

МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное образовательное учреждение  
высшего образования

**«САРАТОВСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ  
ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ИМЕНИ Н.Г. ЧЕРНЫШЕВСКОГО»**

Кафедра теоретических основ  
компьютерной безопасности и  
криптографии

**Отношение эквивалентности и отношение порядка**

ОТЧЁТ

ПО ДИСЦИПЛИНЕ

**«ПРИКЛАДНАЯ УНИВЕРСАЛЬНАЯ АЛГЕБРА»**

студента 3 курса 331 группы

специальности 10.05.01 Компьютерная безопасность

факультета компьютерных наук и информационных технологий

Никитина Арсения Владимировича

Преподаватель

профессор, д.ф.-м.н.

\_\_\_\_\_  
подпись, дата

В. А. Молчанов

Саратов 2022

## СОДЕРЖАНИЕ

ВВЕДЕНИЕ .....	3
1 <b>Цель работы и порядок ее выполнения</b> .....	4
2 Теоретические сведения .....	5
2.1 Эквивалентное замыкание бинарного отношения .....	5
2.1.1 Определение эквивалентного замыкания отношения .....	5
2.1.2 Алгоритм построения эквивалентного замыкания бинарного отношения .....	5
2.2 Фактор-множество отношения .....	6
2.2.1 Определение среза отношения через элемент .....	6
2.2.2 Определение фактор-множества отношения .....	6
2.2.3 Алгоритм построения фактор-множества бинарного отношения .....	6
3 Программная реализация рассмотренных алгоритмов .....	7
3.1 Результаты тестирования программы .....	7
3.2 Код программы, реализующей рассмотренные алгоритмы .....	7
ЗАКЛЮЧЕНИЕ .....	12

## **ВВЕДЕНИЕ**

Бинарные отношения могут быть эквивалентными, и, поэтому на них могут строиться фактор-множества. Если же бинарное отношение не является эквивалентностью, то по определенному алгоритму можно построить эквивалентное замыкание данного отношения. Также отношения могут обладать определенным порядком, в зависимости от конкретных свойств. Если же отношение обладает порядком, то для данного отношения можно построить диаграмму Хассе, а также для него могут быть найдены минимальные и максимальные, и наименьшие и наибольшие элементы. Также для бинарных отношений определены понятия контекста и концепта, а также существует алгоритм вычисления решетки концептов.

## **1 Цель работы и порядок ее выполнения**

**Цель работы** — изучение основных свойств бинарных отношений и операций замыкания бинарных отношений.

Порядок выполнения работы:

1. Разобрать определения отношения эквивалентности, фактор-множества. Разработать алгоритмы построения эквивалентного замыкания бинарного отношения и системы представителей фактор-множества.
2. Разобрать определения отношения порядка и диаграммы Хассе. Разработать алгоритмы вычисления минимальных (максимальных) и наименьших (наибольших) элементов и построения диаграммы Хассе.
3. Разобрать определения контекста и концепта. Разработать алгоритм вычисления решетки концептов.

## 2 Теоретические сведения

### 2.1 Эквивалентное замыкание бинарного отношения

#### 2.1.1 Определение эквивалентного замыкания отношения

**Замыканием отношения  $R$  относительно свойства  $P$**  называется такое множество  $R^*$ , что:

1.  $R \subset R^*$ .
2.  $R^*$  Обладает свойством  $P$ .
3.  $R^*$  является подмножеством любого другого отношения, содержащего  $R$  и обладающего свойством  $P$ .

То есть  $R^*$  является минимальным надмножеством множества  $R$ , выдерживается  $P$ .

Итак, исходя из вышесказанного, можно сделать вывод, что существуют 4 вида замыканий отношений: **транзитивное, симметричное, рефлексивное и эквивалентное.**

На множестве  $P(A^2)$  всех бинарных отношений между элементами множества  $A$  следующие отображения являются операторами замыканий:

1.  $f_r(\rho) = \rho \cup \Delta_A$  – наименьшее рефлексивное бинарное отношение, содержащее отношение  $\rho \subset A^2$ .
2.  $f_s(\rho) = \rho \cup \rho^{-1}$  – наименьшее симметричное бинарное отношение, содержащее отношение  $\rho \subset A^2$ .
3.  $f_t(\rho) = \bigcup_{n=1}^{\infty} \rho^n$  – наименьшее транзитивное бинарное отношение, содержащее отношение  $\rho \subset A^2$ .
4.  $f_{eq}(\rho) = f_t f_s f_r(\rho)$  – наименьшее отношение эквивалентности, содержащее отношение  $\rho \subset A^2$ .

#### 2.1.2 Алгоритм построения эквивалентного замыкания бинарного отношения

*Вход.* Матрица  $M(\rho)$  бинарного отношения  $\rho$  размерности  $N \times N$ .

*Выход.* Эквивалентное замыкание бинарного отношения.

1. Создать пустой список для хранения пар замыкания.
  - а) Цикл по  $i$  от 1 до  $N$ .
    1. Если  $M_{ii} = 0$ , пару  $(i, i)$  добавить в замыкание.
  - б) Цикл по  $i$  от 1 до  $N$ , цикл по  $j$  от 1 до  $N$ .
    1. Если  $M_{ij} = 1$  и  $M_{ji} = 0$ , добавить пару  $(j, i)$  в замыкание.

с) Цикл по  $e$  от 1 до  $N$ , цикл по  $k$  от 1 до  $N$ , цикл по  $i$  от 1 до  $N$ , цикл по  $j$  от 1 до  $N$ .

1. Если  $M_{ki} = M_{i,j} = 1$  и  $M_{kj} = 0$ , то добавить пару  $(k, j)$  в замыкание транзитивности и замыкание эквивалентности.

2. Ответ — эквивалентное замыкание бинарного отношения  $\rho$ .

Трудоемкость алгоритма  $O(N + N^2 + N^4) = O(N^4)$

## 2.2 Фактор-множество отношения

### 2.2.1 Определение среза отношения через элемент

Для любого подмножества  $X \subset A$  множество  $\rho(X) = \{b \in B : (x, b) \in \rho \text{ для некоторого } x \in X\}$  называется *образом* множества  $X$  относительно отношения  $\rho$ .

Образ одноэлементного множества  $X = \{a\}$  относительно отношения  $\rho$  обозначается символом  $\rho(a)$  и называется также образом элемента  $a$  или *срезом* отношения  $\rho$  через элемент  $a$ .

### 2.2.2 Определение фактор-множества отношения

Эквивалентное бинарное отношение на множестве  $A$  также принято обозначать как  $\varepsilon$ .

Срезы  $\varepsilon(a)$  называются *классами эквивалентности* по отношению  $\varepsilon$  и сокращенно обозначаются символом  $[a]$ .

Множество всех таких классов эквивалентности  $\{[a] : a \in A\}$  называется *фактор-множеством* множества  $A$  по эквивалентности  $\varepsilon$  и обозначается  $A/\varepsilon$ .

### 2.2.3 Алгоритм построения фактор-множества бинарного отношения

*Вход.* Матрица  $M(\rho)$  эквивалентного бинарного отношения  $\rho$  размерности  $N \times N$ .

*Выход.* Фактор-множество отношения.

1. Создать  $N$  пустых списков.

а) Цикл по  $i$  от 1 до  $N$ , цикл по  $j$  от 1 до  $N$ .

1. Если  $M_{ij} = 1$  добавить  $j$  в список с номером  $i$ .

2. Ответ — фактор-множество отношения.

Трудоемкость алгоритма  $O(N^2)$

### 3 Программная реализация рассмотренных алгоритмов

#### 3.1 Результаты тестирования программы

```
Матрица эквивалентного замыкания бинарного отношения:
1 1 1 1 0
1 1 1 1 0
1 1 1 1 0
1 1 1 1 0
0 0 0 0 1

Фактор-множество бинарного отношения:
[{1, 2, 3, 4}, {1, 2, 3, 4}, {1, 2, 3, 4}, {1, 2, 3, 4}, {5}]
```

Рисунок 1

#### 3.2 Код программы, реализующей рассмотренные алгоритмы

```
1 def make_set(matrix, size):
2
3     set_view = []
4
5     for i in range(size):
6         for j in range(size):
7             if matrix[i][j] == 1:
8                 set_view.append((i + 1, j + 1))
9     return sorted(set_view)
10
11
12 def matrix_set_view(matrix_set, flag=None):
13     if not flag:
14         print('Исходное отношение: {', end='')
15         print(*matrix_set, sep=', ', end='} \n ')
16     else:
17         print('{', end='')
18         print(*matrix_set, sep=', ', end='; ')
19
20
21 def factor_set(matrix, size):
22     return [{j + 1 for j, value in enumerate(matrix[i]) if value == 1} for i
23             ↪ in range(size)]
```

```

24
25 def make_equivalent_closure(copy, size):
26     list_for_equivalent_closure = set()
27     for i in range(size):
28         for j in range(size):
29             if matrix[i][j] == 1 and matrix[j][i] == 0:
30                 copy[j][i] = 1
31                 list_for_equivalent_closure.add((j + 1, i + 1))
32         if matrix[i][i] == 0:
33             copy[i][i] = 1
34             list_for_equivalent_closure.add((i + 1, i + 1))
35
36     for _ in range(size):
37         for k in range(size):
38             for i in range(size):
39                 for j in range(size):
40                     if copy[k][i] == copy[i][j] == 1 and copy[k][j] == 0:
41                         copy[k][j] = 1
42                         list_for_equivalent_closure.add((k + 1, j + 1))
43
44     return sorted(list_for_equivalent_closure), copy
45
46
47 def is_transitive(matrix, size):
48
49     for k in range(size):
50         for i in range(size):
51             for j in range(size):
52                 if matrix[k][i] == matrix[i][j] == 1 and matrix[k][j] == 0:
53                     return False
54     return True
55
56
57 def is_symmetric_or_antisymmetric(matrix, size):
58
59     flag_symmetric = True
60     flag_antisymmetric = True
61
62     for i in range(size):
63         for j in range(size):
64             if not matrix[i][j] == matrix[j][i]:

```



```

65         flag_symmetric = False
66         elif matrix[i][j] == matrix[j][i] and not i == j:
67             flag_antisymmetric = False
68         elif not flag_symmetric and not flag_antisymmetric:
69             return False, False
70
71     return flag_symmetric, flag_antisymmetric
72
73
74 def is_reflexive_or_anti_reflexive(matrix, size):
75
76     flag_reflexive = True
77     flag_anti_reflexive = True
78
79     for i in range(size):
80         if matrix[i][i] == 0:
81             flag_reflexive = False
82         elif matrix[i][i] == 1:
83             flag_anti_reflexive = False
84         if not flag_reflexive and not flag_anti_reflexive:
85             return False, False
86
87     return flag_reflexive, flag_anti_reflexive
88
89
90 def get_data():
91     n = int(input())
92     m = [[int(elem) for elem in input().split()] for _ in range(n)]
93     m_set = [(i + 1, j + 1) for i in range(n) for j in range(n) if m[i][j] ==
94               ↪ 1]
95
96     return m, sorted(m_set), n
97
98 matrix, matrix_set, size = get_data()
99
100 copy = matrix
101
102 ls, mt = make_equivalent_closure(copy, size)
103
104 print('Матрица эквивалентного замыкания бинарного отношения:')
105 for i in range(len(mt)):
106     print(*mt[i])

```

```

105 print('\n ')
106 print('Фактор-множество бинарного отношения:')
107 print(factor_set(mt, size))
108 '''
109 Примеры входных данных:
110
111 4
112 0 1 1 0
113 1 1 1 0
114 0 1 1 0
115 0 0 0 1
116
117 4
118 0 1 0 0
119 0 0 0 0
120 0 0 0 1
121 0 1 0 0
122
123 3
124 0 1 0
125 0 0 1
126 1 0 0
127
128 5
129 1 0 1 1 0
130 0 1 0 1 0
131 1 0 1 1 0
132 1 1 1 1 0
133 0 0 0 0 1
134
135 4
136 1 1 0 1
137 0 1 1 0
138 0 0 1 1
139 0 0 0 1
140
141 4
142 0 0 1 0
143 1 0 0 1
144 0 0 0 0
145 0 1 0 0

```



## **ЗАКЛЮЧЕНИЕ**

В ходе лабораторной работы были рассмотрены понятия эквивалентного замыкания бинарного отношения и получения представителей фактормножества. Также были получены алгоритмы вычисления минимальных и максимальных, и наименьших и наибольших элементов бинарного отношения, а также был определен и программно реализован алгоритм построения диаграммы Хассе. Был описан алгоритм построения решетки концептов. Для всех алгоритмов произведена асимптотическая оценка.