

## Лабораторная работа

### Нечеткая система управления

Задание 1. Сформируйте нечеткое множество с 10 элементами и вычислите его характеристики:

- точку перехода
  - мощность
  - носители
  - высоту
- определите является ли множество нормальным или субнормальным

*Пример.*

```
import numpy as np
import skfuzzy as fuzz
import matplotlib.pyplot as plt
A=[0.4, 0.4, 1, 0.2, 0.4, 0.2, 0.2, 0.2, 0.6, 0.8];
a=0.5 # точка перехода
print('Функции принадлежности нечеткого множества A')
print(A)
```

```
Функции принадлежности нечеткого множества A
[0.4, 0.4, 1, 0.2, 0.4, 0.2, 0.2, 0.2, 0.6, 0.8]
```

```
maxzn=max(A)
print('Максимальное значение функции принадлежности- высота НМ')
print (maxzn)
```

```
Максимальное значение функции принадлежности- высота НМ
1
```

```

setA = [0.1, 0.2, 0.8, 0.5, 0.9, 1 ];
Sum=sum(setA)
print("мощность нечеткого множества:", Sum)
#носители нечеткого множества
print ("Носители нечеткого множества:", len(setA))
##Определим есть ли точка перехода в нечетком множестве и вернем номер элемента
point=setA.index(.5)+1
print ("Точка перехода:", point)
#высота нечеткого множества
MaxElem=max(setA)
print ("Высота нечеткого множества:", MaxElem)
#Определим является ли множество нормальным
if MaxElem>=1:
    a="нормальное"
    print ("Нечеткое множество:", a)

мощность нечеткого множества: 3.5
Носители нечеткого множества: 6
Точка перехода: 4
Высота нечеткого множества: 1
Нечеткое множество: нормальное

```

Задание 2. Сформируйте два нечетких множества А и В выполните операции:

- пересечения
- объединения
- дополнения
- разности
- симметричной разности
- алгебраической разности и др.

*Пример. Операция объединения двух нечетких множеств*

```

A = dict()
B = dict()
Y = dict()

A = {"a": 0.2, "b": 0.3, "c": 0.6, "d": 0.6}
B = {"a": 0.9, "b": 0.9, "c": 0.4, "d": 0.5}

print('The First Fuzzy Set is :', A)
print('The Second Fuzzy Set is :', B)

for A_key, B_key in zip(A, B):
    A_value = A[A_key]
    B_value = B[B_key]

    if A_value > B_value:
        Y[A_key] = A_value
    else:
        Y[B_key] = B_value

print('Fuzzy Set Union is :', Y)

```

Выход:

```
The First Fuzzy Set is : {'a': 0.2, 'b': 0.3, 'c': 0.6, 'd': 0.6}
The Second Fuzzy Set is : {'a': 0.9, 'b': 0.9, 'c': 0.4, 'd': 0.5}
Fuzzy Set Union is : {'a': 0.9, 'b': 0.9, 'c': 0.6, 'd': 0.6}
```

**Пример. Операция пересечения двух нечетких множеств**

```
A = dict()
B = dict()
Y = dict()

A = {"a": 0.2, "b": 0.3, "c": 0.6, "d": 0.6}
B = {"a": 0.9, "b": 0.9, "c": 0.4, "d": 0.5}

print('The First Fuzzy Set is :', A)
print('The Second Fuzzy Set is :', B)

for A_key, B_key in zip(A, B):
    A_value = A[A_key]
    B_value = B[B_key]

    if A_value < B_value:
        Y[A_key] = A_value
    else:
        Y[B_key] = B_value
print('Fuzzy Set Intersection is :', Y)
```

Выход:

```
The First Fuzzy Set is : {'a': 0.2, 'b': 0.3, 'c': 0.6, 'd': 0.6}
The Second Fuzzy Set is : {'a': 0.9, 'b': 0.9, 'c': 0.4, 'd': 0.5}
Fuzzy Set Intersection is : {'a': 0.2, 'b': 0.3, 'c': 0.4, 'd': 0.5}
```

**Пример. Операция дополнения для элементов множества A**

```
A = dict()
Y = dict()

A = {"a": 0.2, "b": 0.3, "c": 0.6, "d": 0.6}

print('The Fuzzy Set is :', A)

for A_key in A:
    Y[A_key] = 1 - A[A_key]

print('Fuzzy Set Complement is :', Y)

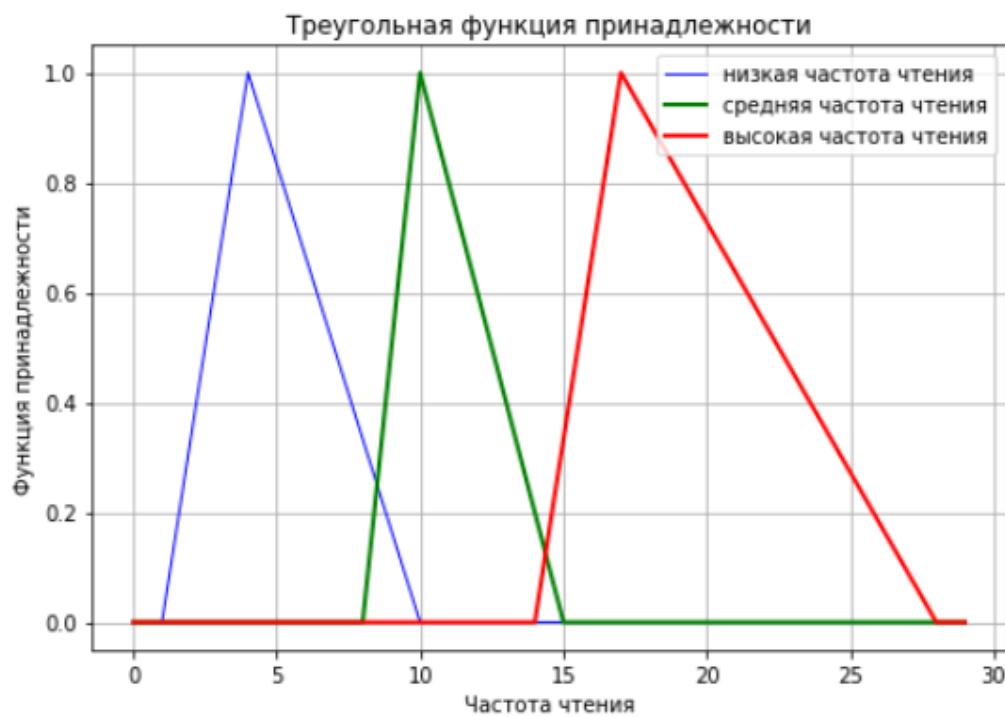
The Fuzzy Set is : {'a': 0.2, 'b': 0.3, 'c': 0.6, 'd': 0.6}
Fuzzy Set Complement is : {'a': 0.8, 'b': 0.7, 'c': 0.4, 'd': 0.4}
```

### Задание 3.

Постройте функции принадлежности для одного нечеткого множества A

- треугольную
- трапециевидную
- сигмоидальную
- гауссову

```
import numpy as np
import skfuzzy as fuzz
import matplotlib.pyplot as plt
x_qual = np.arange(0, 30, 1)
qual_lo = fuzz.trimf(x_qual, [1, 4, 10])
qual_md = fuzz.trimf(x_qual, [8, 10, 15])
qual_hi = fuzz.trimf(x_qual, [14, 17, 28])
fig, (ax0) = plt.subplots(nrows=1, figsize=(7, 5))
#ax0.set_title('График функции принадлежности нечеткого множества видеокарт по частоте чтения')
ax0.plot(x_qual, qual_lo, 'b', linewidth=1, label = 'низкая частота чтения')
ax0.plot(x_qual, qual_md, 'g', linewidth=2, label = 'средняя частота чтения')
ax0.plot(x_qual, qual_hi, 'r', linewidth=2, label = 'высокая частота чтения')
ax0.legend()
plt.title('Треугольная функция принадлежности')
plt.grid(True)
plt.xlabel('Частота чтения')
plt.ylabel('Функция принадлежности')
plt.tight_layout()
```



```

qual_lo=fuzz.sigmf(x_qual, 1, 0.6)

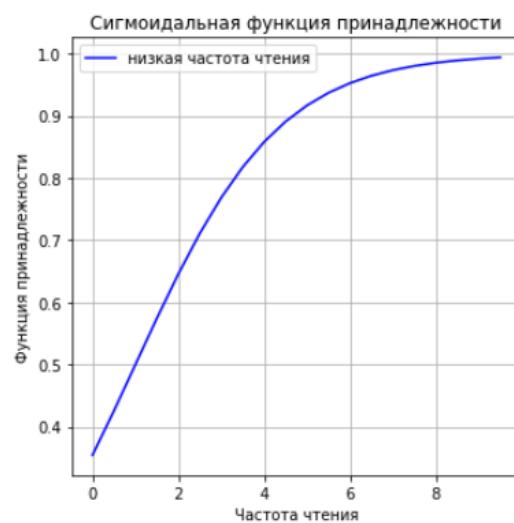
#qual_md = fuzz.trapmf(x_qual, [8, 10, 12, 13])
#qual_hi = fuzz.trapmf(x_qual, [14, 16, 18, 20])

fig, (ax0) = plt.subplots(nrows=1, figsize=(5, 5))

ax0.plot(x_qual, qual_lo, 'b', linewidth=1.5, label ='низкая частота чтения')

#ax0.plot(x_qual, qual_md, 'g', linewidth=1.5, label ='средняя частота чтения')
#ax0.plot(x_qual, qual_hi, 'r', linewidth=1.5, label ='высокая частота чтения')
#ax0.set_title('Нечёткое множество видеокарт по частоте чтения')
ax0.legend()
plt.tight_layout()
plt.title('Сигмоидальная функция принадлежности')
plt.grid(True)
plt.xlabel('Частота чтения')
plt.ylabel('Функция принадлежности')
ax0.legend()
plt.tight_layout()

```



```

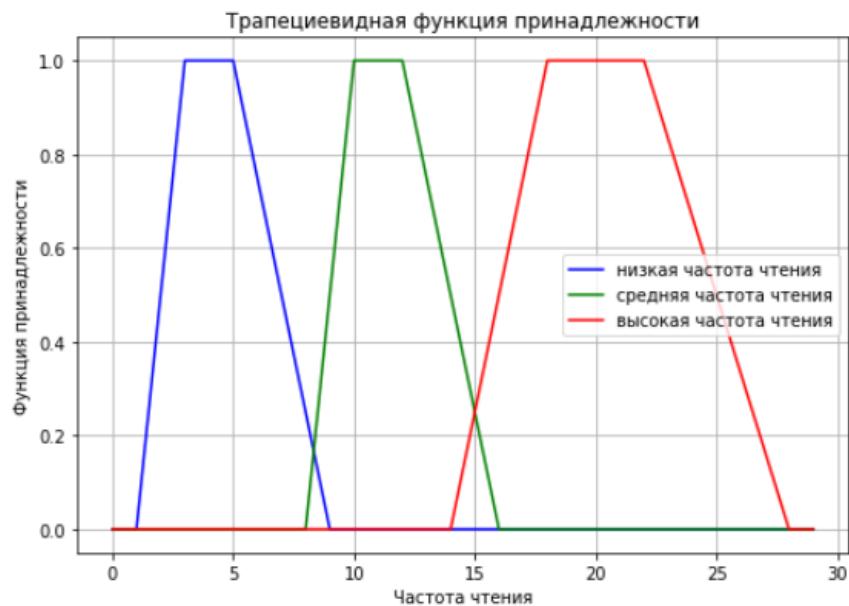
x_qual = np.arange(0, 30, 1)

qual_lo = fuzz.trapmf(x_qual, [1, 3, 5, 9])
qual_md = fuzz.trapmf(x_qual, [8, 10, 12, 16])
qual_hi = fuzz.trapmf(x_qual, [14, 18, 22, 28])

fig, (ax0) = plt.subplots(nrows=1, figsize=(7, 5))

ax0.plot(x_qual, qual_lo, 'b', linewidth=1.5, label ='низкая частота чтения')
ax0.plot(x_qual, qual_md, 'g', linewidth=1.5, label ='средняя частота чтения')
ax0.plot(x_qual, qual_hi, 'r', linewidth=1.5, label ='высокая частота чтения')
ax0.set_title('Трапециевидная функция принадлежности')
ax0.legend()
plt.tight_layout()
plt.grid(True)
plt.xlabel('Частота чтения')
plt.ylabel('Функция принадлежности')

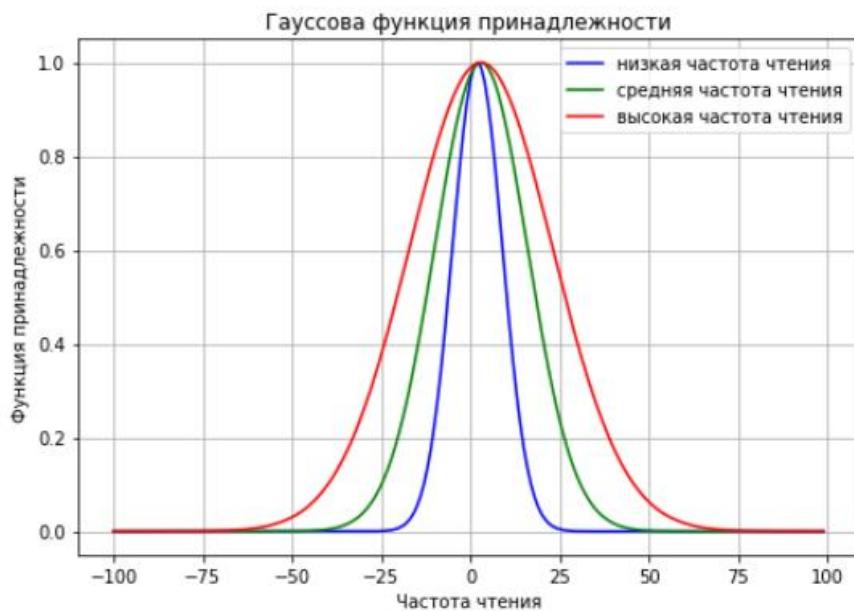
```



```

x_qual = np.arange(-100, 100, 1)
qual_lo = fuzz.gaussmf (x_qual, 2, 7)
qual_md = fuzz.gaussmf (x_qual, 3, 13)
qual_hi = fuzz.gaussmf (x_qual, 3, 20)
fig, (ax0) = plt.subplots(nrows=1, figsize=(7, 5))
ax0.plot(x_qual, qual_lo, 'b', linewidth=1.5, label = 'низкая частота чтения')
ax0.plot(x_qual, qual_md, 'g', linewidth=1.5, label = 'средняя частота чтения')
ax0.plot(x_qual, qual_hi, 'r', linewidth=1.5, label = 'высокая частота чтения')
ax0.set_title('Гауссова функция принадлежности')
ax0.legend()
plt.grid(True)
plt.xlabel('Частота чтения')
plt.ylabel('Функция принадлежности')
ax0.legend()
plt.tight_layout()

```



**Задание 4.** Разработайте нечеткую систему управления. Опишите входные данные и термы (3 входные переменные).

Опишите выходную переменную и терм множество (1 выходная переменная) приведите базу знаний.

Приведите комментарии к программному коду и базе знаний

**Задание 5.** Разработайте графический интерфейс для нечеткой системы управления с использованием Django.

Приведите рисунки работы с приложением

Для сдачи лабораторной работы:

Предоставьте программный код Python  
Приложение Django

**Контрольные вопросы для защиты лабораторной работы:**