

МІНІСТЕРСТВО ОСВІТИ І НАУКИ НАЦІОНАЛЬНИЙ
УНІВЕРСИТЕТ «ЛЬВІВСЬКА ПОЛІТЕХНІКА»

Інститут комп'ютерних наук та
інформаційних технологій Кафедра систем
штучного інтелекту

**Лабораторна робота №3
з курсу “Дискретна математика”**

Виконав: ст. гр. КН-113
Іванюшенко Нестор

Викладач: Мельникова Н.І.

[варіант 2]

Побудова матриці бінарного відношення

Мета роботи: набуття практичних вмінь та навичок при побудові матриць бінарних відношень та визначені їх типів.

Теоретичні відомості

Декартів добуток множин A і B (позначається $A \times B$) – це множина всіх упорядкованих пар елементів (a, b) , де $a \in A$, $b \in B$.

Бінарним відношенням R називається підмножина декартового добутку $A \times B$ (тобто $R \subset A \times B$). Якщо пара (a, b) належить відношенню R , то пишуть $(a, b) \in R$, або aRb .

Завдання 1. +

Чи є вірною рівність $(A \cap B) \times (C \cap D) = (A \times C) \cap (B \times D)$?

Нехай $(x, y) \in (A \cap B) \times (C \cap D) \Rightarrow (x \in A \ \& \ x \in B) \ \& \ (y \in C \ \& \ y \in D) \Rightarrow (x \in A \ \& \ y \in C) \ \& \ (x \in B \ \& \ y \in D) \Rightarrow (x, y) \in (A \times C) \ \& \ (x, y) \in (B \times D) \Rightarrow (x, y) \in (A \times C) \cap (B \times D)$

Завдання 2.

Знайти матрицю відношення $R \subset 2^A \times 2^B$

$R = \{(x, y) \mid x \in A \ \& \ y \in B \ \& \ |y| = |x|, x \cap y = \emptyset\}$

де $A = \{1, 2\}$, $B = \{1, 3, 5\}$

	$\{\emptyset\}$	$\{1\}$	$\{3\}$	$\{5\}$	$\{1, 3\}$	$\{1, 5\}$	$\{3, 5\}$	$\{1, 3, 5\}$
$\{\emptyset\}$	1	1	1	1	0	0	0	0
$\{1\}$	1	0	1	1	0	0	1	0
$\{2\}$	1	1	1	1	0	0	0	0
$\{1, 2\}$	0	0	0	0	0	0	1	0

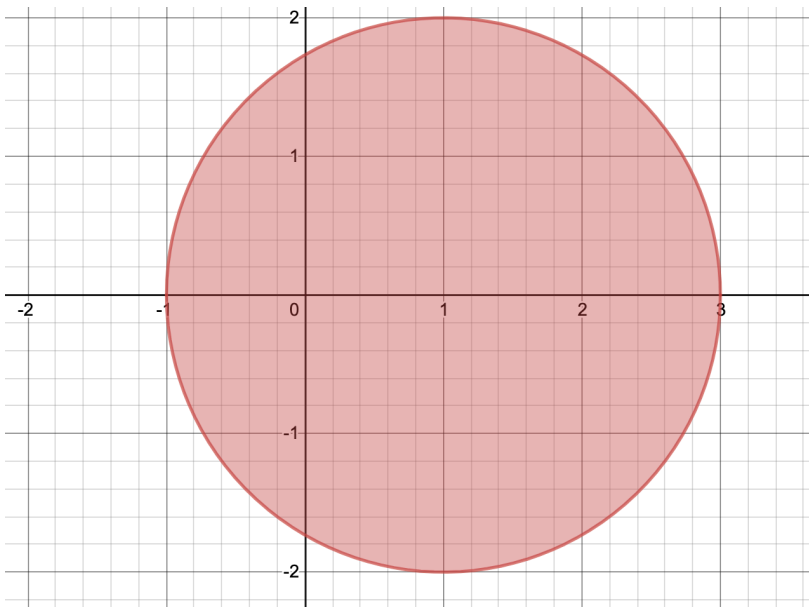
Завдання 3.

Зобразити відношення графічно: $a = \{(x, y) \mid (x, y) \in R^2 \ \& \ x^2 - 2x + y^2 \leq 3\}$, де R - множина дійсних чисел.

розглянемо нерівність $x^2 - 2x + y^2 = 3$:

$$(x^2 - 2x + 1) + y^2 \leq 4;$$

$(x - 1)^2 + y^2 \leq 2^2$ - рівняння круга з центром у $(1, 0)$ з радіусом 2:



Завдання 4.

Маємо бінарне відношення $R \subset A \times A$, де $A = \{a, b, c, d, e\}$, яке задане своєю матрицею: $A(R) =$

1	1	0	0	0
0	0	1	1	0
1	0	0	0	1
0	0	0	0	0
0	0	0	1	0

Перевірити чи є дане відношення є рефлексивним, симетричним, транзитивним, антисиметричним

дане відношення не є ні рефлексивним, ні антирефлексивним, ні симетричним, ні антисиметричним, ні транзитивним.

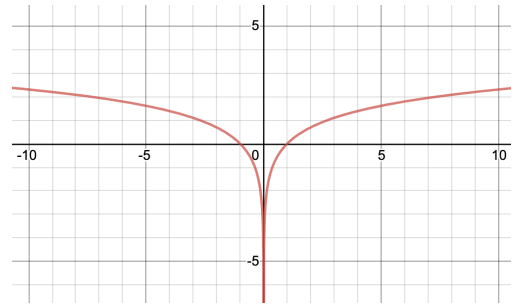
Завдання 5.

Визначити множину (якщо це можливо), на якій дане відношення є:

а) функціональним; б) бієктивним:

$$a = \{(x, y) \mid (x, y) \in R^2 \ \& \ y = \ln |x| \}$$

$$x \in R \setminus \{0\}; \quad y \in R;$$



на графіку можемо спостерігати
функціональне відношення на множині

$$x \in R \setminus \{0\}$$

і бієктивне на множинах

$$x \in (0, \infty) \text{ і } x \in (-\infty, 0)$$

додаток 2

Написати програму, яка знаходить матрицю бінарного відношення $\rho \subset A \times B$, заданого на двох числових множинах. Реалізувати введення цих множин, та виведення на екран матриці відношення. Перевірити програмно якого типу є задане відношення. Навести різні варіанти тестових прикладів.

$$\rho = \{(a, b) \mid a \in A \ \& \ b \in B \ \& \ a < b\}$$

```
1  #include <iostream>
2  #include <iomanip>
3
4  #define GET_VARIABLE_NAME(Variable) (#Variable)
5
6  void printproperty(bool boo, const char *booname) {
7      if (boo) {
8          std::cout << booname << "\n";
9      }
10 }
11 #define printproperty(var) printproperty(var, GET_VARIABLE_NAME(var))
12
13 void deduplicate(int *set, int *length);
```

ВВІД МНОЖИН

```
15 ▶ int main() {
16     //temporary arrays for input
17     int *seta = new(int);
18     int *setb = new(int);
19     int alen = 0, blen = 0;
20     //input
21     int input = 0;
22     ///getting elements of A
23     std::cout << "Enter the elements of set A split up with space: \n";
24     do {
25         std::cin >> input;
26         seta[alen] = input;
27         alen++;
28         seta = (int *) realloc(seta, size: sizeof(int) * (alen + 1));
29     } while (std::cin.get() != '\n');
30     ///getting elements of b
31     std::cout << "Enter the elements of set B split up with space: \n";
32     do {
33         std::cin >> input;
34         setb[blen] = input;
35         blen++;
36         setb = (int *) realloc(setb, size: sizeof(int) * (blen + 1));
37     } while (std::cin.get() != '\n');
```

ВИДАЛЯЄМ ОДНАКОВІ ЕЛЕМЕНТИ

```
40     //processing input
41     delduplicate(seta, &alen);
42     delduplicate(setb, &blen);
43     const int rows = alen;
44     const int columns = blen;
45     //for random sets
46     int MAX = 20;
47     srand(time(0));
48     //
49     int a[rows];
50     int b[columns];
51     memcpy(a, seta, n: sizeof(int) * rows);
52     memcpy(b, setb, n: sizeof(int) * columns);
53     ///randomise
54     // for(int i = 0; i<rows; i++){
55     //     a[i] = rand()%MAX;
56     // }
57     // for(int i = 0; i<columns; i++){
58     //     b[i] = rand()%MAX;
59     // }
```

заповнюємо і виводимо матрицю відношення

```
60  ///relation set
61      int R[rows][columns];
62
63      for (int i : a) {
64          for (int j : b) {
65              std::cout << std::setw( n: 3) << i << " " << j << " |";
66          }
67          std::cout << "\n";
68      }
69      std::cout << "\n\n";
70  ///relation matrix
71      for (int i = 0; i < rows; i++) {
72          for (int j = 0; j < columns; j++) {
73              R[i][j] = a[i] < b[j] ? 1 : 0;
74              std::cout << std::setw( n: 3) << R[i][j] << " | ";
75          }
76          std::cout << "\n\n";
77      }
78  }
```

перевіряємо на властивості:

```
78
79      bool symmetric = true;
80      bool antisymmetric = true;
81      bool reflexive = true;
82      bool antireflexive = true;
83      bool transitive = true;
84      for (int i = 0; i < rows; i++) {
85          for (int j = 0; j < columns; j++) {
86              if (R[i][j] != R[j][i]) {
87                  symmetric = false;
88              }
89              if (R[i][j] == R[j][i] && i != j) {
90                  antisymmetric = false;
91              }
92              if (R[i][j] == 0 && i == j) {
93                  reflexive = false;
94              }
95              if (R[i][j] == 1 && i == j) {
96                  antireflexive = false;
97              }
98          }
99      }
100  }
```

транзитивність:

```
101 //checking transitivity
102 for (int i = 0; i < rows; i++) {
103     for (int j = 0; j < rows; j++) {
104         for (int k = 0; k < columns; k++) {
105             if (R[i][j] && R[j][k] && !R[i][k]) {
106                 transitive = false;
107             }
108         }
109     }
110 }
111 }
```

друкуємо властивості:

```
111 }
112 //if relation set has this property – print it
113 printproperty(symmetric);
114 printproperty(antisymmetric);
115 printproperty(reflexive);
116 printproperty(antireflexive);
117 printproperty(transitive);
118
119 return 0;
120 }
```

функція для видалення однакових елементів:

```
122 void deduplicate(int *set, int *length) {
123     bool smth_deleted = false;
124
125     for (int i = 0; i < *length; i++) {
126         for (int j = 0; j < *length; j++) {
127             if (i != j && set[i] == set[j]) {
128                 smth_deleted = true;
129                 for (int k = (i >= j ? i : j); k < *length; k++) {
130                     set[k] = set[k + 1];
131                 }
132                 *length -= 1;
133                 set = (int *) realloc(set, sizeof(int) * (*length + 1));
134             }
135         }
136     }
137     if (smth_deleted) { deduplicate(set, length); }
138 }
```

виконання і приклади:

```
Enter the elements of set A split up with space:
1 1 2 3
Enter the elements of set B split up with space:
2 3 4

1 2 | 1 3 | 1 4 |
2 2 | 2 3 | 2 4 |
3 2 | 3 3 | 3 4 |

1 | 1 | 1 |
0 | 1 | 1 |
0 | 0 | 1 |

antisymmetric
reflexive
transitive

Process finished with exit code 0
```

```
Enter the elements of set A split up with space:
6 4 2
Enter the elements of set B split up with space:
1 3 5

6 1 | 6 3 | 6 5 |
4 1 | 4 3 | 4 5 |
2 1 | 2 3 | 2 5 |

0 | 0 | 0 |
0 | 0 | 1 |
0 | 1 | 1 |

symmetric

Process finished with exit code 0
```

Висновок: я набув практичних вмінь та навичок при побудові матриць бінарних відношень та визначені їх типів.