Name : Aryan Deepak Saraf
Div : D20B
Roll No. : 49

AI&DS2 Experiment 07

Aim: To implement fuzzy set Properties.

Theory:

Introduction to Fuzzy Sets

Imagine you're trying to describe something that isn't strictly
black or white, like how "tall" someone is. Is someone 5 feet tall "tall"? What about 5 feet 10 inches? Classical (or traditional) sets would force you to draw a sharp line: either you are tall, or you are not. This can be limiting.

Fuzzy set theory offers a more flexible way to think about categories that have shades of gray. Instead of an item either being completely in a set (membership value of 1) or completely out (membership value of 0), fuzzy sets allow for partial membership. This means an item can belong to a set to a certain degree, represented by a value between 0 and 1. For example, someone who is 5 feet 10 inches might have a 0.8 membership in the set of "tall people," while someone 5 feet 5 inches might have a 0.4 membership.

This concept is very useful in areas where human-like reasoning is needed, such as in artificial intelligence for video games, controlling appliances, or medical diagnosis.

Key Properties of Fuzzy Sets

Just like regular sets, fuzzy sets have operations that allow us to combine or modify them. Here are some of the most important ones:

1. **Union of Fuzzy Sets (OR)**

The union of two fuzzy sets is like saying "A OR B." If you have two fuzzy sets, say "Player is Weak" (Set A) and "AI is Aggressive" (Set B), the union would represent scenarios where either the player is weak or the AI is aggressive (or both). For any given element, its membership in the union is the highest of its membership values in either Set A or Set B.

Formula: **Membership(A OR B) = Maximum(Membership(A), Membership(B))**

2. **Intersection of Fuzzy Sets (AND)**

The intersection of two fuzzy sets is like saying "A AND B." Using our video game example, the intersection of "Player is Weak" (Set A) and "AI is Aggressive" (Set B) would represent scenarios where the player is weak and the AI should be aggressive. For any given element, its membership in the intersection is the lowest of its membership values in either Set A or Set B.

Formula: **Membership(A AND B) = Minimum(Membership(A), Membership(B))**

3. **Complement of a Fuzzy Set (NOT)**

The complement of a fuzzy set is like saying "NOT A." If Set A is "Player is Weak," its complement would be "Player is NOT Weak" (or "Player is Strong"). For any given element, its membership in the complement is simply 1 minus its membership in the original set.

Formula: **Membership(NOT A) = 1 - Membership(A)**

4. **Scalar Multiplication of a Fuzzy Set**

Scalar multiplication involves scaling the membership values of a fuzzy set by a constant number (a scalar). If you have a fuzzy set "AI is Aggressive" and you multiply it by a scalar like 0.5, it would effectively reduce the AI's aggressiveness across all scenarios. This is useful for fine-tuning the influence of a fuzzy concept.

Formula: **Membership(alpha * A) = alpha * Membership(A)**
(where alpha is a number, usually between 0 and 1)

5. **Sum of Fuzzy Sets**

The sum of fuzzy sets is a way to combine the influence of two sets, often used when you want to consider both aspects together without strictly limiting by the minimum (as in intersection). For any given element, its membership in the sum is the sum of its membership values in both sets, but capped at 1. This ensures that the membership value doesn't go beyond the maximum possible degree of membership.

Formula: **Membership(A + B) = Minimum(1, Membership(A) + Membership(B))**

Code :

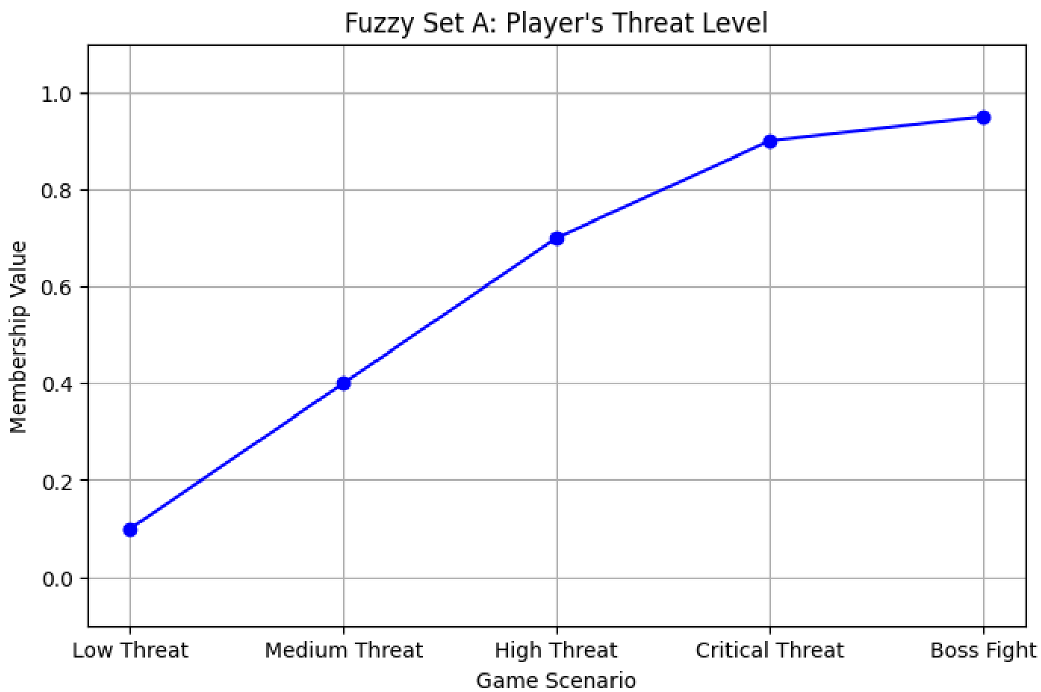
```
import matplotlib.pyplot as plt

# Universe of Discourse: Different game scenarios/states
x = ['Low Threat', 'Medium Threat', 'High Threat', 'Critical Threat', 'Boss Fight']

# Fuzzy Set A: Player's Threat Level (membership values for each scenario)
A = [0.1, 0.4, 0.7, 0.9, 0.95] # e.g., how much the player is a threat in each scenario

# Fuzzy Set B: AI's Aggressiveness (membership values for each scenario)
B = [0.3, 0.6, 0.8, 0.5, 0.2] # e.g., how aggressive the AI should be in each scenario

# 1. Union of Fuzzy Sets (A U B)
# Represents scenarios where either player is a threat OR AI should be aggressive
A_union_B = [max(a, b) for a, b in zip(A, B)]

# 2. Intersection of Fuzzy Sets (A n B)
# Represents scenarios where player is a threat AND AI should be aggressive
A_intersection_B = [min(a, b) for a, b in zip(A, B)]

# 3. Complement of a Fuzzy Set (A')
```

```python
# Represents scenarios where the player is NOT a threat
A_complement = [1 - a for a in A]
```

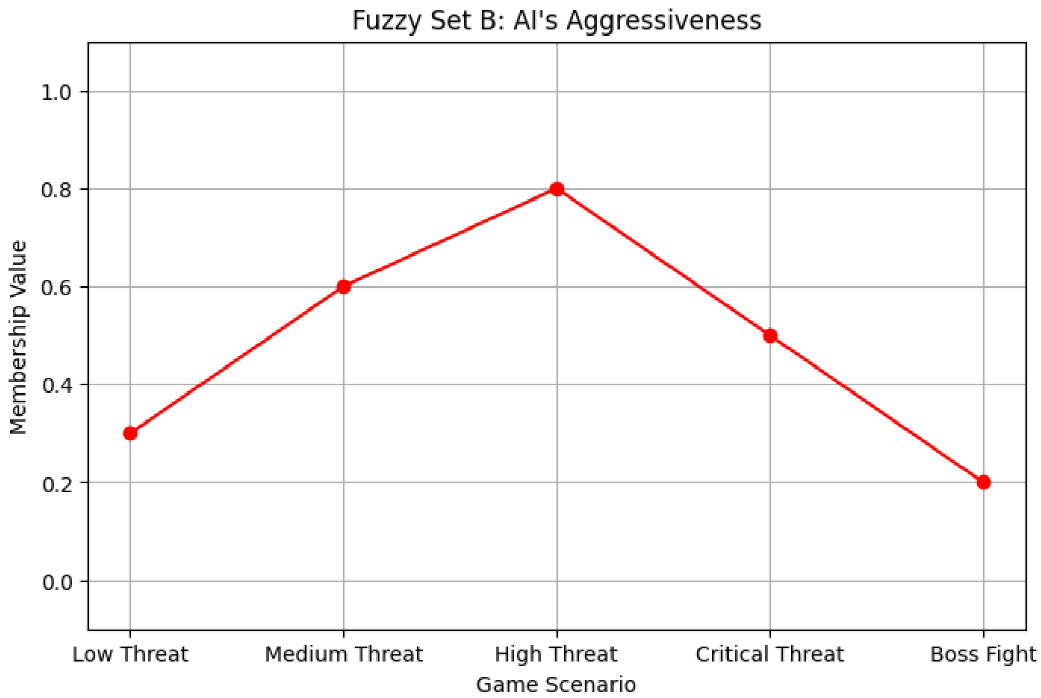
```python
# 4. Scalar Multiplication of a Fuzzy Set (alpha * A)
# Let's say alpha = 0.7 (e.g., reducing the overall perceived threat level by 30%)
alpha = 0.7
A_scalar_multiplication = [alpha * a for a in A]
```

```python
# 5. Sum of Fuzzy Sets (A + B)
# Represents a combined measure of player threat and AI aggressiveness, capped at 1
A_sum_B = [min(1, a + b) for a, b in zip(A, B)]
```

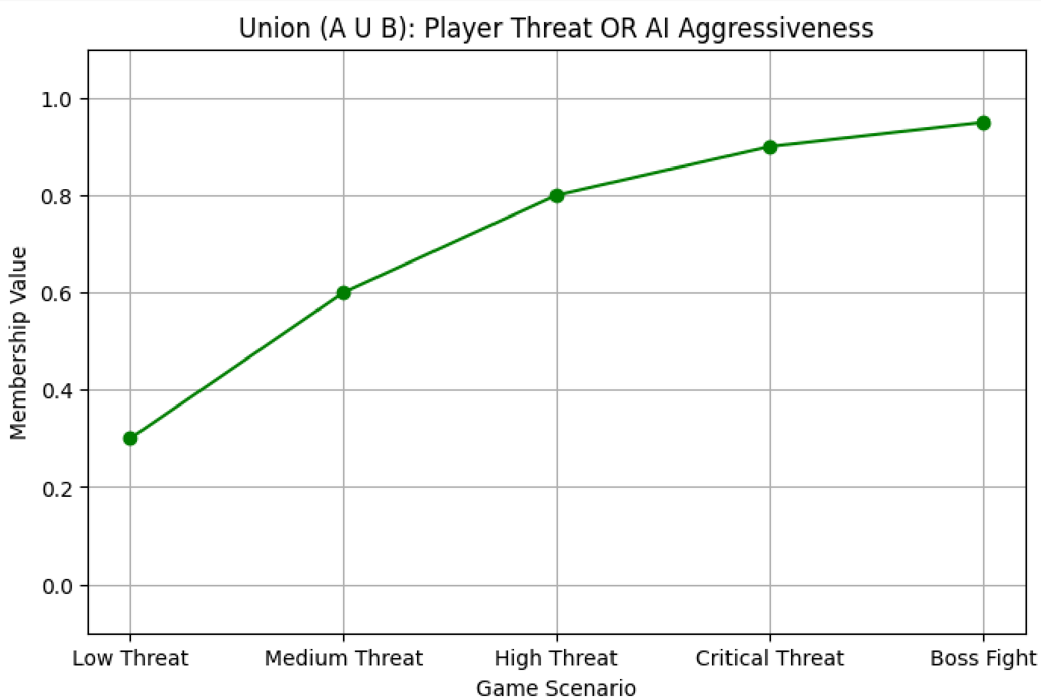
```python
# Plot Fuzzy Set A
plt.figure(figsize=(8, 5))
plt.plot(x, A, 'o-', color='blue')
plt.title("Fuzzy Set A: Player's Threat Level")
plt.xlabel('Game Scenario')
plt.ylabel('Membership Value')
plt.ylim(-0.1, 1.1)
plt.grid(True)
plt.show()
```




```python
# Plot Fuzzy Set B
plt.figure(figsize=(8, 5))
plt.plot(x, B, 'o-', color='red')
plt.title("Fuzzy Set B: AI's Aggressiveness")
plt.xlabel('Game Scenario')
plt.ylabel('Membership Value')
plt.ylim(-0.1, 1.1)
plt.grid(True)
plt.show()
```

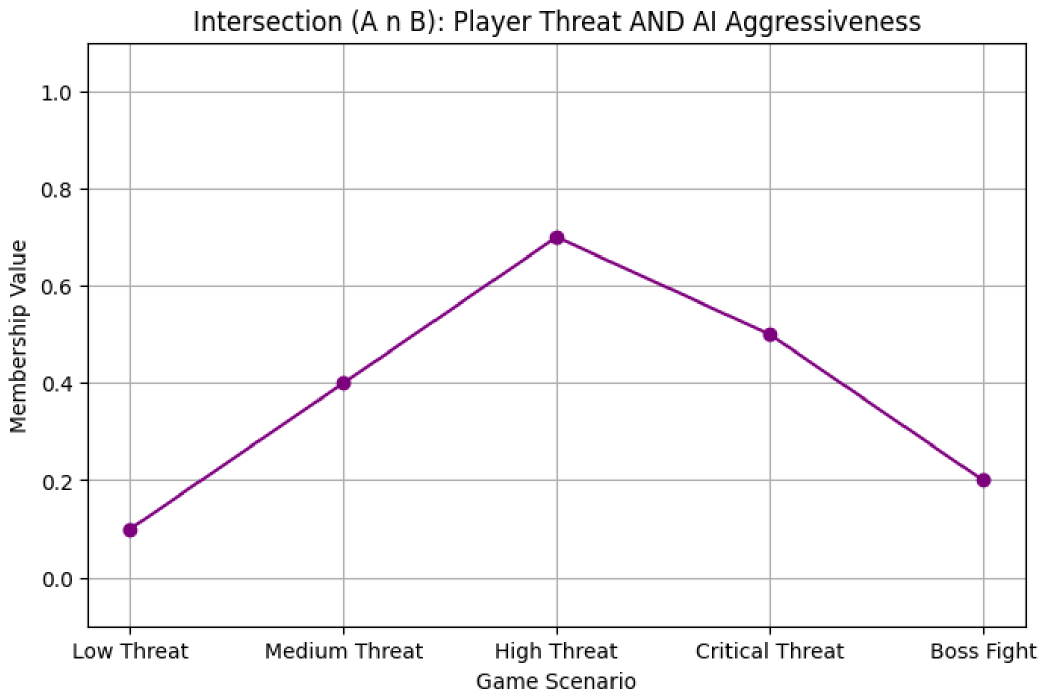



```
# Plot Union (A U B)
plt.figure(figsize=(8, 5))
plt.plot(x, A_union_B, 'o-', color='green')
plt.title("Union (A U B): Player Threat OR AI Aggressiveness")
plt.xlabel('Game Scenario')
plt.ylabel('Membership Value')
plt.ylim(-0.1, 1.1)
plt.grid(True)
plt.show()
```

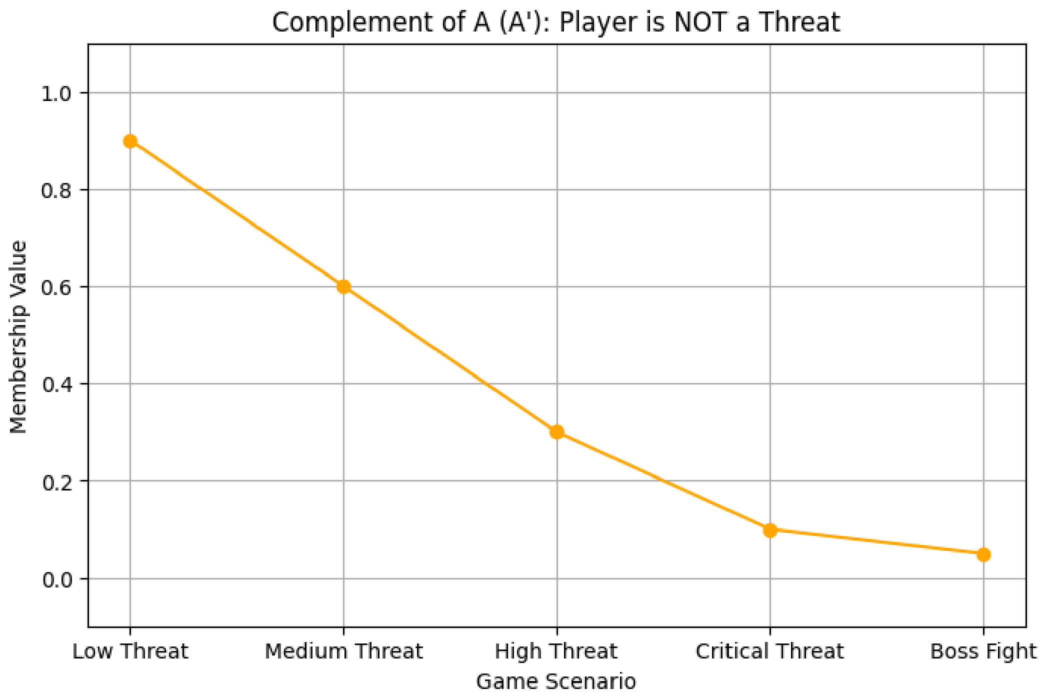



```
# Plot Intersection (A n B)
```

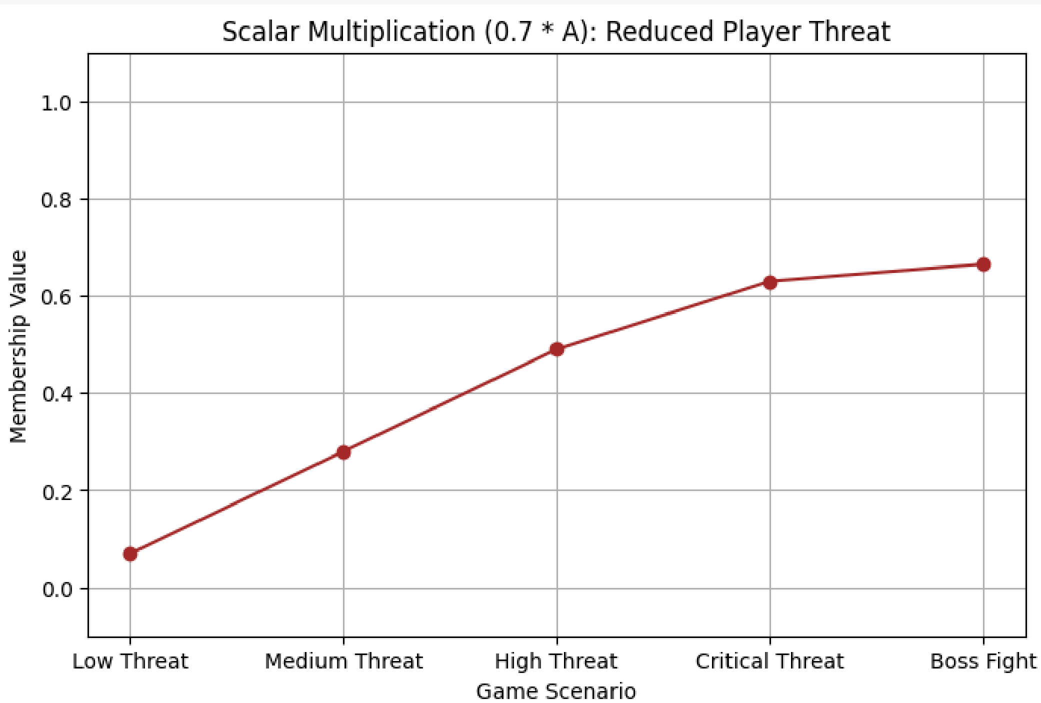
```python
plt.figure(figsize=(8, 5))
plt.plot(x, A_intersection_B, 'o-', color='purple')
plt.title("Intersection (A n B): Player Threat AND AI Aggressiveness")
plt.xlabel('Game Scenario')
plt.ylabel('Membership Value')
plt.ylim(-0.1, 1.1)
plt.grid(True)
plt.show()
```




```python
# Plot Complement of A (A')
plt.figure(figsize=(8, 5))
plt.plot(x, A_complement, 'o-', color='orange')
plt.title("Complement of A (A'): Player is NOT a Threat")
plt.xlabel('Game Scenario')
plt.ylabel('Membership Value')
plt.ylim(-0.1, 1.1)
plt.grid(True)
plt.show()
```

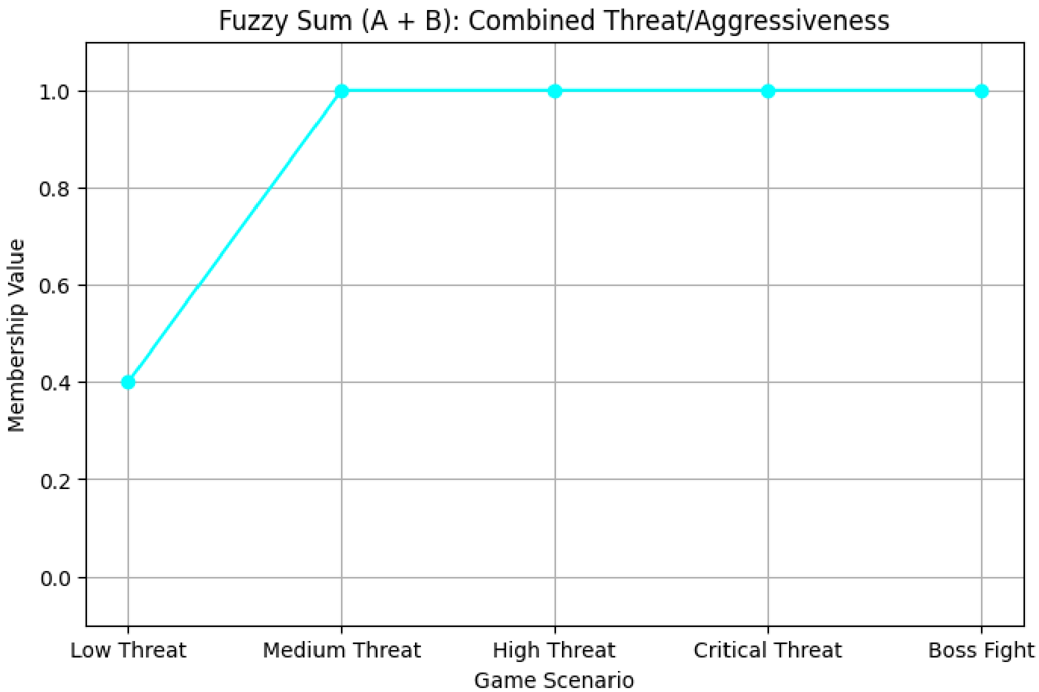



```
# Plot Scalar Multiplication (0.7 * A)
plt.figure(figsize=(8, 5))
plt.plot(x, A_scalar_multiplication, 'o-', color='brown')
plt.title("Scalar Multiplication (0.7 * A): Reduced Player Threat")
plt.xlabel('Game Scenario')
plt.ylabel('Membership Value')
plt.ylim(-0.1, 1.1)
plt.grid(True)
plt.show()
```




```
# Plot Fuzzy Sum (A + B)
plt.figure(figsize=(8, 5))
```

```
plt.plot(x, A_sum_B, 'o-', color='cyan')
plt.title("Fuzzy Sum (A + B): Combined Threat/Aggressiveness")
plt.xlabel('Game Scenario')
plt.ylabel('Membership Value')
plt.ylim(-0.1, 1.1)
plt.grid(True)
plt.show()
```




```
print("Fuzzy Set A (Player's Threat Level):", A)
print("Fuzzy Set B (AI's Aggressiveness):", B)
print("Union (A U B):", A_union_B)
print("Intersection (A n B):", A_intersection_B)
print("Complement of A (A'):", A_complement)
print("Scalar Multiplication (0.7 * A):", A_scalar_multiplication)
print("Fuzzy Sum (A + B):", A_sum_B)
```

```
Fuzzy Set A (Player's Threat Level): [0.1, 0.4, 0.7, 0.9, 0.95]
Fuzzy Set B (AI's Aggressiveness): [0.3, 0.6, 0.8, 0.5, 0.2]
Union (A U B): [0.3, 0.6, 0.8, 0.9, 0.95]
Intersection (A n B): [0.1, 0.4, 0.7, 0.5, 0.2]
Complement of A (A'): [0.9, 0.6, 0.30000000000000004, 0.09999999999999998,
0.050000000000000044]
Scalar Multiplication (0.7 * A): [0.06999999999999999, 0.27999999999999997,
0.48999999999999994, 0.63, 0.6649999999999999]
Fuzzy Sum (A + B): [0.4, 1, 1, 1, 1]
```

Conclusion:

Fuzzy set theory provides a powerful framework for dealing with uncertainty and vagueness, making it incredibly useful in fields like artificial intelligence, control systems, and decision-making. By allowing for degrees of membership, fuzzy sets can model human-like reasoning more effectively than traditional binary logic. 49_Prac_07