Ex.no 7 fuzzy inference

Aim - to write a python code to implement fuzzy inference system .

Algorithm :

Step 1: Define the triangular membership function `trimf(x, params)` with parameters `a`, `b`, and `c`.

Step 2: Define the input and output variable ranges using `np.arange`:

- Define `temperature\_values` and `humidity\_values` ranging from 0 to 100.

- Define `fan\_speed\_values` ranging from 0 to 100.

Step 3: Define membership functions for the input variables using the `trimf` function:

- `temperature\_low` using `trimf(temperature\_values, [0, 0, 50])`.

- `temperature\_medium` using `trimf(temperature\_values, [0, 50, 100])`.

- `temperature\_high` using `trimf(temperature\_values, [50, 100, 100])`.

- `humidity\_low` using `trimf(humidity\_values, [0, 0, 50])`.

- `humidity\_medium` using `trimf(humidity\_values, [0, 50, 100])`.

- `humidity\_high` using `trimf(humidity\_values, [50, 100, 100])`.

Step 4: Define membership functions for the output variable using the `trimf` function:

- `fan\_speed\_low` using `trimf(fan\_speed\_values, [0, 0, 50])`.

- `fan\_speed\_medium` using `trimf(fan\_speed\_values, [0, 50, 100])`.

- `fan\_speed\_high` using `trimf(fan\_speed\_values, [50, 100, 100])`.

Step 5: Implement fuzzy rules in the `fuzzy\_rule` function based on the given conditions for temperature and humidity.

Step 6: Define input values for temperature and humidity (`input\_temperature`, `input\_humidity`).

Step 7: Apply fuzzy rules to determine the output fan speed (`output\_fan\_speed`) using the `fuzzy\_rule` function.

Step 8: Print the output fan speed.

Program:

import numpy as np

# Define triangular membership function

def trimf(x, params):

a, b, c = params

epsilon = 1e-10 # Small value to prevent division by zero

return np.maximum(0, np.minimum((x - a) / max((b - a), epsilon), (c - x) / max((c - b), epsilon)))

# Define input variables

temperature\_values = np.arange(0, 101, 1)

humidity\_values = np.arange(0, 101, 1)

# Define membership functions for input variables

temperature\_low = trimf(temperature\_values, [0, 0, 50])

temperature\_medium = trimf(temperature\_values, [0, 50, 100])

temperature\_high = trimf(temperature\_values, [50, 100, 100])

humidity\_low = trimf(humidity\_values, [0, 0, 50])

humidity\_medium = trimf(humidity\_values, [0, 50, 100])

humidity\_high = trimf(humidity\_values, [50, 100, 100])

# Define output variable membership functions

fan\_speed\_values = np.arange(0, 101, 1)

fan\_speed\_low = trimf(fan\_speed\_values, [0, 0, 50])

fan\_speed\_medium = trimf(fan\_speed\_values, [0, 50, 100])

fan\_speed\_high = trimf(fan\_speed\_values, [50, 100, 100])

# Define rules

def fuzzy\_rule(temperature, humidity):

if temperature == 'low' or humidity == 'low':

return 'low'

elif temperature == 'medium' and humidity == 'medium':

return 'medium'

elif temperature == 'high' or humidity == 'high':

return 'high'

# Input values

input\_temperature = 75

input\_humidity = 40

# Apply fuzzy rules

output\_fan\_speed = fuzzy\_rule(

'low' if input\_temperature <= 50 else ('medium' if input\_temperature <= 100 else 'high'),

'low' if input\_humidity <= 50 else ('medium' if input\_humidity <= 100 else 'high')

)

# Output

print(f'Fan speed: {output\_fan\_speed}')