the perceptron.
5. **Define Hyperparameters**: Sets the learning rate and number of epochs for training.

6. Training Loop: Begins the loop over the specified number of epochs.
   7. For each Epoch: Iterates over each epoch.
      8. Initialize all_correct flag: Flags whether all predictions in the current epoch are correct.
      9. For each Data Point in Dataset: Iterates over each data point in the dataset.
         10. Forward Pass: Computes the weighted sum of inputs and bias.
         11. Make Prediction: Determines the output prediction based on the forward pass result.
         12. If Prediction is Incorrect: Checks if the prediction is incorrect.
            13. Update Weights and Bias: Adjusts the weights and bias if the prediction is incorrect.
            14. Set all_correct to False: Flags that at least one prediction in the current epoch is incorrect.
      15. If all_correct is True: Checks if all predictions in the epoch are correct.
         16. Print First Correct Prediction: Prints the epoch number, weights, and bias when the first correct prediction is made.
         17. Break Training Loop: Exits the training loop since the perceptron has learned the correct classification.
18. End Training: Marks the end of the training process.

Testing Pseudocode:

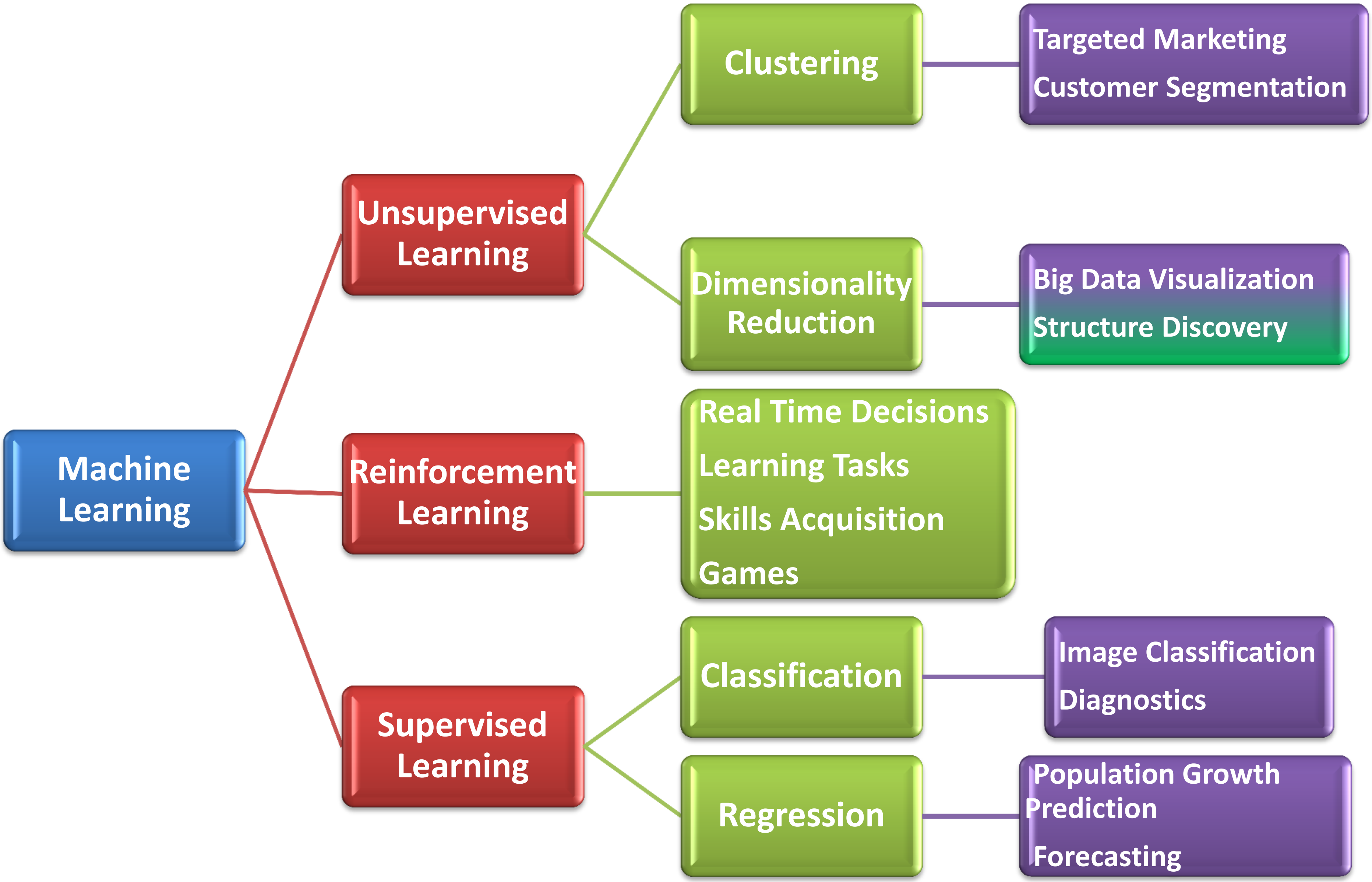
1. **Start Testing**: Marks the beginning of the testing process.
2. **Import Libraries**: Imports necessary libraries for testing, such as NumPy.
3. **Define Dataset**: Defines the input-output pairs for the testing dataset.
4. **Load Trained Weights and Bias**: Loads the weights and bias obtained from the training phase.
5. **Testing Loop**: Begins the loop for testing each data point.
    6. **For each Data Point in Dataset**: Iterates over each data point in the testing dataset.
        7. **Forward Pass using Trained Weights/Bias**: Computes the output using the trained weights and bias.
        8. **Make Prediction**: Determines the output prediction based on the forward pass result.
        9. **Print Input and Predicted Output**: Displays the input data and the predicted output.
10. **End Testing**: Marks the end of the testing process.

Multi Layer Perceptron for XOR

```
from sklearn.neural_network import MLPClassifier

# Define XOR input and output
X = [[0, 0], [0, 1], [1, 0], [1, 1]]
y = [0, 1, 1, 0]

# Create MLPClassifier
mlp = MLPClassifier(hidden_layer_sizes=(4,), activation='relu', solver='adam', max_iter=10000, verbose=True)

# Train the model
mlp.fit(X, y)

# Print weights values
print("Weights values:")
for i, coef in enumerate(mlp.coefs_):
    print(f"Layer {i} weights:")
    print(coef)

# Print number of epochs
print("Number of epochs:", mlp.n_iter_)
```

```python
from sklearn.cluster import KMeans
import numpy as np

# Generate synthetic data
np.random.seed(0)
X = np.random.randn(100, 2)  # 100 samples with 2 features

# Create KMeans model
kmeans = KMeans(n_clusters=3, random_state=0)

# Fit the model to the data
kmeans.fit(X)

# Get cluster centroids and labels
centroids = kmeans.cluster_centers_
labels = kmeans.labels_

# Print cluster centroids and labels
print("Cluster centroids:")
print(centroids)
print("\nLabels:")
print(labels)
```

```python
# Example input values (features)
X = [
    [1.0, 2.0],
    [2.0, 3.0],
    [3.0, 4.0],
    [2.5, 1.5],
    [3.5, 2.5],
    [4.0, 4.0]
]

# Example output values (labels)
y = [0, 0, 0, 1, 1, 1]
```



Machine Learning
Unsupervised Learning
Reinforcement Learning
Supervised Learning
Clustering
Targeted Marketing
Customer Segmentation
Dimensionality Reduction
Big Data Visualization
Structure Discovery
Real Time Decisions
Learning Tasks
Skills Acquisition
Games
Classification
Image Classification
Diagnostics
Regression
Population Growth Prediction
Forecasting

Fuzzy logic

Figure 3.1: A fuzzy logic system accepting imprecise data and providing a decision

In 1965 Lotfi Zadeh, published his famous paper “Fuzzy sets”. This new logic for representing and manipulating fuzzy terms was called fuzzy logic, and Zadeh became the Master/Father of fuzzy logic.

Fuzzy logic is the logic underlying approximate, rather than exact, modes of reasoning. It operates on the concept of membership. The membership was extended to possess various "degrees of membership" on the real continuous interval [0, 1].

In fuzzy systems, values are indicated by a number (called a truth value) ranging from 0 to 1, where 0.0 represents absolute falseness and 1.0 represents absolute truth.

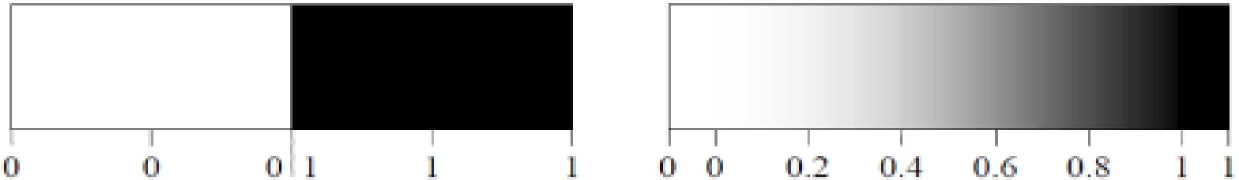


Figure 3.2: (a) Boolean Logic (b) Multi-valued Logic

Classical sets(Crisp sets)

A classical set is a collection of objects with certain characteristics. For example, the user may define a classical set of negative integers, a set of persons with height less than 6 feet, and a set of students with passing grades. Each individual entity in a set is called a member or an element of the set.

There are several ways for defining a set. A set may be defined using one of the following:

1. The list of all the members of a set may be given.
   Example A= {2,4,6,8,10} (**Roaster form**)
2. The properties of the set elements may be specified.
   Example A = {x|x is prime number < 20} (**Set builder form**)
3. The formula for the definition of a set may be mentioned. Example

$$A = \left\{ x_i = \frac{x_i + 1}{5}, i = 1 \text{ to } 10, \quad \text{where } x_i = 1 \right\}$$

4. The set may be defined on the basis of the results of a logical operation.
   Example A = {x|x is an element belonging to P AND Q}
5. There exists a membership function, which may also be used to define a set. The membership is denoted by the letter χ and the membership function for a set A is given by (for all values of x).

$$\chi_A(x) = \begin{cases} 1, & if\ x \in A \\ 0, & if\ x \notin A \end{cases}$$

The set with no elements is defined as an empty set or null set. It is denoted by symbol Ø.

The set which consist of all possible subset of a given set A is called power set

$$P(A) = \{x | x \subseteq A\}$$

Example

Consider a simple set containing three elements:

$$A = \{1, 2, 3\}$$

To find the power set of A, we list out all possible subsets of A, including the empty set and the set itself.

1. The empty set: $\{\}$
2. Single-element subsets: $\{1\}, \{2\}, \{3\}$
3. Two-element subsets: $\{1, 2\}, \{1, 3\}, \{2, 3\}$
4. The set itself: $\{1, 2, 3\}$

So, the power set of A contains all possible combinations of elements from A, including the empty set and the set itself. It looks like this:

$$\mathcal{P}(A) = \{\{\}, \{1\}, \{2\}, \{3\}, \{1, 2\}, \{1, 3\}, \{2, 3\}, \{1, 2, 3\}\}$$

The power set of A contains $2^3 = 8$ elements, which is 2^n where n is the number of elements in the original set A.

Operations on Classical sets

1. Union

The union between two sets gives all those elements in the universe that belong to either set A or set B or both sets A and B. The union operation can be termed as a logical OR operation. The union of two sets A and B is given as

$$A \cup B = \{x | x \in A \; or \; x \in b\}$$

The union of sets *A* and *B* is illustrated by the Venn diagram shown below

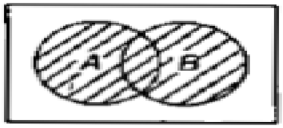


Figure 3.3: Union of two sets

2. Intersection

The intersection between two sets represents all those elements in the universe that simultaneously belong to both the sets. The intersection operation can be termed as a logical AND operation. The intersection of sets A and B is given by

$$A \cap B = \{x | x \in A \; and \; x \in b\}$$

The intersection of sets *A* and *B* is illustrated by the Venn diagram shown below

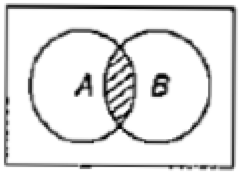


Figure 3.4: Intersection of two sets

3. Complement

The complement of set A is defined as the collection of all elements in universe X that do not reside in set A, i.e., the entities that do not belong to A. It is denoted by A and is defined as

$$\bar{A} = \{x | x \notin A, x \in X\}$$

where X is the universal set and A is a given set formed from universe X. The complement operation of set A is show below

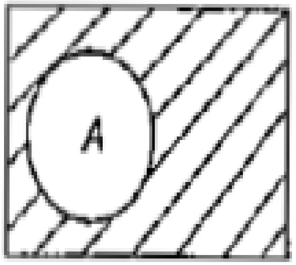


Figure 3.5: Complement of set A

4. Difference (Subtraction)

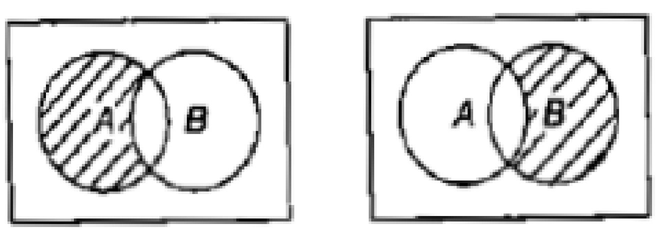
The difference of set A with respect to ser B is the collection of all elements in the universe that belong to A but do not belong to B, i.e., the difference set consists of all elements that belong to A bur do not belong to B. It is denoted by A l B or A- B and is given by

$$A|B \; or \; (A - B) = \{x|x \in A \; and \; x \notin B\} = A - (A \cap B)$$

The vice versa of it also can be performed

$$B|A \; or \; (B - A) = B - (B \cap A) = \{x|x \in B \; and \; x \notin A\}$$

The above operations are shown below



(A) (B)

Figure 3.6: (A) Difference A|B or (A-B); (B) Difference B|A or (B-A)

Union of 2 sets:

```
def set_union(set1, set2):
    """
    Function to compute the union of two sets without using built-in functions.

    Arguments:
    set1 (set): The first set.
    set2 (set): The second set.

    Returns:
    set: The union of set1 and set2.
    """
    result_set = set(set1)  # Create a copy of set1
    for element in set2:
        if element not in set1:
            result_set.add(element)
    return result_set

# Example usage:
set1 = {1, 2, 3, 4, 5}
set2 = {4, 5, 6, 7, 8}
print("Union of set1 and set2:", set_union(set1, set2))
```

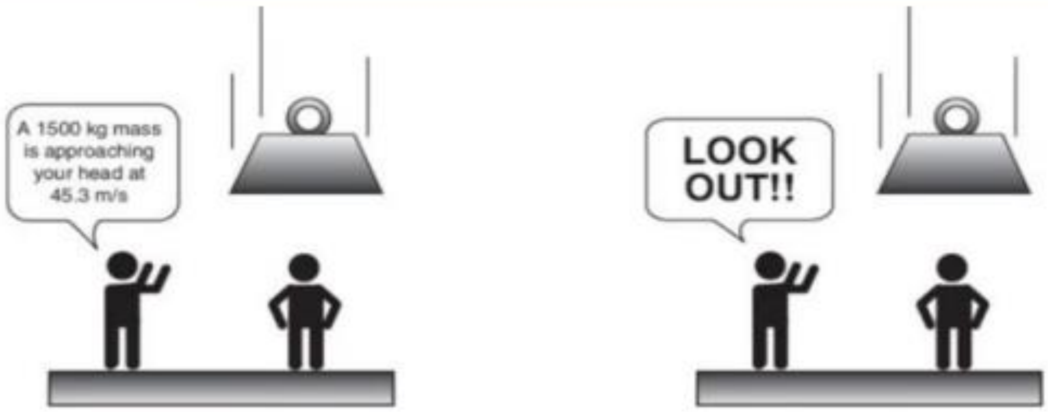
مطلوب بامتحان العملي كل من union، subtraction،intersection، complement

Precision in Fuzzy Logic

Lotfi A. Zadeh: *As complexity rises, precise statements lose meaning and meaningful statements lose precision*

في بعض الاحيان المعنى يكون اهم من دقة النص

ميزة المنطق الغامض انه يركز على المعنى اكثر من الدقة



- **Aristole** (born BC - died BC): Law of Excluded Middle (any statement can be either TRUE or FALSE, there is no middle value). Statement can take one value at a time (T or F).
- **Heraclitus** (6th to 5h century BC): Things can be simultaneously TRUE and NOT TRUE. Statement can take two values at the same time (T by one person and F by another person).
- **Plato** (428 BC - 348 BC): Laid foundation of what is fuzzy logic. There exist some value between the extremias. Statement can have certain value between TRUE and FALSE.
- **Lucasiewicz** (1878 - 1956): Described three logic value (True, False, Possible) along with mathematics. Possible varies between 0 to 1. He proofed his concepts through mathematics.
- **Zadeh** (1921- 2017): Introduced Fuzzy Logic in 1965, through his paper "Fuzzy Sets".

- Zadeh noticed that conventional computer logic can not handle data that are represented subjective or vague ideas. Hence, he developed fuzzy logic to allow computers to find the distinctions among data with shades of gray which mimicks the process of human reasoning.

What is Fuzzy Logic?

- A computational paradigm that is based on how humans think.

  is a precise problem-solving technique able to handle numerical data and linguistic knowledge.

- Fuzzy Logic looks at the world in imprecise terms somehow the same way our brain takes in information (e.g. speed is fast, temperature is cold), then responds with precise actions.

- Human brains can reason with uncertainties, vagueness, and judgments. On the other side, computers can manipulate precise data . Fuzzy logic tries to combine the two techniques.

Fuzzy logic is a convenient way to map an input space to an output space. Mapping input to output is the starting point for everything. Consider the following examples:

- With information about how good your service was at a restaurant, a fuzzy logic system can tell you what the tip should be.

- With your specification of how hot you want the water, a fuzzy logic system can adjust the faucet valve to the right setting.

- With information about how far away the subject of your photograph is, a fuzzy logic system can focus the lens for you.

- With information about how fast the car is going and how hard the motor is working, a fuzzy logic system can shift gears for you.

- Fuzzy logic uses the continuum of logical values between 0 (completely false) and 1 (completely true): spectrum of colours instead of 2 colours (crisp set).

The term "Crisp" means "Precise".

Crisp deals with strict boundaries, for example True or False

Universal Set: A set consisting of all possible elements

Example: $A =$ set of natural positive integer numbers $= \{1, 2, 3, 4, 5, \ldots\}$

Example: $B =$ set of natural positive even integer numbers $= \{2, 4, 6, \ldots\} \subset A$

$X = \{1, 2, 3\} \subset A,$

$Y = \{2, 4, 6, 8\} \subset B \subset A, \quad Z = \{-1, 0, 1\} \notin A$

X, Y are Crisp Sets of Set A, Z is a Crisp Set but not part of Set A

$\Rightarrow$ 1, 3 are members of Set A and members of the Crisp Set X
$\Rightarrow$ -1 is a member of the Crisp Set Z but not member of Set A

ϕ : Empty set or null set (set that contains no element)

Example: you have the following sets below

$A = \{1, 2, 3, 4, 5, 6, 7, 8, 9\}$

$X = \{2, 4, 6, 8\}$

$Y = \{1, 3, 5, 7\}$

$Z = \{2, 4\}$

$V = \{6, 4, 2, 8\}$

$W = \{w \in X | 20 < w < 100\} = \phi$

Then:

$x = 2 \in X$, x is a member of X

$x = 8 \notin Y$, x is not a member of Y

$Z \subseteq X$, Z is a subset of X

$X \supseteq Z$, X is a superset contains all the elements of Z

$A \supseteq X$, A is a superset contains all the elements of X

$X \subset A$, A is a proper set contains all the elements of X as well as some additional elements

$X \supset A$, A is a proper superset contains all the elements of X as well as some additional elements

$X = V$, X & V are equal sets, $X \neq Y$, X & Y are not equal

Example: you have the following sets below

$$A = \{1, 2, 3, 4, 5, 6, 7, 8, 9\}$$

$$X = \{2, 4, 6, 8\}$$

$$Y = \{1, 3, 5, 7\}$$

$$Z = \{2, 3, 4\}$$

Then:

$$X \cup Y = \{1, 2, 3, 4, 5, 6, 7, 8\}$$

$$X \cap Z = \{2, 4\}$$

Example: you have the following sets below

$$A = \{1, 2, 3, 4, 5, 6, 7, 8, 9\}$$

$$X = \{1, 2, 3\}$$

$$Y = \{2, 3, 4\}$$

$$Z = \{5, 6\}$$

Then:

Complement: $\overline{X} = A - X = \{4, 5, 6, 7, 8, 9\}$

Involution: $\overline{\overline{X}} = X = \{1, 2, 3\}$

Associativity: $(X \cup Y) \cup Z = X \cup (Y \cup Z) = \{1, 2, 3, 4, 5, 6\}$

Associativity: $(X \cap Y) \cap Z = X \cap (Y \cap Z) = \phi$

Distributive: $X \cup (Y \cap Z) = \underbrace{(X \cup Y)}_{\{1,2,3,4\}} \cap \underbrace{(X \cup Z)}_{\{1,2,3,5,6\}} = \{1,2,3\}$

Distributive: $X \cap (Y \cup Z) = \underbrace{(X \cap Y)}_{\{2,3\}} \cup \underbrace{(X \cap Z)}_{\phi} = \{2,3\}$

Example: **you have the following sets below**

$$A = \{1,2,3,4,5,6\}$$

$$X = \{1,2,3\}$$

$$Y = \{2,3,4\}$$

$$Z = \{5,6\}$$

Then:

Absorption: $X \cup (X \cap Y) = \underbrace{X}_{\{1,2,3\}} \cup \underbrace{(X \cap Y)}_{\{2,3\}} = X$

Absorption: $X \cap (X \cup Y) = \underbrace{X}_{\{1,2,3\}} \cap \underbrace{(X \cup Y)}_{\{1,2,3,4\}} = X$

Identity: $A \cap X = X$, $A \cup X = A$, $X \cup \phi = X$, $X \cap \phi = \phi$

De Morgan's Law: $\overline{X \cup Y} = \overline{X} \cap \overline{Y} = \underbrace{\overline{X}}_{\{4,5,6\}} \cap \underbrace{\overline{Y}}_{\{1,5,6\}} = \{5,6\}$

De Morgan's Law: $\overline{X \cap Y} = \overline{X} \cup \overline{Y} = \underbrace{\overline{X}}_{\{4,5,6\}} \cup \underbrace{\overline{Y}}_{\{1,5,6\}} = \{1,4,5,6\}$

Law of Contradiction: $X \cap \overline{X} = \phi$