

1. Пусть $MA(x)$ – функция принадлежности нечеткого множества A , x из универсального множества I . Что означает запись $MA(x)=0$? Что означает запись $MA(x)=1$?

```
In [20]: # Функция принадлежности для нечеткого множества A
def MA(x):
    if x < 5:
        return 0 # элемент не принадлежит
    elif x >= 5 and x <= 10:
        return 1 # элемент полностью принадлежит
    else:
        return 0 # элемент не принадлежит

# Пример значений для демонстрации
values = [2, 6, 12]

# Проверка значений функции принадлежности
for x in values:
    membership_value = MA(x)
    print(f"MA({x}) = {membership_value}")
```

```
MA(2) = 0
MA(6) = 1
MA(12) = 0
```

2. Приведите пример нечеткого множества, заданного перечислением его элементов.

```
In [21]: # Пример нечеткого множества, заданного перечислением элементов
A = {
    2: 0.3, # степень принадлежности для элемента 2
    4: 0.7, # степень принадлежности для элемента 4
    8: 0.2, # степень принадлежности для элемента 8
    16: 0.1 # степень принадлежности для элемента 16
}

# Вывод нечеткого множества
print("Нечеткое множество A:")
for element, membership_value in A.items():
    print(f"Элемент: {element}, Значение функции принадлежности: {membership_val
```

```
Нечеткое множество A:
Элемент: 2, Значение функции принадлежности: 0.3
Элемент: 4, Значение функции принадлежности: 0.7
Элемент: 8, Значение функции принадлежности: 0.2
Элемент: 16, Значение функции принадлежности: 0.1
```

3. Приведите пример нечеткого множества, заданного аналитическим представлением функции принадлежности.

```
In [23]: import numpy as np
import matplotlib.pyplot as plt

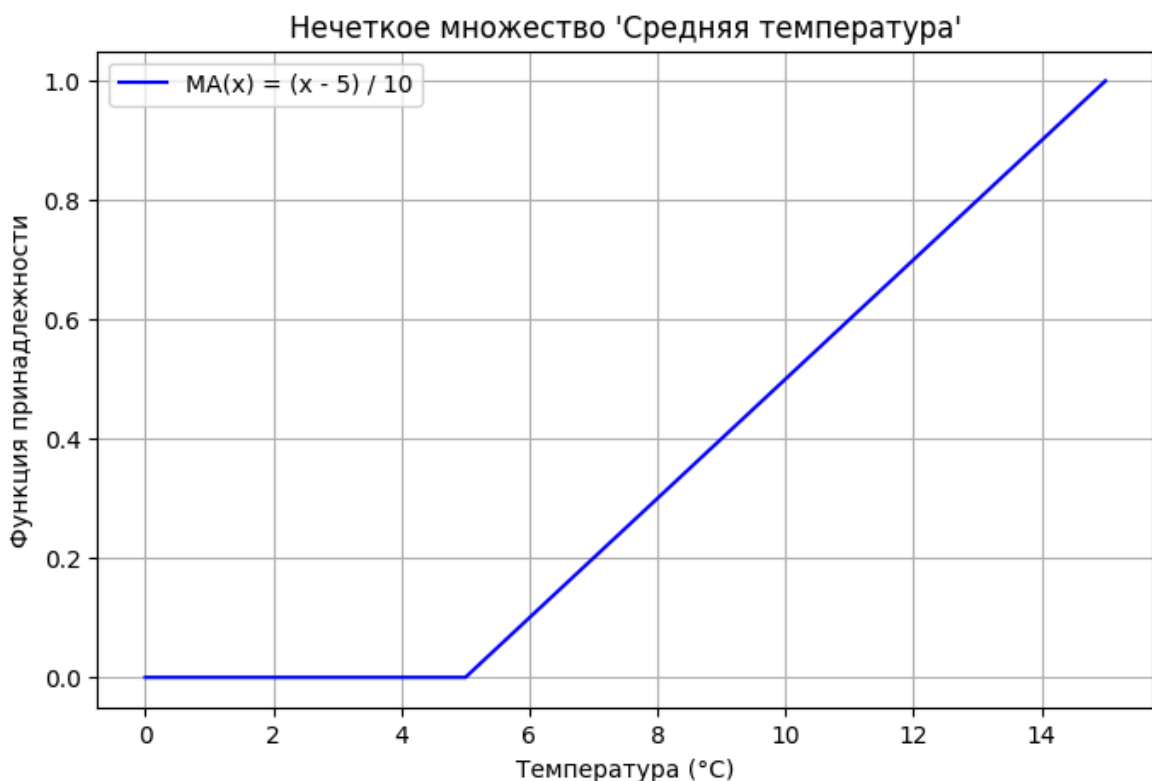
# Аналитическое представление функции принадлежности
```

```
def MA(x):
    return np.maximum(0, np.minimum((x - 5) / 10, 1))

# Массив значений для x
x = np.linspace(0, 15, 400)

# Значения функции принадлежности
membership_values = MA(x)

# Построение графика
plt.figure(figsize=(8, 5))
plt.plot(x, membership_values, 'b-', label='MA(x) = (x - 5) / 10')
plt.title("Нечеткое множество 'Средняя температура'")
plt.xlabel("Температура (°C)")
plt.ylabel("Функция принадлежности")
plt.grid(True)
plt.legend()
plt.show()
```



4. Приведите примеры стандартных функций принадлежности.

```
In [24]: import numpy as np
import matplotlib.pyplot as plt

# Функция треугольной формы
def triangular(x, a, b, c):
    return np.maximum(0, np.minimum(np.minimum((x - a) / (b - a), (c - x) / (c -

# Функция трапецевидной формы
def trapezoidal(x, a, b, c, d):
    return np.maximum(0, np.minimum(np.minimum((x - a) / (b - a), (d - x) / (d -

# Функция гауссиана
def gaussian(x, mu, sigma):
```

```

return np.exp(-((x - mu) ** 2) / (2 * sigma ** 2))

# Массив значений для x
x = np.linspace(0, 10, 400)

# Значения функций принадлежности
triangular_values = triangular(x, 2, 5, 8)
trapezoidal_values = trapezoidal(x, 2, 4, 6, 8)
gaussian_values = gaussian(x, 5, 1)

# Построение графиков
plt.figure(figsize=(12, 8))

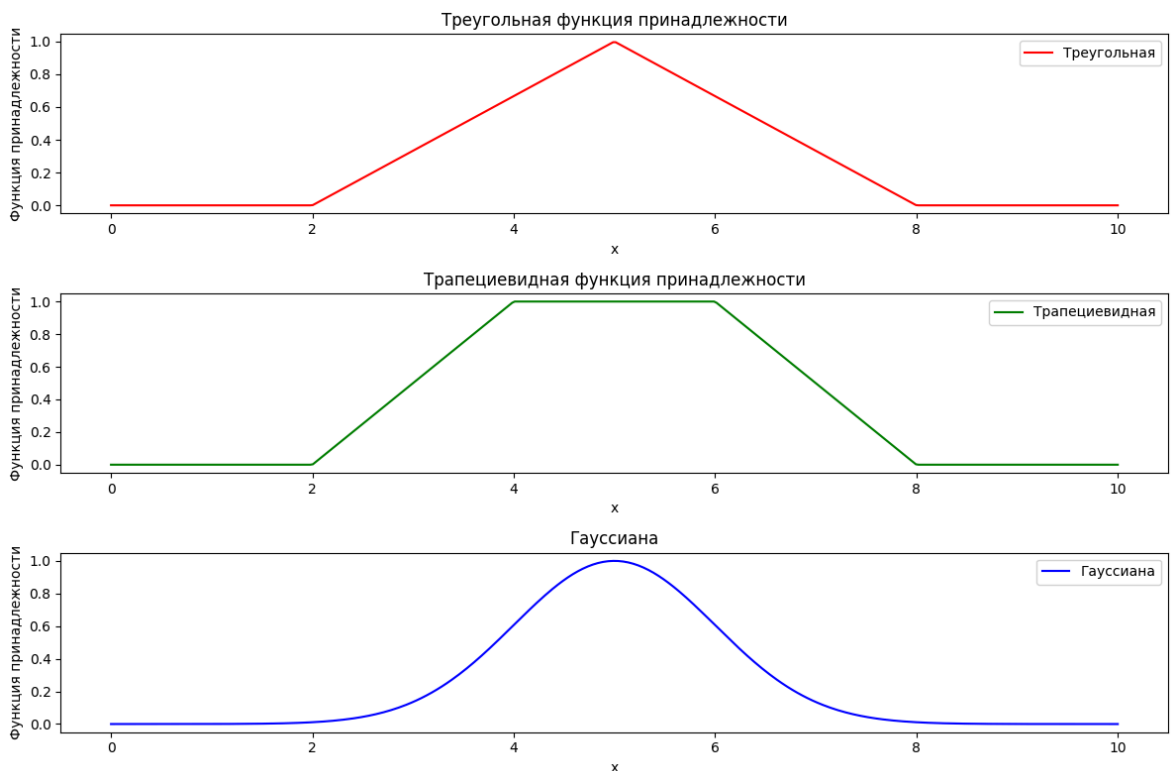
plt.subplot(3, 1, 1)
plt.plot(x, triangular_values, 'r-', label='Треугольная')
plt.title("Треугольная функция принадлежности")
plt.xlabel("x")
plt.ylabel("Функция принадлежности")
plt.legend()

plt.subplot(3, 1, 2)
plt.plot(x, trapezoidal_values, 'g-', label='Трапецевидная')
plt.title("Трапецевидная функция принадлежности")
plt.xlabel("x")
plt.ylabel("Функция принадлежности")
plt.legend()

plt.subplot(3, 1, 3)
plt.plot(x, gaussian_values, 'b-', label='Гауссиана')
plt.title("Гауссиана")
plt.xlabel("x")
plt.ylabel("Функция принадлежности")
plt.legend()

plt.tight_layout()
plt.show()

```



5. Приведите расчетную формулу для обратного к данному нечеткому числу.

```
In [9]: def reverse_fuzzy_number_triangular(a, b, c):
        """
        Функция для нахождения обратного треугольного нечеткого числа (a, b, c).
        Возвращает кортеж обратных значений.
        """
        if a > 0 and b > 0 and c > 0:
            return (1 / c, 1 / b, 1 / a)
        else:
            raise ValueError("Все значения (a, b, c) должны быть положительными и не равными нулю")

    def reverse_fuzzy_number_interval(a, b):
        """
        Функция для нахождения обратного интервала [a, b].
        Возвращает список обратных значений [1/b, 1/a].
        """
        if a > 0 and b > 0:
            return [1 / b, 1 / a]
        else:
            raise ValueError("Границы интервала должны быть положительными и не равными нулю")

    # Пример для треугольного нечеткого числа
    triangular_fuzzy = (2, 3, 4) # треугольное число
    triangular_reverse = reverse_fuzzy_number_triangular(*triangular_fuzzy)

    print(f"Исходное треугольное нечеткое число: {triangular_fuzzy}")
    print(f"Обратное треугольное нечеткое число: {triangular_reverse}")

    # Пример для интервала
    interval_fuzzy = [2, 4] # интервал
    interval_reverse = reverse_fuzzy_number_interval(*interval_fuzzy)

    print(f"\nИсходный интервал: {interval_fuzzy}")
    print(f"Обратный интервал: {interval_reverse}")
```

Исходное треугольное нечеткое число: (2, 3, 4)

Обратное треугольное нечеткое число: (0.25, 0.3333333333333333, 0.5)

Исходный интервал: [2, 4]

Обратный интервал: [0.25, 0.5]

6. На универсальном множестве всех натуральных чисел задано нечеткое множество $A = \{2:0.3; 4:0.7; 8:0.2; 16:0.1\}$. Найти дополнение.

```
In [10]: # Исходное нечеткое множество
A = {2: 0.3, 4: 0.7, 8: 0.2, 16: 0.1}

# Вычисление дополнения
A_complement = {x: round(1 - mu, 2) for x, mu in A.items()}

# Вывод результата
print("Исходное нечеткое множество A:")
print(A)

print("\nДополнение нечеткого множества A^c:")
print(A_complement)
```

Исходное нечеткое множество A:
{2: 0.3, 4: 0.7, 8: 0.2, 16: 0.1}

Дополнение нечеткого множества A^c :
{2: 0.7, 4: 0.3, 8: 0.8, 16: 0.9}

7. На универсальном множестве всех натуральных чисел заданы нечеткие множества $A = \{2:0.3; 4:0.7; 8:0.2; 16:0.1\}$ и $B = \{4:0.1; 8:0.3; 16:0.5; 32:0.8\}$. Найти объединение и пересечение.

```
In [11]: # Исходные нечеткие множества
A = {2: 0.3, 4: 0.7, 8: 0.2, 16: 0.1}
B = {4: 0.1, 8: 0.3, 16: 0.5, 32: 0.8}

# Вычисление объединения
A_union_B = {x: max(A.get(x, 0), B.get(x, 0)) for x in set(A.keys()).union(B.keys())}

# Вычисление пересечения
A_intersection_B = {x: min(A.get(x, 0), B.get(x, 0)) for x in set(A.keys()).intersection(B.keys())}

# Вывод результата
print("Объединение нечетких множеств A ∪ B:")
print(A_union_B)

print("\nПересечение нечетких множеств A ∩ B:")
print(A_intersection_B)
```

Объединение нечетких множеств $A \cup B$:
{32: 0.8, 2: 0.3, 4: 0.7, 8: 0.3, 16: 0.5}

Пересечение нечетких множеств $A \cap B$:
{8: 0.2, 16: 0.1, 4: 0.1}

8. На универсальном множестве всех натуральных чисел заданы нечеткие множества $A = \{2:0.3; 4:0.7; 8:0.2; 16:0.1\}$ и $B = \{4:0.1; 8:0.3; 16:0.5; 32:0.8\}$. Найти симметрическую разность.

```
In [12]: # Исходные нечеткие множества
A = {2: 0.3, 4: 0.7, 8: 0.2, 16: 0.1}
B = {4: 0.1, 8: 0.3, 16: 0.5, 32: 0.8}

# Вычисление симметрической разности
A_sym_diff_B = {}
all_keys = set(A.keys()).union(B.keys())

for x in all_keys:
    M_A_x = A.get(x, 0)
    M_B_x = B.get(x, 0)
    sym_diff = abs(M_A_x - M_B_x)
    A_sym_diff_B[x] = sym_diff

# Вывод результата
print("Симметрическая разность нечетких множеств A Δ B:")
print(A_sym_diff_B)
```

Симметрическая разность нечетких множеств $A \Delta B$:
{32: 0.8, 2: 0.3, 4: 0.6, 8: 0.09999999999999998, 16: 0.4}

9. На универсальном множестве всех натуральных чисел заданы нечеткие числа $A = \{2:0.3; 4:0.7; 8:0.2; 16:0.1\}$ и $B = \{4:0.1; 8:0.3; 16:0.5; 32:0.8\}$. Найти сумму.

```
In [13]: # Исходные нечеткие множества
A = {2: 0.3, 4: 0.7, 8: 0.2, 16: 0.1}
B = {4: 0.1, 8: 0.3, 16: 0.5, 32: 0.8}

# Вычисление суммы
A_sum_B = {}
all_keys = set(A.keys()).union(B.keys())

for x in all_keys:
    M_A_x = A.get(x, 0)
    M_B_x = B.get(x, 0)
    A_sum_B[x] = M_A_x + M_B_x

# Вывод результата
print("Сумма нечетких чисел A + B:")
print(A_sum_B)
```

Сумма нечетких чисел A + B:
{32: 0.8, 2: 0.3, 4: 0.7999999999999999, 8: 0.5, 16: 0.6}

10. На универсальном множестве всех натуральных чисел заданы нечеткие числа $A = \{2:0.3; 4:0.7; 8:0.2; 16:0.1\}$ и $B = \{4:0.1; 8:0.3; 16:0.5; 32:0.8\}$. Найти разность.

```
In [14]: # Исходные нечеткие множества
A = {2: 0.3, 4: 0.7, 8: 0.2, 16: 0.1}
B = {4: 0.1, 8: 0.3, 16: 0.5, 32: 0.8}

# Вычисление разности
A_minus_B = {}
all_keys = set(A.keys()).union(B.keys())

for x in all_keys:
    M_A_x = A.get(x, 0)
    M_B_x = B.get(x, 0)
    difference = M_A_x - M_B_x
    A_minus_B[x] = difference

# Вывод результата
print("Разность нечетких чисел A - B:")
print(A_minus_B)
```

Разность нечетких чисел A - B:
{32: -0.8, 2: 0.3, 4: 0.6, 8: -0.09999999999999998, 16: -0.4}

11. На универсальном множестве всех натуральных чисел заданы нечеткие числа $A = \{2:0.3; 4:0.7; 8:0.2; 16:0.1\}$ и $B = \{4:0.1; 8:0.3; 16:0.5; 32:0.8\}$. Найти произведение.

```
In [15]: # Исходные нечеткие множества
A = {2: 0.3, 4: 0.7, 8: 0.2, 16: 0.1}
B = {4: 0.1, 8: 0.3, 16: 0.5, 32: 0.8}

# Вычисление произведения
A_product_B = {}
```

```

all_keys = set(A.keys()).union(B.keys())

for x in all_keys:
    M_A_x = A.get(x, 0)
    M_B_x = B.get(x, 0)
    product = M_A_x * M_B_x
    A_product_B[x] = product

# Вывод результата
print("Произведение нечетких чисел A * B:")
print(A_product_B)

```

Произведение нечетких чисел A * B:
{32: 0.0, 2: 0.0, 4: 0.06999999999999999, 8: 0.06, 16: 0.05}

12. На универсальном множестве всех натуральных чисел задано нечеткое число A = {2:0.3; 4:0.7; 8:0.2; 16:0.1}. Найти обратное число.

```

In [16]: # Исходное нечеткое множество
A = {2: 0.3, 4: 0.7, 8: 0.2, 16: 0.1}

# Вычисление обратного числа
A_inverse = {}
for x, membership in A.items():
    if membership != 0:
        inverse_value = 1 / membership
    else:
        inverse_value = 0 # Undefined for membership = 0
    A_inverse[x] = inverse_value

# Вывод результата
print("Обратное число для нечеткого множества A:")
print(A_inverse)

```

Обратное число для нечеткого множества A:
{2: 3.3333333333333335, 4: 1.4285714285714286, 8: 5.0, 16: 10.0}

13. На универсальном множестве всех натуральных чисел заданы нечеткие множества A = {2:0.3; 4:0.7; 8:0.2; 16:0.1} и B = {4:0.1; 8:0.3; 16:0.5; 32:0.8}. Найти ближайшие четкие множества.

```

In [17]: # Исходные нечеткие множества
A = {2: 0.3, 4: 0.7, 8: 0.2, 16: 0.1}
B = {4: 0.1, 8: 0.3, 16: 0.5, 32: 0.8}

# Поиск ближайшего четкого множества
closest_set = {}
all_keys = set(A.keys()).union(B.keys())

for x in all_keys:
    M_A_x = A.get(x, 0)
    M_B_x = B.get(x, 0)
    if M_A_x > M_B_x:
        closest_set[x] = 'A'
    else:
        closest_set[x] = 'B'

# Вывод результата

```

```
print("Ближайшие четкие множества A и B:")
print(closest_set)
```

Ближайшие четкие множества A и B:
{32: 'B', 2: 'A', 4: 'A', 8: 'B', 16: 'B'}

14. На универсальном множестве всех натуральных чисел заданы нечеткие множества $A = \{2:0.3; 4:0.7; 8:0.2; 16:0.1\}$ и $B = \{4:0.1; 8:0.3; 16:0.5; 32:0.8\}$. Найти α -срезы с $\alpha=0.6$.

```
In [18]: # Исходные нечеткие множества
A = {2: 0.3, 4: 0.7, 8: 0.2, 16: 0.1}
B = {4: 0.1, 8: 0.3, 16: 0.5, 32: 0.8}

# Порог
alpha = 0.6

# Определение  $\alpha$ -срезов для множества A
A_alpha_slice = {x: M_A for x, M_A in A.items() if M_A >= alpha}

# Определение  $\alpha$ -срезов для множества B
B_alpha_slice = {x: M_B for x, M_B in B.items() if M_B >= alpha}

# Вывод результатов
print("α-срезы множества A с α = 0.6:")
print(A_alpha_slice)

print("α-срезы множества B с α = 0.6:")
print(B_alpha_slice)
```

α -срезы множества A с $\alpha = 0.6$:
{4: 0.7}
 α -срезы множества B с $\alpha = 0.6$:
{32: 0.8}

15. Привести пример лингвистической переменной.

```
In [19]: # Лингвистическая переменная - Температура
temperature = {
    'холодно': (0, 10), # диапазон для "холодно"
    'тепло': (10, 25), # диапазон для "тепло"
    'горячо': (25, 35) # диапазон для "горячо"
}

# Пример значений для демонстрации
values = [5, 15, 28]

# Проверка значений температуры и соответствие лингвистическим выражениям
for value in values:
    if value < temperature['холодно'][1]:
        print(f"Температура {value} °C: холодно")
    elif value < temperature['тепло'][1]:
        print(f"Температура {value} °C: тепло")
    else:
        print(f"Температура {value} °C: горячо")
```


Температура 5 °C: холодно

Температура 15 °C: тепло

Температура 28 °C: горячо