

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ НАЦІОНАЛЬНОМУ  
УНІВЕРСИТЕТУ “ЛЬВІВСЬКА ПОЛІТЕХНІКА”**

**Кафедра систем штучного інтелекту**

**Лабораторна робота №2**

з дисципліни

«Дискретна математика»

**Виконала:**

студентка групи КН-114

Церковник Оксана

**Викладач:**

Мельникова Н.І.

**Львів – 2019р.**

## Тема: Моделювання основних операцій для числових множин.

**Мета роботи:** Ознайомитись на практиці із основними поняттями теорії множин, навчитись будувати діаграми Ейлера-Венна операцій над множинами, використовувати закони алгебри множин, освоїти принцип включень-виключень для двох і трьох множин та комп'ютерне подання множин.

### Теоретичні відомості:

Закони асоціативності	
$A \cup (B \cap C) = (A \cup B) \cap C$	$A \cap (B \cup C) = (A \cap B) \cup C$
Закони комутативності	
$A \cup B = B \cup A$	$A \cap B = B \cap A$
Закони тотожності	
$A \cup \emptyset = A$	$A \cap U = A$
Закони домінування	
$A \cup U = U$	$A \cap \emptyset = \emptyset$
Закони ідемпотентності	
$A \cup A = A$	$A \cap A = A$
Закони дистрибутивності	
$A \cap (B \cup C) = (A \cap B) \cup (A \cap C)$	$A \cup (B \cap C) = (A \cup B) \cap (A \cup C)$
Закони поглинання	
$(A \cup B) \cap A = A$	$(A \cap B) \cup A = A$
Закони доповнення	
$A \cup \bar{A} = U$	$A \cap \bar{A} = \emptyset$
$\bar{U} = \emptyset$	$\bar{\emptyset} = U$
$\overline{\bar{A}} = A$	$\overline{\bar{A}} = A$
Закони де Моргана	
$\overline{(A \cup B)} = \bar{A} \cap \bar{B}$	$\overline{(A \cap B)} = \bar{A} \cup \bar{B}$

Індивідуальний варіант: 13

### Варіант №13

1. Для даних скінчених множин  $A = \{1, 2, 3, 4, 5, 6, 7\}$ ,  $B = \{4, 5, 6, 7, 8, 9, 10\}$ ,  $C = \{2, 4, 6, 8, 10\}$  та універсаму  $U = \{1, 2, 3, 4, 5, 6, 7, 8, 9, 10\}$  знайти множину, яку задано за допомогою операцій: а)  $A \cap (B \cup C)$ ; б)  $\overline{B \Delta C}$ . Розв'язати, використовуючи комп'ютерне подання множин. Розв'язати, використовуючи комп'ютерне подання множин.

2. На множинах задачі 1 побудувати булеан множини  $C \setminus (B \setminus \overline{C}) \cap A$ . Знайти його потужність.

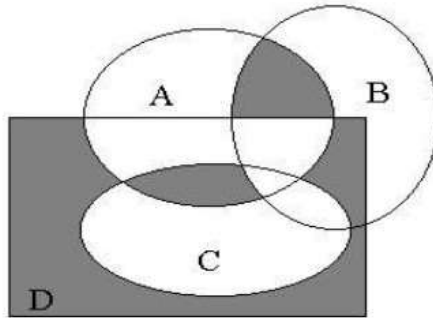
3. Нехай маємо множини:  $N$  – множина натуральних чисел,  $Z$  – множина цілих чисел,  $Q$  – множина раціональних чисел,  $R$  – множина дійсних чисел;  $A, B, C$  – будь-які множини. Перевірити які твердження є вірними (в останній задачі у випадку невірного твердження достатньо навести контрприклад, якщо твердження вірне – навести доведення):

- а)  $\emptyset \in \{1, 2, 3\}$ ;      б)  $Z \subset R$ ;  
 в)  $Q \cup Z = Q$ ;      г)  $R \setminus Z \subset R \setminus N$ ;  
 д) якщо  $A \subset B$ , то  $A \cap C \subset B \cap C$ .

4. Логічним методом довести тотожність:  $\overline{A \cup B} \cap A = \emptyset$ .

5. Зобразити на діаграмі Ейлера-Венна множину:  $(B \cup C) \Delta A \setminus (B \cap C)$ .

6. Множину зображено на діаграмі. Записати її за допомогою операцій.



7. Спростити вигляд множини, яка задана за допомогою операцій, застосовуючи закони алгебри множин (у відповідь множини можуть входити не більше одного разу):  $(A \cap \overline{B}) \cup (A \cap B \cap C) \cup \overline{A \cap C}$ .

8. Зі 100 студентів англійську мову вивчають 28 студентів, німецьку – 30, французьку – 42, англійську і французьку – 10, англійську і німецьку – 8, німецьку і французьку – 5, всі 3 мови студіюють троє. Скільки студентів не вивчають жодної із цих трьох мов?

13. Ввести з клавіатури множину дійсних чисел. Реалізувати операцію доповнення до цієї множини. Реалізувати програмно побудову булеану цієї множини. Усі результати виконання вивести на екран.

Розв'язання:

1 )

$$A = \{1, 2, 3, 4, 5, 6, 7\}$$

$$B = \{4, 5, 6, 7, 8, 9, 10\}$$

$$C = \{2, 4, 6, 8, 10\}$$

$$U = \{1, 2, 3, 4, 5, 6, 7, 8, 9, 10\}$$

$$A) A \cap (B \cup C) = \{2, 4, 5, 6, 7\}$$

$$B \cup C = \{2, 4, 5, 6, 7, 8, 9, 10\}$$

Комп'ютерне подання:  $\{0, 1, 0, 1, 1, 1, 1, 0, 0, 0\}$

Б)

$$\overline{B \Delta C} = \{1, 3, 4, 6, 8, 10\}$$

$$B \Delta C = \{2, 5, 7, 9\}$$

Комп'ютерне подання :  $\{1, 0, 1, 1, 0, 1, 0, 1, 0, 1\}$

2 )

$$A = \{1, 2, 3, 4, 5, 6, 7\}$$

$$B = \{4, 5, 6, 7, 8, 9, 10\}$$

$$C = \{2, 4, 6, 8, 10\}$$

$$U = \{1, 2, 3, 4, 5, 6, 7, 8, 9, 10\}$$

$$\overline{C} = \{1, 3, 5, 7, 9\}$$

$$B \setminus \overline{C} = \{4, 6, 8, 10\}$$

$$C \setminus (B \setminus \overline{C}) = \{2\}$$

$$C \setminus (B \setminus \overline{C}) \cap A = \{2\}$$

$$\text{Булеан} : \{\{\emptyset\}, \{2\}\}$$

$$P(C \setminus (B \setminus \overline{C}) \cap A) = 2$$

3 )

- a) Неправильне
- b) Правильне
- c) Неправильне
- d) Правильне
- e) Правильне

4)

$$\overline{(A \cup B)} \cap A = \emptyset$$

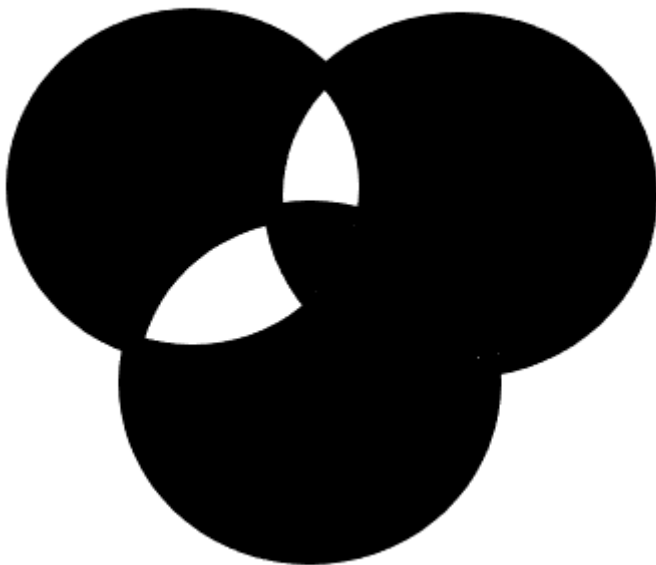
$$\bar{A} \cap \bar{B} \cap A = \emptyset$$

$$\emptyset \cap \bar{B} = \emptyset$$

$$\emptyset$$

5)

$$(B \cup C) \Delta A \setminus (B \cap C)$$



6)

$$(D \setminus (A \cap C)) \cup ((A \cap C) \setminus B) \cup ((A \cap B) \setminus D)$$

7)

$$\begin{aligned} (A \cap \bar{B}) \cup (A \cap B \cap C) \cup (\bar{A} \cap \bar{C}) &= (A \cap \bar{B}) \cup (A \cap B \cap C) \cup (\bar{A} \cup \bar{C}) = A \cap (\bar{B} \cup (B \cap C)) \cup (\bar{A} \cup \bar{C}) = A \cap ((\bar{B} \cup B) \cap (\bar{B} \cup C)) \cup (\bar{A} \cup \bar{C}) \\ &= A \cap (U \cap (\bar{B} \cup C)) \cup (\bar{A} \cup \bar{C}) = (A \cap \bar{B}) \cup (A \cap C) \cup (\bar{A} \cup \bar{C}) \\ \bar{A} \cup (A \cap C) \cup \bar{C} \cup (A \cap \bar{B}) &= ((\bar{A} \cup A) \cap (\bar{A} \cup C)) \cup \bar{C} \cup (A \cap \bar{B}) = (\bar{A} \cup C) \cup \bar{C} \cup (A \cap \bar{B}) = U \cup \bar{A} \cup (A \cap \bar{B}) = U \end{aligned}$$

8)



$$100 - (20 + 13 + 30 + 5 + 2 + 7 + 3) = 100 - 80 = 20$$

Відповідь: 20.

Додаток 2.

```
#include<iostream>
```

```
#include <cstdlib>
```

```
#include <string>
```

```
#include <stdlib.h>
```

```
#include <cmath>
```

```
#include <limits>
```

```
using namespace std;
```

```
void SubPrint(double *a, int n, int i)
```

```
{
```

```
    if (n)
```

```
    {
```

```
        if (n & 1)
```

```
            std::cout << " " << a[i] << " ";
```

```
            SubPrint(a, n >> 1, i + 1);
```

```
    }
```

```
}
```

```
void MainPrint(double* array, int size)
```

```
{
```

```
    int r, i;
```

```
    std::cout<<"{"<<"";
```

```
    r = 1 << size;
```

```
    for (i = 0; i < r; i++)
```

```
    {
```



```
SubPrint(array, i, 0);
```

```
if(i == r -1 )
```

```
{
```

```
    std::cout<<"}"<<"";
```

```
}
```

```
else
```

```
{
```

```
    std::cout<<"}, " << endl;
```

```
    std::cout<<"{";
```

```
}
```

```
}
```

```
}
```

```
void PrintArray(double *a, int n, string message)
```

```
{
```

```
    std::cout<<message << ": {"<<"";
```

```
for(int i = 0; i < n; i++)
```

```
{
```

```
    cout << a[i];
```

```
    if(i < n - 1)
    {
        cout << ", ";
    }
}

cout << "}" << endl;
}
```

```
double* GetUnivesum(double min, double max, double
step)
{
    int count = 2 + (int)(max - min)/0.1;
    double* univesum = new double[count];
    univesum[0] = min;
    double univesumValue = min;
    int index = 1;

    for(univesumValue = min + 0.1; univesumValue < max;
univesumValue += 0.1)
    {
        univesum[index] = univesumValue;
```

```
        index++;
    }

    return universum;
}

double* GetArray(double min, double max, int size)
{
    double* array = new double[size];
    cout << "Enter the value in range from " << min << " to "
    << max << endl;
    for(int j = 0; j < size; j++)
    {
        while(true)
        {
            double value;
            cin >> value;

            while(cin.fail())
            {
                cin.clear();
```

```

cin.ignore(numeric_limits<streamsize>::max(),'\n');

        cout << "Please, enter the Decimal Number from "
<< min << " to " << max << ": ";

        cin >> value;

    }

    if(value <= max && value >= min)
    {
        array[j] = value;
        break;
    }
    else
    {
        cout << "Please, enter the Decimal Number from "
<< min << " to " << max << ": ";

    }

}

return array;
}

```

```
bool relativeToleranceCompare(double x, double y)
{
    double maxXY = std::max( std::fabs(x) , std::fabs(y) ) ;
    return std::fabs(x - y) <=
std::numeric_limits<double>::epsilon()*maxXY ;
}
```

```
void BuildOpositArray(double *universum, int count,
double *array, int size)
{
    double* opositeArray = new double[count];
    int k = 0;
    int opositSize = 0;
    for(int i = 0; i < count; i++)
    {
        double val = universum[i];
        for (int n = 0; n < size; n++)
        {
            if (relativeToleranceCompare(val, array[n]))
            {
                break;
            }
        }
    }
}
```

```
        else if(n == size -1)
        {
            opositeArray[k] = val;
            opositSize++;
            k++;
        }
    }
}
```

```
    PrintArray(opositeArray, opositSize, "Oposit array");
}
```

```
int main()
{
    int size;

    double min = 0;
    double max = 1;
    double step = 0.1;
    int count = 2 + (int)(max - min)/0.1;
    double* universum;
    universum = GetUniversum(min, max, step);
```

```
PrintArray(universum, count, "Universum array");

cout << "Enter array size : " << endl;
cin >> size;
double* array;
array = GetArray(min, max, size);
PrintArray(array, size, "Your array is");
BuildOpositArray(universum, count, array, size);
MainPrint(array, size);

return 0;
}
```

```

Universum array: {0, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1}
Enter array size :
5
Enter the value in range from 0 to 1
0.1
0.3
0.4
0.6
0.9
Your array is: {0.1, 0.3, 0.4, 0.6, 0.9}
Oposit array: {0, 0.2, 0.5, 0.7, 0.8, 1}
{},
{ 0.1 },
{ 0.3 },
{ 0.1 0.3 },
{ 0.4 },
{ 0.1 0.4 },
{ 0.3 0.4 },
{ 0.1 0.3 0.4 },
{ 0.6 },
{ 0.1 0.6 },
{ 0.3 0.6 },
{ 0.1 0.3 0.6 },
{ 0.4 0.6 },
{ 0.1 0.4 0.6 },
{ 0.3 0.4 0.6 },
{ 0.1 0.3 0.4 0.6 },
{ 0.9 },
{ 0.1 0.9 },
{ 0.3 0.9 },
{ 0.1 0.3 0.9 },
{ 0.4 0.9 },
{ 0.1 0.4 0.9 },
{ 0.3 0.4 0.9 },
{ 0.1 0.3 0.4 0.9 },
{ 0.6 0.9 },
{ 0.1 0.6 0.9 },
{ 0.3 0.6 0.9 },
{ 0.1 0.3 0.6 0.9 },
{ 0.4 0.6 0.9 },
{ 0.1 0.4 0.6 0.9 },
{ 0.3 0.4 0.6 0.9 },
{ 0.1 0.3 0.4 0.6 0.9 }
Process returned 0 (0x0)   execution time : 20.057 s
Press any key to continue.

```

Висновок: на цій роботі я ознайомилась на практиці із основними поняттями теорії множин, навчилась будувати діаграми Ейлера-Венна операцій над множинами, використовувати закони алгебри множин, освоїла принцип включень-виключень для двох і трьох множин та комп'ютерне подання множин.