

**Міністерство освіти і науки Україн**

**Національний університет  
“Львівська політехніка”**

**Кафедра систем штучного інтелекту**

**Лабораторна робота № 2**

**з дисципліни**

**«Дискретна математика»**

**Виконав:**

студент групи КН-115

Поставка Маркіян

**Викладач:**

Мельникова Н. І.

## Тема: “Моделювання основних операцій для числових множин”

**Мета роботи:** Ознайомитись на практиці із основними 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.

### Додаток 1:

Варіант 11.

Постановка задачі:

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

2. На множинах задачі 1 побудувати булеан множини  $(\overline{C} \Delta B) \cap A$ . 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)  $\{4, 5\} \subset \{\{1\}, 2, 3, 4, 5\}$ ; б)  $N \in R$ ;

в)  $Q \cup N \subset N$ ; г)  $Q \setminus Z \subset R$ ;

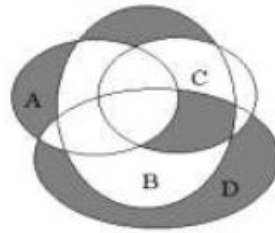
д) якщо  $A \subset B$  і  $B \subset \overline{C}$ , то  $A \cap C = \emptyset$ .

4. Логічним методом довести тотожність:

$$A \setminus (B \cap C) = (A \setminus B) \cup (A \setminus C).$$

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

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):

$$(A \cup B) \cap \overline{C} \cup (\overline{A \cap B} \cap C) \cup (A \cap B \cap C).$$

8. У групі 35 студентів. З них 20 відвідують курси англійської мови, 11 німецької, а 10 студентів не відвідують жодних курсів. Скільки студентів відвідують лише курси англійської мови?

Рішення:

1.

Комп'ютерно подамо множини:

$$A = 1111111000;$$

$$B = 0001111111;$$

$$C = 1010101010.$$

a)  $A \cap (B \cup C)$

B	0	0	0	1	1	1	1	1	1	1
C	1	0	1	0	1	0	1	0	1	0
$B \cup C$	1	0	1	1	1	1	1	1	1	1
A	1	1	1	1	1	1	1	0	0	0
$A \cap (B \cup C)$	1	0	1	1	1	1	1	0	0	0

б)  $\text{not}(B) \Delta \text{not}(C)$

$\text{not}(B)$	1	1	1	0	0	0	0	0	0	0
$\text{not}(C)$	0	1	0	1	0	1	0	1	0	1
$\text{not}(B) \Delta \text{not}(C)$	1	0	1	1	0	1	0	1	0	1

2.

$$F = (\text{not}(C) \Delta B) \cap A = 0100101000$$

$$F = \{2, 5, 7\}$$

$P(F) = \{\{\}, \{2\}, \{5\}, \{7\}, \{2, 5\}, \{2, 7\}, \{5, 7\}, \{2, 5, 7\}\}$  – Булеан множини  $F$

$$|P(F)| = 2^3 = 8 \text{ – Потужність булеану.}$$

3.

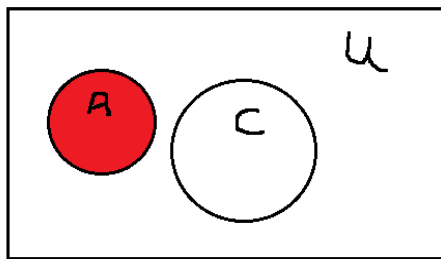
а)  $\{4, 5\} \subset \{1, 2, 3, 4, 5\} \rightarrow \text{True}$

б)  $N \in R \rightarrow \text{True}$

в)  $Q \cup n \subset N \rightarrow \text{False}$

г)  $Q \setminus Z \subset R \rightarrow \text{True}$

д)  $A \subset B \wedge B \subset \text{not}(C) \rightarrow A \subset \text{not}(C)$ . Тому  $A \cap C = \{\} \rightarrow \text{True}$ ;



4.

$$A \setminus (B \cap C) = (A \setminus B) \cup (A \setminus C) \rightarrow \text{False}$$

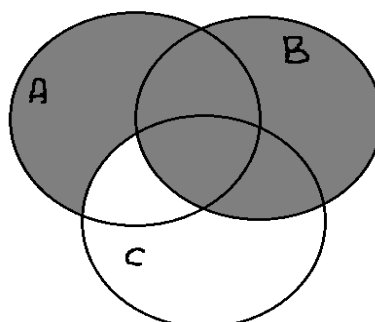
$$f1 = A \setminus (B \cap C) = (A \setminus B) \cap (A \setminus C)$$

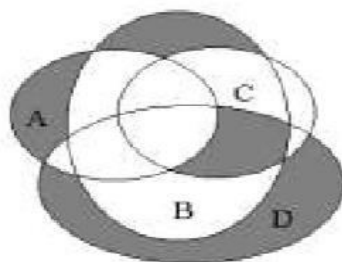
$$f2 = (A \setminus B) \cup (A \setminus C) = A \setminus (B \cup C)$$

$f1 \neq f2$  отже вирази нетотожні.

5.

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





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

$$(A \cup B) \cap \bar{C} \cup (\overline{A \cap B} \cap C) \cup (A \cap B \cap C).$$

7.

$$\begin{aligned} &= ((A \cup B) \cap \text{not}(C)) \cup (\text{not}(A \cap B) \cap C) \cup (A \cap B \cap C) = \\ &= ((A \cup B) \cap \text{not}(C)) \cup (\text{not}(A) \cup \text{not}(B)) \cap C \cup (A \cap B \cap C) = \\ &= ((A \cup B) \cup (\text{not}(A) \cup \text{not}(B))) \cap (((A \cup B) \cap C) \cap ((\text{not}(A) \cup \text{not}(B)) \cap \text{not}(C))) \cap \\ &\cap (\text{not}(C) \cup C) \cup (A \cap B \cap C) = U \cap \{ \} \cup U \cup (A \cap B \cap C) = \{ \} \cup (A \cap B \cap C) = \\ &= (A \cap B \cap C) \end{aligned}$$

8.

$|B| = 35$  — всі учні

$|A| = 20$  — англійська

$|H| = 11$  — німецька

$|HЧ| = 10$  — нічого

$$|B| - |HЧ| = |A| + |H| = 25;$$

$$(|A| + |H|) - |H| = 25 - 11 = 14 \text{ — людей вивчає тільки англійську.}$$

## Додаток 2:

Постановка задачі:

11. Ввести з клавіатури множину дійсних чисел. Реалізувати операцію доповнення до цієї множини. Вивести на екран новоутворену множину. Побудувати булеан цієї множини. Знайти програмно його потужність.

Вимоги до програми:

Програма має передбачати такі можливості:

1. Автоматичне знаходження результуючих множин, поданих списками елементів, для відповідного завдання:
  - запис характеристичних векторів заданих множин (в універсальну включити всі елементи заданих);
  - запис отриманих множин списком елементів;
  - запис потужності утворених множин;
  - запис булеану однієї з них.
2. Введення вхідних даних вручну:
  - задати елементи першої множини;
  - задати елементи другої множини.

Код програми:

```
#include "stdafx.h"
#include "conio.h"
#include <iostream>
#include <math.h>
#include <string.h>

using namespace std;

const int n = 5;
const long double hu = trunc(0.1 * 10) / 10;

struct set {

    void inp_set(double *a) //////////
    {
        char ck[5];
        cout << "Enter set a with different values" << endl;
        for (int i = 0; i < n; i++)
        {
            p1:
                cin >> ck;
                if (atof(ck) || ck[0] == '0') a[i] = atof(ck);
                else
                {
                    cout << "Enter again" << endl;
                    goto p1;
                }
                a[i] = trunc(a[i] * 10) / 10;
        }
    }

    void out_set(double *a, int n) //////////
    {
        for (int i = 0; i < n; i++)
        {
            cout << a[i] << "\t";
        }
        cout << endl;
    }
}
```

```

void out_set(bool *a, int n) ///////////////
{
    for (int i = 0; i < n; i++)
    {
        cout << a[i] << "\t";
    }
    cout << endl;
}

```

```

void sort_set(double m[n]) ///////////////
{
    for (int j = 0; j < n; j++)
    {
        for (int i = 0; i < n - 1; i++)
        {
            double temp = m[i];
            if (m[i + 1] < m[i])
            {
                m[i] = m[i + 1];
                m[i + 1] = temp;
            }
        }
    }
}

```

```

int u_range(double *a)
{
    int r = 0;
    double x = a[0];
    while (x <= a[n - 1] + hu / 2)
    {
        x += hu;
        r++;
    }
    return r;
}

```

```

void universe(double *a, double *u, int u_r)
{
    double k = a[0];
    for (int i = 0; i < u_r; i++, k += hu)
    {
        u[i] = k;
        u[i] = round(u[i] * 10) / 10;
        if (u[i] == -0) u[i] = 0;
    }
}

```

```

void mask(double *a, double *u, bool *a_mask, bool *u_mask, int u_r)
{
    int k = 0;
    for (int i = 0; i < u_r; i++)
    {
        u_mask[i] = 1;
        if (a[k] == u[i])
        {
            a_mask[i] = 1;
            k++;
        }
        else a_mask[i] = 0;
    }
}

```

```

void complement(bool *a_mask, bool *u_mask, bool *a_mask_comp, int u_r)
{
    for (int i = 0; i < u_r; i++)
    {
        a_mask_comp[i] = u_mask[i] - a_mask[i];
    }
}

```

```

void boolean(double *a, int n) // Принцип: будується ніби табл істинності для всіх варіацій значень
// тоді ми робимо проходи по табличці і
якщо є 1 то виводимо ці значення якщо 0 то ніякі не виводимо
// & - побітове порівняння а не порівняння
значень напри 3(11) & 1(01) = true
// f - побітове представлення індексів
таблиці істинності яка і є булеаном
// i - це бітова маска кожного рядка в

```

```

{
    int nos = 1 << n; // Кількість підмножин
    int f = 0;
    for (int i = 0; i < nos; i++) //Перебір бітових масок(значення числа по таблиці істиності)
    {
        cout << ("{" );
        for (int j = 0; j < n; j++) // Перебір бітів(значення в таблиці істиності)
        {
            f = 1 << j; // Бітова маска(побітове представлення) індексів, а індекси в
свою чергу представляють елементи масиву
            if (i & f) // Якщо в значення i (по таблиці істиності) є таке двійкове f що
вони мають спільні елементи(якщо двійкове f входить до i)( то виводимо ті спільні елементи)
                cout << a[j] << " "; // То виводимо jй елемент множини
            }
            cout << "}\n";
        }
    }
};

int main()
{
    set s;

    double a[n] = { 0 };

    s.inp_set(a);
    cout << endl;
    cout << "Set a:" << endl;
    s.out_set(a, n);
    cout << endl;
    cout << "Set a sorted:" << endl;
    s.sort_set(a);
    s.out_set(a, n);
    cout << endl;

    int u_r = s.u_range(a);
    double *u = new double[u_r];
    bool *a_mask = new bool[u_r];
    bool *u_mask = new bool[u_r];
    bool *a_mask_compl = new bool[u_r];
    cout << "Set universe:" << endl; // Від найменшого a до найбільшого a - межі універсуму
    s.universe(a, u, u_r);
    s.out_set(u, u_r);
    cout << endl;

    cout << "Power of universe is: " << u_r << endl;
    cout << endl;

    cout << "Mask a:" << endl;
    s.mask(a, u, a_mask, u_mask, u_r);
    s.out_set(a_mask, u_r);
    cout << endl;

    cout << "Mask a complementation:" << endl;
    s.complement(a_mask, u_mask, a_mask_compl, u_r);
    s.out_set(a_mask_compl, u_r);
    cout << endl;

    int pa = 0;
    int pac = 0;
    for (int i = 0; i < u_r; i++)
    {
        if (a_mask[i] == 1) pa++;
        if (a_mask_compl[i] == 1) pac++;
    }
    cout << "Power of a = " << pa << " Power of a complemented = " << pac << endl;
    cout << endl;

    cout << "A complementation:" << endl;
    for (int i = 0; i < u_r; i++)
    {
        if (a_mask_compl[i] == 1) cout << u[i] << "\t";
    }
    cout << endl;
    cout << endl;
    ///////////////////////////////////
    cout << "Boolean of a:" << endl;
    s.boolean(a, n);
    cout << endl;

```



```

delete[] a_mask_compl;
delete[] a_mask;
delete[] u_mask;
delete[] u;

_getch();
return 0;
}

```

Результат роботи програми:

```

C:\Windows\system32\cmd.exe
Enter set a with different values
-1 0 -2.5 0.7 1.5

Set a:
-1      0      -2.5    0.7    1.5

Set a sorted:
-2.5    -1      0      0.7    1.5

Set universe:
-2.5    -2.4    -2.3    -2.2    -2.1    -2      -1.9    -1.8    -1.7    -1.6
-1.5    -1.4    -1.3    -1.2    -1.1    -1      -0.9    -0.8    -0.7    -0.6
-0.5    -0.4    -0.3    -0.2    -0.1    0      0.1    0.2    0.3    0.4
0.5     0.6     0.7     0.8     0.9     1      1.1    1.2    1.3    1.4
1.5

Power of universe is: 41

Mask a:
1      0      0      0      0      0      0      0      0      0
0      0      0      0      0      1      0      0      0      0
0      0      0      0      0      1      0      0      0      0
0      0      1      0      0      0      0      0      0      0
1

Mask a complementation:
0      1      1      1      1      1      1      1      1      1
1      1      1      1      1      0      1      1      1      1
1      1      1      1      1      0      1      1      1      1
1      1      0      1      1      1      1      1      1      1
0

Power of a = 5   Power of a complemented = 36

A complementation:
-2.4    -2.3    -2.2    -2.1    -2      -1.9    -1.8    -1.7    -1.6    -1.5
-1.4    -1.3    -1.2    -1.1    -0.9    -0.8    -0.7    -0.6    -0.5    -0.4
-0.3    -0.2    -0.1    0.1     0.2     0.3     0.4     0.5     0.6     0.8
0.9     1      1.1    1.2    1.3    1.4

Boolean of a:
< >
< -2.5 >
< -1 >
< -2.5 -1 >
< 0 >
< -2.5 0 >
< -1 0 >
< -2.5 -1 0 >
< 0.7 >
< -2.5 0.7 >
< -1 0.7 >
< -2.5 -1 0.7 >
< 0 0.7 >
< -2.5 0 0.7 >
< -1 0 0.7 >
< -2.5 -1 0 0.7 >
< 1.5 >
< -2.5 1.5 >
< -1 1.5 >
< -2.5 -1 1.5 >
< 0 1.5 >
< -2.5 0 1.5 >
< -1 0 1.5 >
< -2.5 -1 0 1.5 >
< 0.7 1.5 >
< -2.5 0.7 1.5 >
< -1 0.7 1.5 >
< -2.5 -1 0.7 1.5 >
< 0 0.7 1.5 >
< -2.5 0 0.7 1.5 >
< -1 0 0.7 1.5 >
< -2.5 -1 0 0.7 1.5 >

```

**Висновок:** Я ознайомився на практиці із основними теорії множин, навчився будувати діаграми Ейлера-Венна операцій над множинами, використовувати закони алгебри множин, освоїв принцип включень-виключень для двох і трьох множин та комп'ютерне подання множин.