МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ<<ЛЬВІВСЬКА ПОЛІТЕХНІКА>>

Інститут ІКНІ

Кафедра систем штучного інтелекту

ЗВІТ

Лабораторна робота **№2**
З курсу “Дискретна математика”

Виконав:

Гавриляк Тарас

гр. **КН-110**

Прийняв(ла):

ст. вк. **Мельникова Н.І.**

Львів – 2018

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.

2.1. Основні поняття теорії множин. Операції над множинами

Множина – це сукупність об'єктів, які називають елементами.
Кажуть, що множина *A* є **підмножиною** множини *S* (цей факт позначають $A \subseteq S$, де $\subseteq$ – знак нестрогого включення), якщо кожен її елемент автоматично є елементом множини *S*. Досить часто при цьому кажуть, що множина *A* міститься в множині *S*.
Якщо $A \subseteq S$ і $S \neq A$, то *A* називають **власною (строгою, істинною) підмножиною** *S* (позначають $A \subset S$, де $\subset$ – знак строгого включення).
Дві множини *A* та *S* називаються **рівними**, якщо вони складаються з однакових елементів. У цьому випадку пишуть $A=S$.
Якщо розглядувані множини є підмножинами деякої множини, то її називають **універсумом** або **універсальною множиною** і позначають літерою *U* (зауважимо, що універсальна множина існує не у всіх випадках).
Множини як об'єкти можуть бути елементами інших множин, Множину, елементами якої є множини, інколи називають **сімейством**.

Множину, елементами якої є всі підмножини множини *A* і тільки вони (включно з порожньою множиною та самою множиною *A*), називають **булеаном** або **множиною-степенем** множини *A* і позначають *P(A).*

Потужністю скінченної множини *A* називають число її елементів, позначають $|A|$.
Множина, яка не має жодного елемента, називається *порожньою* і позначається $\varnothing$.
Вважається, що порожня множина є підмножиною будь-якої множини, а також $A \subset A$.
Множину, елементами якої є всі підмножини множини *A* і тільки вони (включно з порожньою множиною та самою множиною *A*), називають **булеаном** або **множиною-степенем** множини *A* і позначають *P(A).*

Множина всіх підмножин множини *A* називається *булеаном* і позначається $P(A)$. Потужність скінченної множини дорівнює кількості її елементів, позначається A . Потужність порожньої множини дорівнює 0.
Якщо $A = n$, то $_n P(A) = 2$.

Приклад. $\{1, 4, 5\} \subset \{-1, 0,1, 2, 3, 4, 5, 7\}$, але

$$\{1, 4, 5\} \notin \{-1, 0,1, 2, 3, 4, 5, 7\}$$

Приклад. Знайти булеан множини $A = \{a, ,b\ c\}$.

Розв’язання.

Потужності множин $A = 3$, $P(A) = 8$. Булеан має вигляд

$P(A) = \{\varnothing,\{a\},\{b\},\{c\},\{a,b\},\{a\ c,\ \},\{b\ c,\ \},\{a,\ b,\ c\}\}$.

Дві множини A і B *рівні* між собою, якщо $A \subset B$ і $B \subset A$.

Над множинами можна виконувати дії: об’єднання, переріз, доповнення, різницю, симетричну різницю, декартів добуток.

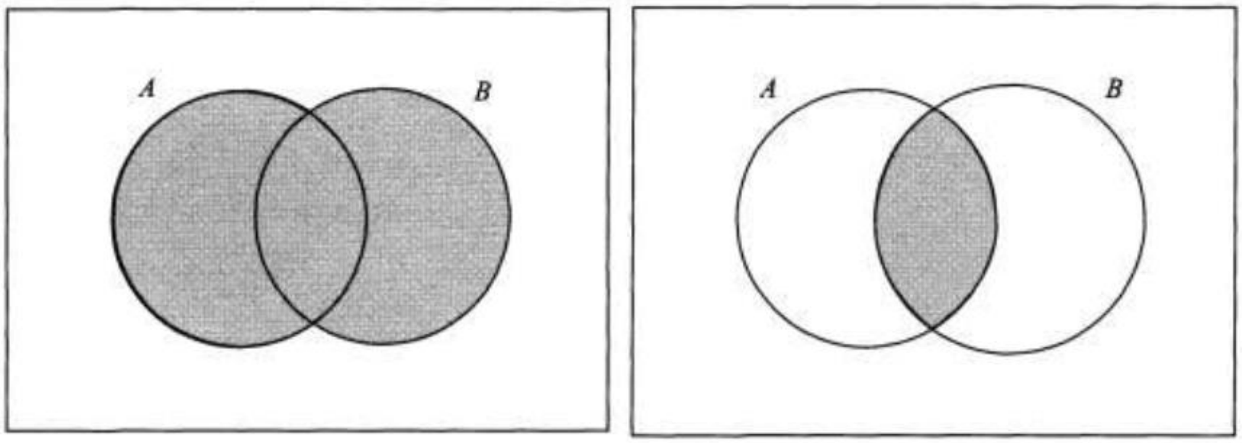
Об’єднанням двох множин *A* і *B* (рис. 2.1, *а*) називають множину

$A \cup B = \{x : (x \in A) \vee (x \in B)\}$.

Перетином (перерізом) двох множин *A* і *B* (рис. 2.1, *б*) називають множину

$A \cap B = \{x : (x \in A) \wedge (x \in B)\}$.

а) *б)*



Різницею множин *A* та *B* (рис. 2.2, *а*) називають множину

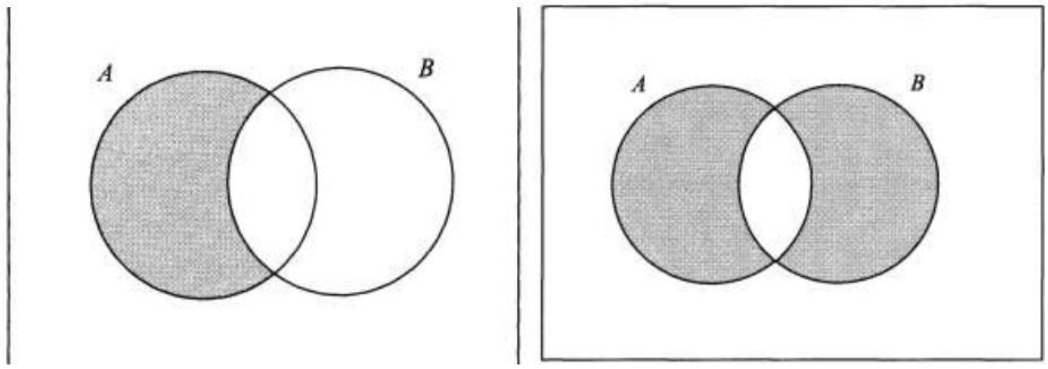
$A \setminus B = \{x : (x \in A) \wedge (x \notin B)\}$.

3

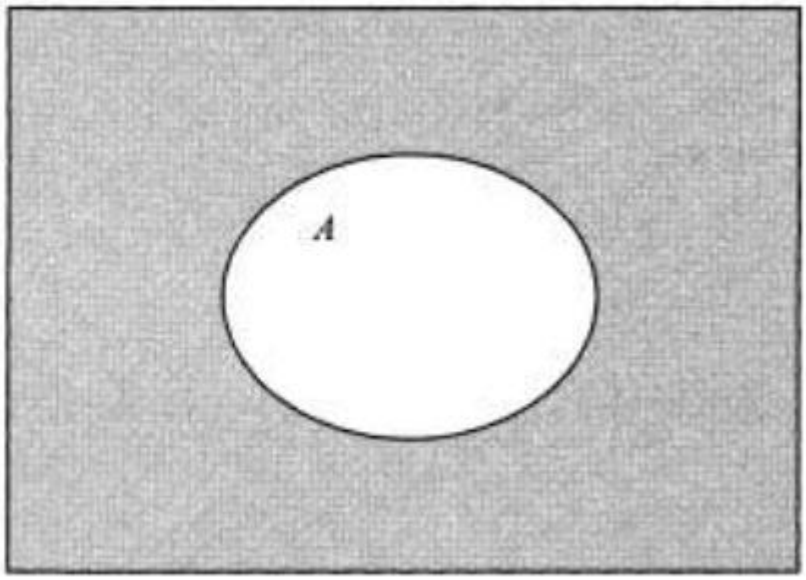
Зазначимо, що $A \setminus B = A \cap B$.

Симетричною різницею множин *A* та *B* (рис. 2.2, *а*) називають множину

$A \Delta B = \{x : ((x \in A) \wedge (x \notin B)) \vee ((x \in B) \wedge (x \notin A))\}$.



В означенні різниці не розглядають випадок $B \subset A$. Якщо $B \subset A$, то різницю $A \setminus B$ називають **доповненням множини *B* до множини *A*** і позначають $_A B$. Для підмножини *A* універсальної множини *U* можна розглядати доповнення A до U, тобто $U \setminus A$, її позначають $A=\{x:\neg(x\in A)\}\Leftrightarrow A=\{x: x\notin A)\}$ і називають **доповненням множини** A.



Пріоритет виконання операцій у спадному порядку – доповнення, переріз, об’єднання, різниця, симетрична різниця.

2.2. Закони алгебри множин

Закони асоціативності	
$A\cup(B\cup C)=(A\cup B)\cup C$	$A\cap(B\cap C)=(A\cap B)\cap C$
Закони комутативності	
$A\cup B=B\cup A$	$A\cap B=B\cap A$
Закони тотожності	
$A\cup\varnothing=A$	$A\cap U=A$
Закони домінування	
$A\cup U=U$	$A\cap\varnothing=\varnothing$
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	
$A\cup\overline{A}=U$	$A\cap\overline{A}=\varnothing$
$\overline{U}=\varnothing$	$\overline{\varnothing}=U$
$\overline{\overline{A}}=A$	$\overline{\overline{A}}=A$
Закони де Моргана	
$\overline{(A\cup B)}=\overline{A}\cap\overline{B}$	$\overline{(A\cap B)}=\overline{A}\cup\overline{B}$

Вивчення законів алгебри множин дозволяє зауважити, що кожна з тотожностей правої колонки може бути одержана з відповідної тотожності лівої шляхом заміни $\cup$ на $\cap$, $\varnothing$ на U і навпаки. Таку відповідність тотожностей називають *законом двоїстості*, а відповідні тотожності – *двоїстими* одна одній. Використовуючи цей закон, можна обгрунтувати двоїсту тотожність, довівши пряму і обернувши операції.

2.3. Формули включень та виключень для двох і трьох множин. Комп'ютерне подання множин

$|A\cup B|=|A|+|B|-|A\cap B|$ **,**

$A\cup B\cup C = A + B + C - \left(A\cap B + B\cap C + A\cap C \right) + A\cap B\cap C.$

Одним із найпоширеніших та найпростіших способів є подання множин за допомогою *бітових рядків*. Нехай універсальна множина U

містить *n* елементів. Упорядкуємо довільним способом елементи універсальної множини. Тоді $\{\ \}$ $_{1\,2\,3\,1}$, , ,... , . $_{n\,n}U\ a\ a\ a\ a\ a\ _{-}=$

Множину $A \subset U$ зображають у комп'ютері рядком із 0 та 1 довжини n так: якщо $a\ A\ _i \in$, то *i*-й біт дорівнює 1, якщо $a\ A\ _i \notin$, то *i*-й біт дорівнює 0. Такий рядок бітів називають *характеристичним вектором* підмножини *A*.

Завдання до практичних занять та лабораторної роботи з теми №2:

1. Отримати індивідуальний варіант завдань.
2. Розв'язати індивідуальне завдання згідно варіанту з Додатку № 1.
3. Написати на будь-якій відомій студентові мові програмування програму для реалізації вказаних операцій над заданими множинами відповідно до варіанту з Додатку №2.
4. Оформити звіт про виконану роботу.
5. Продемонструвати викладачеві результати, відповісти на запитання стосовно виконання роботи. Лабораторна робота вважається зарахованою, якщо програма протестована разом з викладачем та отримано вірний результат під час аудиторних занять.

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.

3. Некоректне введення даних (символьних чи числових).
4. Виведення відповідного повідомлення у випадку неіснування розв’язку.

Варіант №6

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

а)

$A \cap C = \{1, 2, 3\}$

$(A \cap C) \cup B = \{1, 2, 3, 5, 6, 7, 8, 9, 10\}$

б)

$B \Delta C = \{1, 2, 3, 5, 6, 7\}$

Записали елементи множини B, котрих немає у C, а потім елементи C, котрих немає у B.

2. На множинах задачі 1 побудувати булеан множини
$C \setminus \left(A \cup C\right) \cap B$. Знайти його потужність.

(2) (1) (3) – порядок виконання дій;

C\(!A∪!C) ∩ B

1) {4, 5, 6, 7, 8, 9, 10 }
2) {1, 2, 3}
3) {порожня множина}
4) Булеан = {порожня множина}

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) $\varnothing \cup \{\varnothing\} = \varnothing$; б) $N \in Z$;
в) $Q \cup N = R \cap Q$; г) $R \setminus (N \cup Z) \subset Q$;
д) якщо $A \cap B \subset\ !C$, то $!(A \cap B) \subset C$.

Твердження $\varnothing \cup \{\varnothing\} = \varnothing$ - невірне. Тому, що при об'єднанні з порожньою множиною отримаємо $\{\varnothing\}$.

Твердження $N \in Z$ – вірне. Тому, що множина Z містить всі елементи множини N.

Твердження $Q \cup N = R \cap Q$ – є вірним. Тому, що $Q \cup N$ = R і $R \cap Q$ = R.

Твердження $R \setminus (N \cup Z) \subset Q$ – є вірним. Тому, що порожня множина є підмножиною Q.

Якщо $A \cap B \subset !C$, то $!(A \cap B) \subset C$.

Твердження є не вірним. З першого виразу випливає що будь-які спільні елементи А та В не є підмножиною С, що заперечує другий вираз.

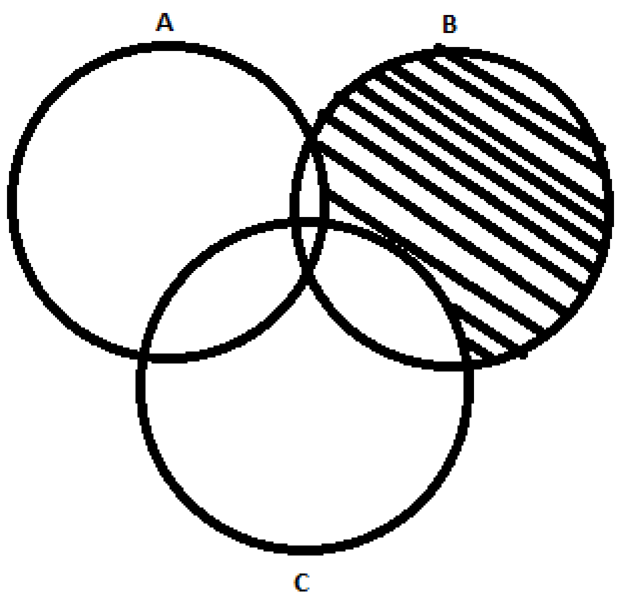
4. Логічним методом довести тотожність:
$A \cap (B \setminus C) = (A \cap B) \setminus C$.

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

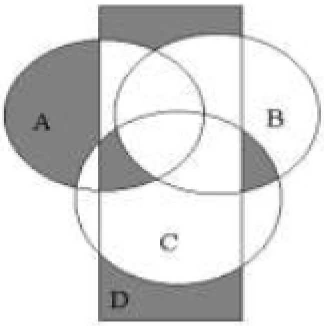
$(A \cap B) \setminus C = (A \cap B) \cap !(C)$

Тотожність доведено.

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



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



(A ∩ !D) ∪ (D ∩ !A ∩ ! B ∩ !C) ∪ (A ∩ D ∩ C ∩ !B) ∪ (B ∩ C ∩ !D

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

(A∆B∩C) ∪ B = (((A\B) ∪ (B\A)) ∩ C) ∪ B = (((A∩ !(B) ∪ (B∩ !(A))) ∩ C) ∪ B=
= ((A∩ !(A)) ∪ (B∩ !(B)) ∩ C) ∪ B = (пуста множина ∩ C) ∪ B = пуста множина ∪ B = B

8. Скільки чисел серед 1, 2, 3,…, 99, 100 таких, що не діляться на жодне з чисел 11, 17?

Нехай A – множина чисел, що діляться на 11,

B – множина чисел, що ділиться на 17, то

|A| = 100/11 = 9,09

|B| = 100/17 = 5,88

|A∩B| = 100/17 * 11 = 0,53

|A ∪ B| = 9,09 + 5,88 – 0,53 = 14,44

100 – 14,44 = 85,56

Відповідь: На множині 1-100 існує 86 таких чисел, що не діляться ні на 17 ні на 11.

Код програми:

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

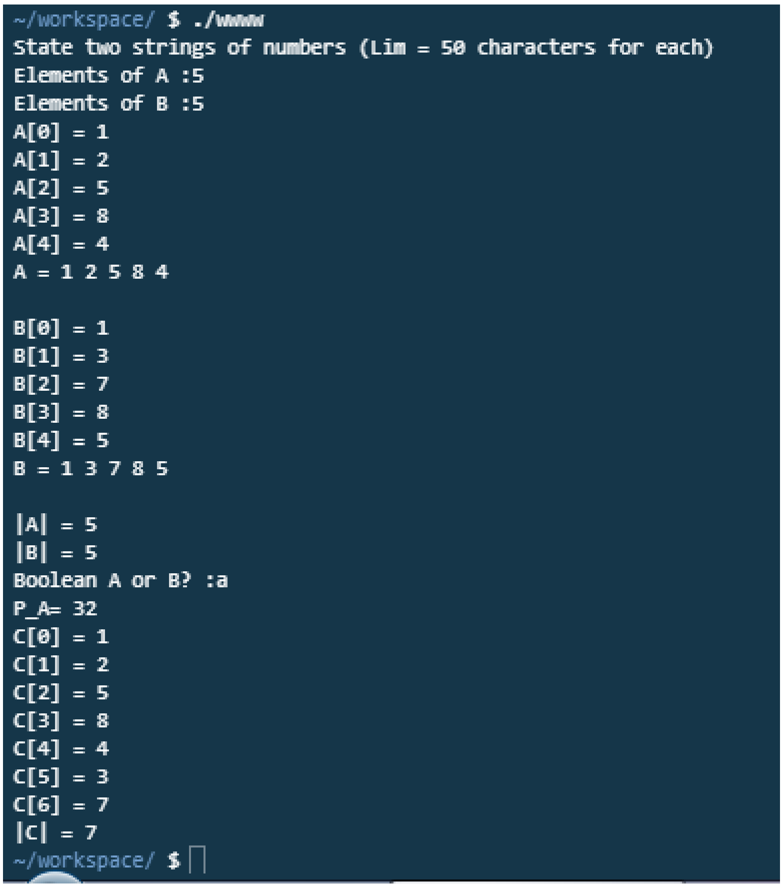
```
int main() {

   int A [50], B [50], C[100], length_1, length_2, length_3 = 0;
   printf("State two strings of numbers (Lim = 50 characters for each)\n");
   printf("Elements of A :");
   scanf("%d",&length_1);
   printf("Elements of B :");
   scanf("%d",&length_2);

   for ( int i = 0; i < length_1; i++) { // заповнення множини A
      printf("A[%d] = ", i);
      scanf("%d", &A[i]);       //
   }

   printf("A = ");

   for (int i = 0; i < length_1; i++ ) { // вивід множини A
      printf("%d ", A[i]);
   }

   puts("\n");

   for (int j = 0; j < length_2; j++) { // заповнення множини B
      printf("B[%d] = ", j);
      scanf("%d", &B[j]);
   }
```

```
printf("B = ");

for( int j = 0; j < length_2; j++ ) { // вивід множини B
   printf("%d ", B[j]);
}

puts("\n");
printf("|A| = %d\n",length_1);      // вивід потужностей
printf("|B| = %d\n",length_2);



int P_A = pow(2, length_1);
int P_B = pow(2, length_2);

printf("Boolean A or B? :");       // запит на множину + вивід булена 1-ї з множин
getchar();
char c = getchar();

while(1) {

   getchar();

   if ( (c == 'a') || ( c == 'A') ) {
      printf("P_A= %d\n", P_A);
      break;
   }
   else if (  (c == 'b') || ( c == 'B') ) {
      printf("P_B= %d\n" , P_B);
```

```
        break;
    }
    else {
        printf("Character is wrong, try again (a or b)! :  ");
        c = getchar();
    }
}




for ( int i = 0, j = 0; i < length_1; i++ ) {

    for ( j = 0; j < length_3; j++ ) {

        if (A[i] == C[j] ) {
            break;
        }

    }
        if(j == length_3 ){
            C[length_3] = A[i];
            length_3++;
        }
}

for (int i = 0, j = 0; i < length_2; i++ ) {

    for ( j = 0; j < length_3; j++ ){
```

```
            if(B[i] == C[j] ) {
                break;
            }
        }
        if( j == length_3 ){
            C[length_3] = B[i];
            length_3++;
        }
    }

    for (int i = 0; i < length_3; i++ ) {

        printf("C[%d] = %d \n",i, C[i]);
    }

    printf("|C| = \n%d\n",length_3);

return 0;
}
```

Результат програми:



```
~/workspace/ $ ./wwww
State two strings of numbers (Lim = 50 characters for each)
Elements of A :5
Elements of B :5
A[0] = 1
A[1] = 2
A[2] = 5
A[3] = 8
A[4] = 4
A = 1 2 5 8 4

B[0] = 1
B[1] = 3
B[2] = 7
B[3] = 8
B[4] = 5
B = 1 3 7 8 5

|A| = 5
|B| = 5
Boolean A or B? :a
P_A= 32
C[0] = 1
C[1] = 2
C[2] = 5
C[3] = 8
C[4] = 4
C[5] = 3
C[6] = 7
|C| = 7
~/workspace/ $
```

Висновок:

Навчився на практиці основним поняттям
теорії множин, навчився будувати діаграми Ейлера-Венна , використовувати закони алгебри множин, освоїв принцип включень-виключень для двох і трьох множин та комп’ютерне подання множин.