Batch: A2 **Roll No.: 16010121045**

Experiment No. 08

Grade: AA / AB / BB / BC / CC / CD /DD

Signature of the Staff In-charge with date

Title: Defuzzification methods.

Aim : To understand the concept of Defuzzification.

Expected Outcome of Experiment:

CO4 : Apply basics of Fuzzy logic and neural networks

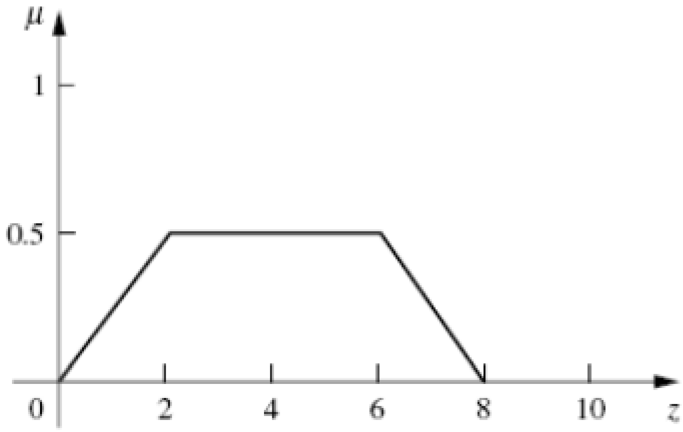
Books/ Journals/ Websites referred:

Pre Lab/ Prior Concepts:

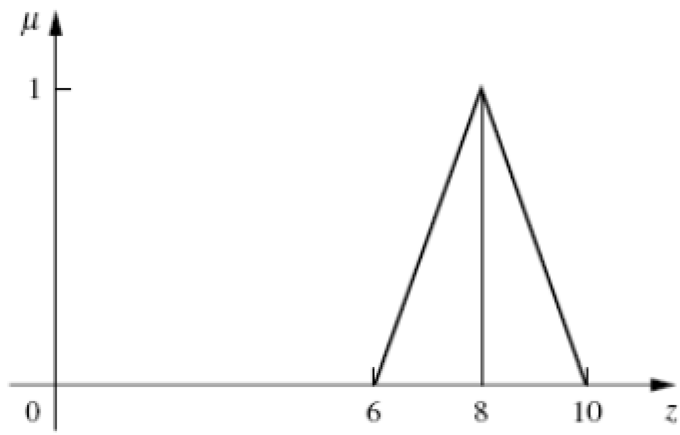
Defuzzification :

Defuzzification is the process of producing a quantifiable result in Crisp logic, given fuzzy sets and corresponding membership degrees. It is the process that maps a fuzzy set to a crisp set. It is typically needed in fuzzy control systems. These will have a number of rules that transform a number of variables into a fuzzy result, that is, the result is described in terms of membership in fuzzy sets. Defuzzification is the conversion of a fuzzy quantity to a precise quantity, just as fuzzification is the conversion of a precise quantity to a fuzzy quantity. μ

For example, **Fig (a)** shows the first part of the Fuzzy output and **Fig (b)** shows the second part of the Fuzzy output.

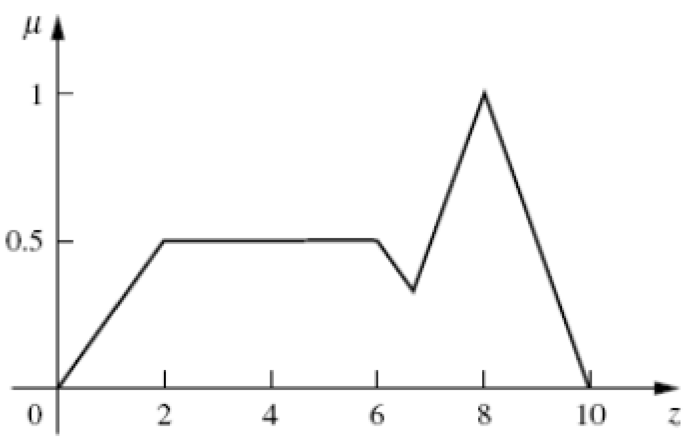


(a)



(b)

Then **Fig (c)** shows the union of the two parts (a) and (b).

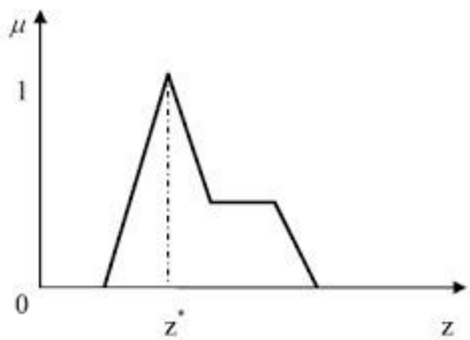


(c)

Different Defuzzification methods

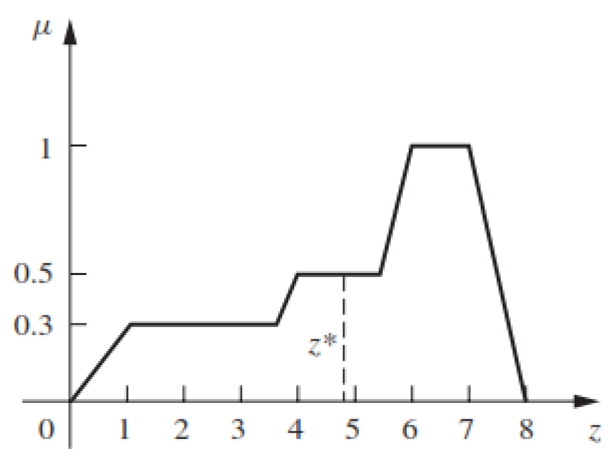
1. **Max membership method**

This method is also known as height method and is limited to peak output functions. This method is given by the algebraic expression:
$\boldsymbol{\mu}(z^*) >= \boldsymbol{\mu}(z)$ for all $z \in Z$.



2. **Center of gravity or centroid**

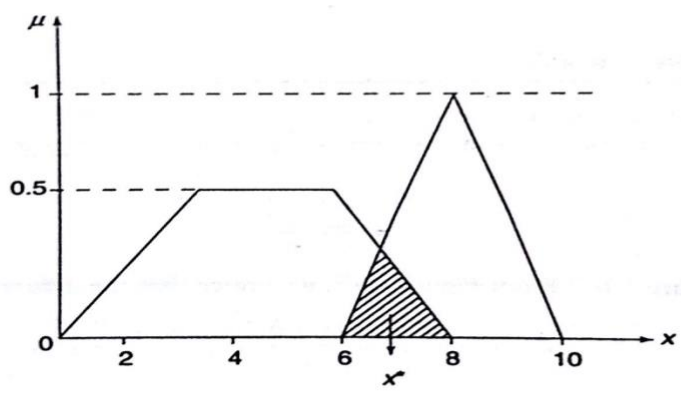
This method is also known as the centre of mass, centre of area or centre of gravity. It is the most commonly used defuzzification method. The defuzzified output z* is given by:
$$\mathbf{z^* = \int \mu(z).z dz \, / \int \mu(z) dz}$$



3. Centre of sums

This method employs the algebraic sum of the individual fuzzy subsets instead of their union. The calculations here are very fast, but the main drawback is that the intersecting areas are added twice. The defuzzified value z* is given by

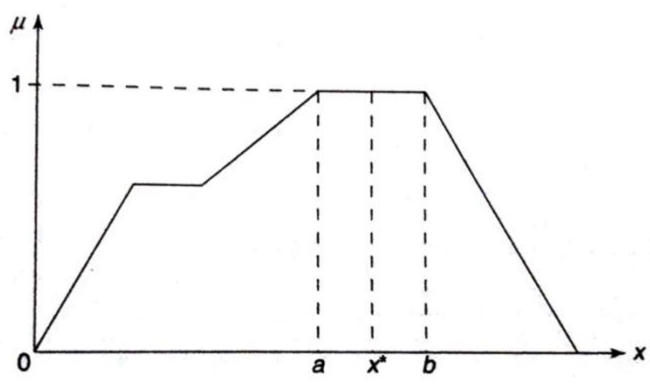
z* = ∫ z*∑µ(z).zdz / ∫ ∑µ(z)dz



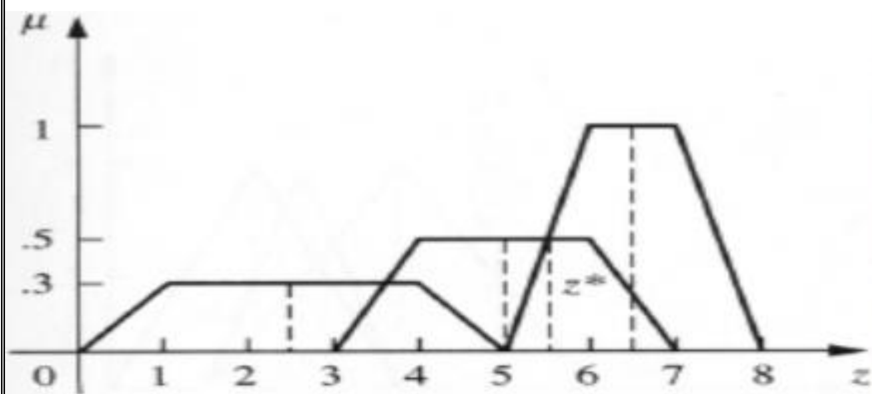
4. Mean of maximum method

This method is also known as the middle of the maxima. This is closely related to the max-membership method, except that the locations of the maximum membership can be nonunique. The output here is given by:

z* = ∑z' / n ; where z' is the maximum value of the membership function.

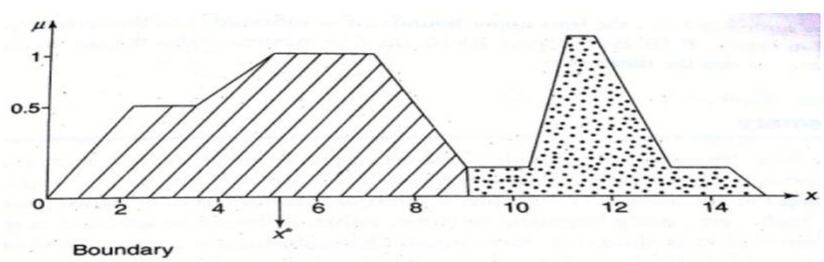


5. **Weighted average method**
This method is valid for symmetrical output membership functions only. Each membership function is weighted by its maximum membership value. The output in the case is given by $z^* = \sum\mu(z').z' / \sum\mu(z')$; where z' is the maximum value of the membership function.



6. **Centre of Largest Area**
This method can be adopted when the output of at least two convex fuzzy subsets which are not overlapping. The output, in this case, is biased towards a side of one membership function. When output fuzzy st has at least two convex regions, then the centre of gravity of the convex fuzzy subregion having the largest are is used to obtain the defuzzified value z*. The value is given by

$z^* = \int \mu c(z).zdz / \int \sum\mu c(z)dz$



Implementation Details:

```
import matplotlib.pyplot as plt


def plot_membership_function(data):
    plt.plot(data, label='Membership Function')
    plt.xlabel('Index')
    plt.ylabel('Membership Value')
    plt.legend()
    plt.show()


def lambda_cut_method(data):
    lambda_value = float(input("Enter lambda value: "))
    cut_data = [x for x in data if x >= lambda_value]
    print("Lambda Cut Method Result:", cut_data)
    plot_membership_function(data)


def weighted_average(data):
    weights = [float(w) for w in input("Enter weights corresponding to each defuzzy values: ").split()]
    if len(weights) != len(data):
        print("Error: Number of weights should match the number of data points.")
        return
    weighted_avg = sum([data[i] * weights[i] for i in range(len(data))]) / sum(weights)
    print("Weighted Average:", weighted_avg)

    # Plot weighted average
    plt.axhline(y=weighted_avg, color='r', linestyle='--', label='Weighted Average')
```

```python
    plot_membership_function(data)

def height_of_maxima(data):
    max_value = max(data)
    print("Height of Maxima:", max_value)

    # Plot maxima
    max_indexes = [i for i, value in enumerate(data) if value == max_value]
    plt.scatter(max_indexes, [max_value] * len(max_indexes), color='r', label='Maxima')
    plot_membership_function(data)

def first_of_maxima(data):
    max_value = max(data)
    first_index = data.index(max_value)
    print("First of Maxima:", data[first_index])

    # Plot first maxima
    plt.scatter(first_index, max_value, color='r', label='First of Maxima')
    plot_membership_function(data)

def last_of_maxima(data):
    max_value = max(data)
    max_indexes = [i for i, value in enumerate(data) if value == max_value]
    last_index = max_indexes[-1]
    print("Last of Maxima:", data[last_index])
```

```
    # Plot last maxima
    plt.scatter(last_index, max_value, color='r', label='Last of Maxima')
    plot_membership_function(data)

def mean_of_maxima(data):
    max_value = max(data)
    max_indexes = [i for i, value in enumerate(data) if value == max_value]
    mean_maxima = sum([data[i] for i in max_indexes]) / len(max_indexes)
    print("Mean of Maxima:", mean_maxima)

    # Plot mean maxima
    plt.axhline(y=mean_maxima, color='r', linestyle='--', label='Mean of Maxima')
    plot_membership_function(data)

def centre_of_centroid(data):
    centroid = sum(data) / len(data)
    print("Centre of Centroid:", centroid)

    # Plot centroid
    plt.axhline(y=centroid, color='r', linestyle='--', label='Centre of Centroid')
    plot_membership_function(data)

def centre_of_sum(data):
    total_sum = sum(data)
    sum_data = [data[i] / total_sum for i in range(len(data))]
    weighted_sum = sum([(i + 1) * sum_data[i] for i in range(len(data))])
```

```
    print("Centre of Sum:", weighted_sum)

    # Plot centre of sum
    plt.axvline(x=weighted_sum, color='r', linestyle='--', label='Centre of Sum')
    plot_membership_function(data)

# Main program
data = [float(x) for x in input("Enter defuzzy values: ").split()]

while True:
    print("\nMenu:")
    print("1. Lambda Cut Method")
    print("2. Weighted Average")
    print("3. Height of Maxima")
    print("4. First of Maxima")
    print("5. Last of Maxima")
    print("6. Mean of Maxima")
    print("7. Centre of Centroid")
    print("8. Centre of Sum")
    print("9. Exit")

    choice = input("Enter your choice: ")

    if choice == '1':
        lambda_cut_method(data)
    elif choice == '2':
```

```
    weighted_average(data)
elif choice == '3':
    height_of_maxima(data)
elif choice == '4':
    first_of_maxima(data)
elif choice == '5':
    last_of_maxima(data)
elif choice == '6':
    mean_of_maxima(data)
elif choice == '7':
    centre_of_centroid(data)
elif choice == '8':
    centre_of_sum(data)
elif choice == '9':
    break
else:
    print("Invalid choice. Please enter a valid option.")
```

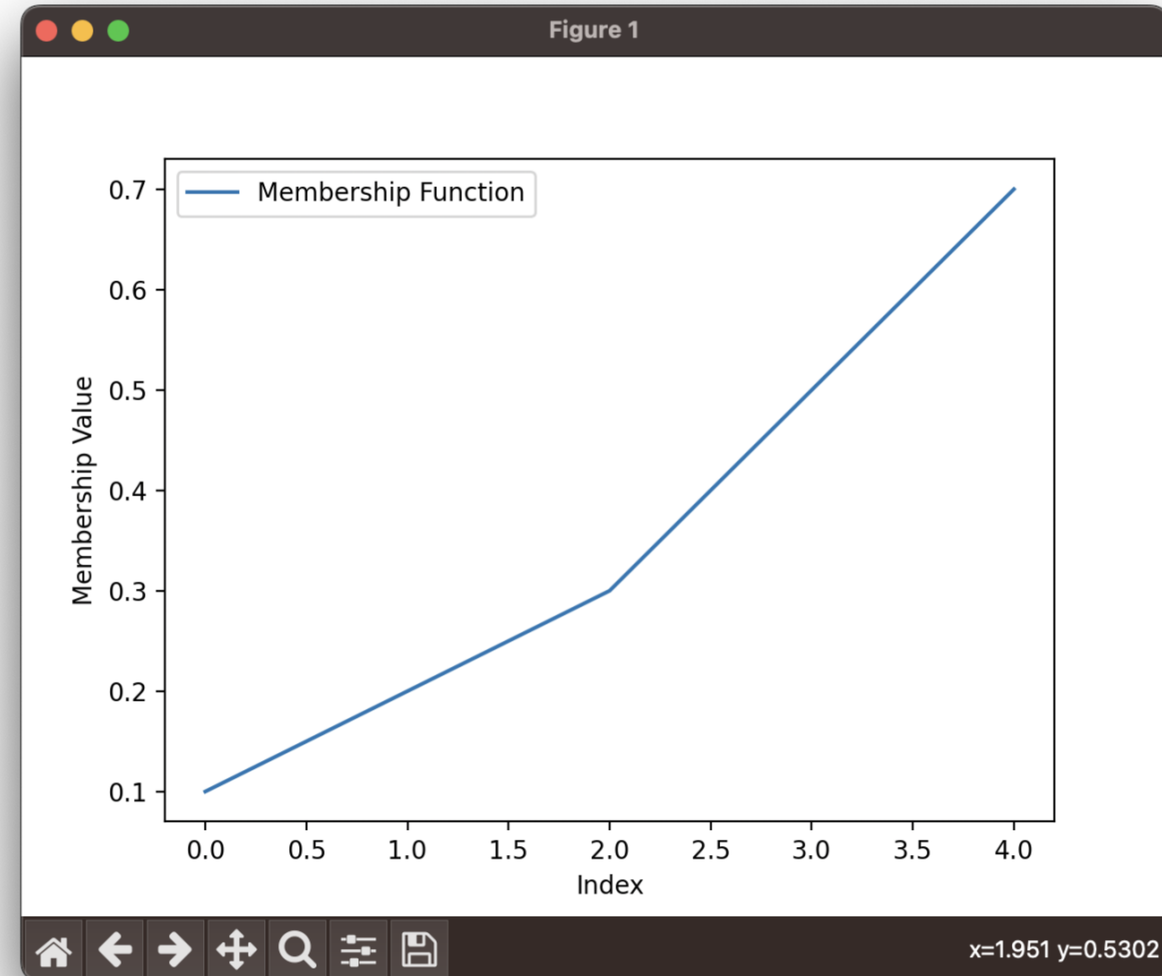


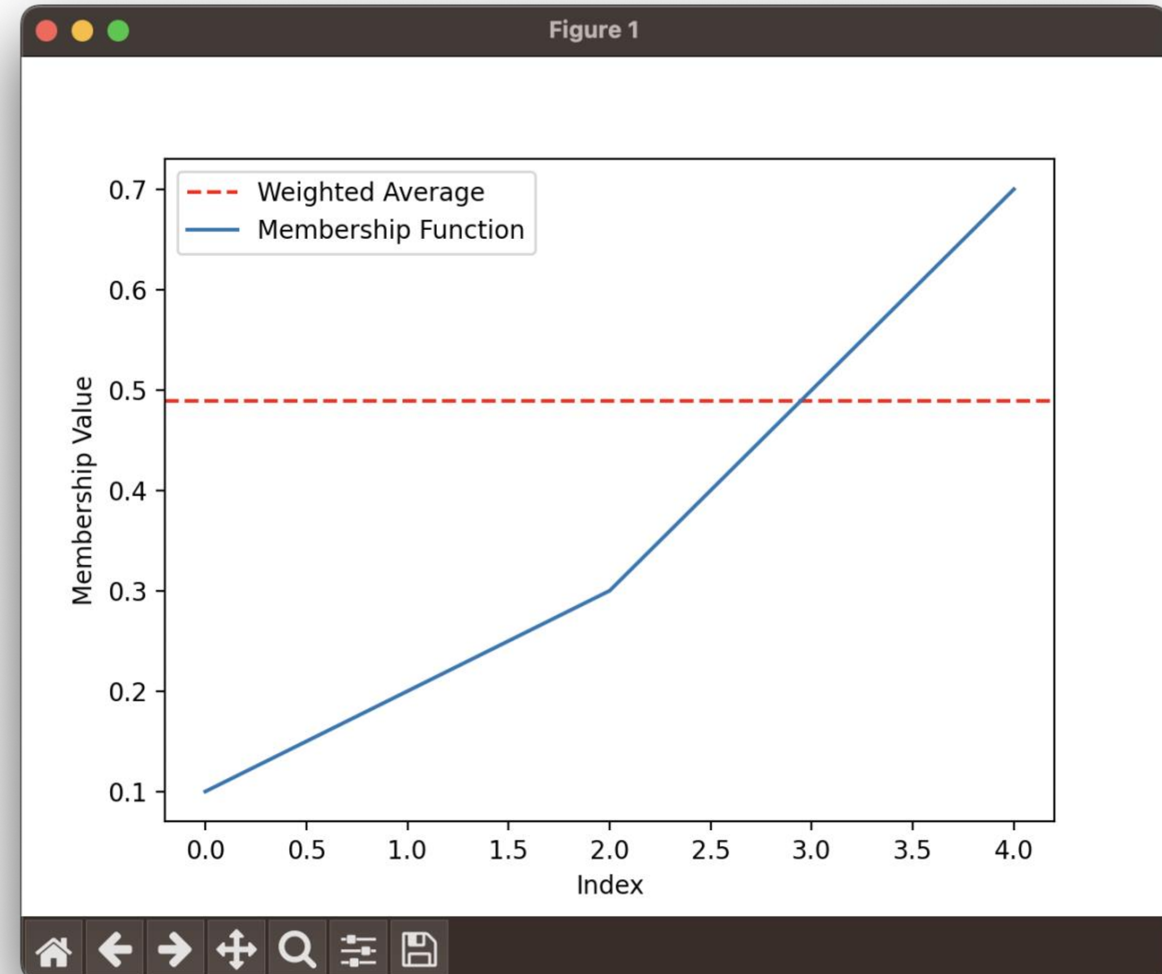
Figure 1
Membership Function
Membership Value
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.0
0.5
1.0
1.5
2.0
2.5
3.0
3.5
4.0
Index
x=1.951 y=0.5302

```
❯ python3 -u "/Users/pargatsinghdhanjal/Desktop/Soft Computing/exp9.py"
Enter defuzzy values: 0.1 0.2 0.3 0.5 0.7

Menu:
1. Lambda Cut Method
2. Weighted Average
3. Height of Maxima
4. First of Maxima
5. Last of Maxima
6. Mean of Maxima
7. Centre of Centroid
8. Centre of Sum
9. Exit
Enter your choice: 1
Enter lambda value: 0.1
Lambda Cut Method Result: [0.1, 0.2, 0.3, 0.5, 0.7]
2023-10-26 14:18:27.621 Python[11149:1057021] WARNING: Secure coding is au
. Opt-in to secure coding explicitly by implementing NSApplicationDelegate

Menu:
1. Lambda Cut Method
2. Weighted Average
3. Height of Maxima
4. First of Maxima
5. Last of Maxima
6. Mean of Maxima
7. Centre of Centroid
8. Centre of Sum
9. Exit
Enter your choice: 2
Enter weights corresponding to each defuzzy values: 0.1 0.2 0.3 0.5 0.7
Weighted Average: 0.4888888888888888

Menu:
1. Lambda Cut Method
2. Weighted Average
3. Height of Maxima
4. First of Maxima
5. Last of Maxima
6. Mean of Maxima
7. Centre of Centroid
8. Centre of Sum
9. Exit
Enter your choice: 9
```

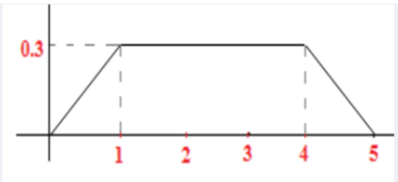


Conclusion: Implementation of defuzzification methods was done successfully.

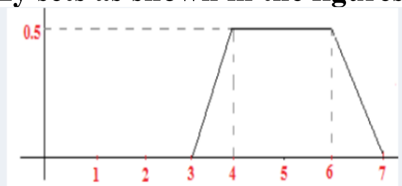
Post Lab Descriptive Questions :

1. **Let there be 3 different fuzzy sets as shown in the figures below:-**

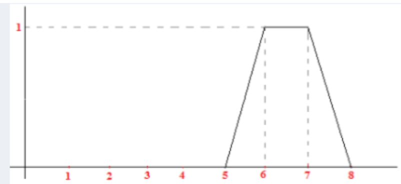
A



B

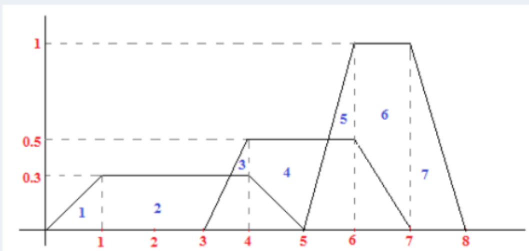


C

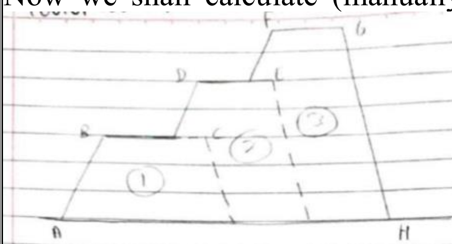


Hence the union of all the three sets can be represented by the following figure:-

D = A U B U C



Now we shall calculate (manually) the defuzzified value using all the above methods one by one.



1) Max Membership

Max value is 1.0 which occurs at $Z^* = 6$

2) Centroid Method

eq^n for line AB = $y = 0.3x$

BC = $y = 0.3$

eq^n for line BD = $0.5x - 1.5$

eq^n for line DF = $y = 0.5x$

eq^n for line EF = $y = x - 5$

eq^n for line FG = $y = 1$

eq^n for line GH = $y = x - 8$

$$\int \mu_x(x)\,dx = \int (0.3)x\,dx + \int^{6} 0.3x\,dx + \int^{4} (0.5x - 1.5)x\,dx + \int 0.5x\,dx + \int_{5.5} (x-5)x\,dx + \int x\,dx + \int_{7}^{8} (x-8)x\,dx$$

$= 24.3$

$\int \mu_x(x)\,dx = 24.3$

$Z^* = 5.65$

x' for (1) = 2.5

Centroid x' for (2) = 5

Centroid x' for (3) = 6.5

Area of (1) = 1.2

Area of (2) = 1.5

Area of (3) = 2

$Z^* = 5$

4) Mean of Maximum

Max = 1 for $z \in [6-7]$

$MOM = \frac{6+7}{2} = 6.5$

$Z^* = 6.5$

5) Weighted average method

$$Z^* = \frac{(2.5 \times 0.3) + (5 \times 0.5) + (6.5 \times 1)}{0.3 + 0.5 + 1}$$

$$= \frac{0.7 + 2.5 + 6.5}{1.8}$$

$$= \frac{9.75}{1.8}$$

$= 5.41$

Date: ________ **Signature of faculty in-charge**