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. ___________ В. А. Молчанов
подпись, дата

Саратов 2022

СОДЕРЖАНИЕ

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. $f_r(\rho) = \rho \cup \triangle_A$ – 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. $f_t(\rho) = \cup_{n=1}^{\infty}\rho^n$ – 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 ρ размерности $N \times N$.

Выход. Эквивалентное замыкание бинарного отношения.

1. Создать пустой список для хранения пар замыкания.
   a) Цикл по i от 1 до N.
      1. Если $M_{ii} = 0$, пару (i, i) добавить в замыкание.
   b) 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.

c) 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 $M_{ki} = 0$, то добавить пару (k, k) в замыкание транзитивности и замыкание эквивалентности.

2. Ответ — эквивалентное замыкание бинарного отношения ρ.

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

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

```python
def make_set(matrix, size):

    set_view = []

    for i in range(size):
        for j in range(size):
            if matrix[i][j] == 1:
                set_view.append((i + 1, j + 1))
    return sorted(set_view)


def matrix_set_view(matrix_set, flag=None):
    if not flag:
        print('Исходное отношение: {', end='')
        print(*matrix_set, sep=', ', end='}\n')
    else:
        print('{', end='')
        print(*matrix_set, sep=', ', end='; ')


def factor_set(matrix, size):
    return [{j + 1 for j, value in enumerate(matrix[i]) if value == 1} for i in range(size)]

```

```
def make_equivalent_closure(copy, size):
    list_for_equivalent_closure = set()
    for i in range(size):
        for j in range(size):
            if matrix[i][j] == 1 and matrix[j][i] == 0:
                copy[j][i] = 1
                list_for_equivalent_closure.add((j + 1, i + 1))
        if matrix[i][i] == 0:
            copy[i][i] = 1
            list_for_equivalent_closure.add((i + 1, i + 1))

    for _ in range(size):
        for k in range(size):
            for i in range(size):
                for j in range(size):
                    if copy[k][i] == copy[i][j] == 1 and copy[k][j] == 0:
                        copy[k][j] = 1
                        list_for_equivalent_closure.add((k + 1, j + 1))

    return sorted(list_for_equivalent_closure), copy


def is_transitive(matrix, size):

    for k in range(size):
        for i in range(size):
            for j in range(size):
                if matrix[k][i] == matrix[i][j] == 1 and matrix[k][j] == 0:
                    return False
    return True


def is_symmetric_or_antisymmetric(matrix, size):

    flag_symmetric = True
    flag_antisymmetric = True

    for i in range(size):
        for j in range(size):
            if not matrix[i][j] == matrix[j][i]:
```

```
                flag_symmetric = False
            elif matrix[i][j] == matrix[j][i] and not i == j:
                flag_antisymmetric = False
            elif not flag_symmetric and not flag_antisymmetric:
                return False, False

    return flag_symmetric, flag_antisymmetric


def is_reflexive_or_anti_reflexive(matrix, size):

    flag_reflexive = True
    flag_anti_reflexive = True

    for i in range(size):
        if matrix[i][i] == 0:
            flag_reflexive = False
        elif matrix[i][i] == 1:
            flag_anti_reflexive = False
        if not flag_reflexive and not flag_anti_reflexive:
            return False, False

    return flag_reflexive, flag_anti_reflexive


def get_data():
    n = int(input())
    m = [[int(elem) for elem in input().split()] for _ in range(n)]
    m_set = [(i + 1, j + 1) for i in range(n) for j in range(n) if m[i][j] == 1]
    return m, sorted(m_set), n


matrix, matrix_set, size = get_data()

copy = matrix
ls, mt = make_equivalent_closure(copy, size)

print('Матрица эквивалентного замыкания бинарного отношения:')
for i in range(len(mt)):
    print(*mt[i])
```

```
print('\n')
print('Фактор-множество бинарного отношения:')
print(factor_set(mt, size))
'''
Примеры входных данных:

4
0 1 1 0
1 1 1 0
0 1 1 0
0 0 0 1

4
0 1 0 0
0 0 0 0
0 0 0 1
0 1 0 0

3
0 1 0
0 0 1
1 0 0

5
1 0 1 1 0
0 1 0 1 0
1 0 1 1 0
1 1 1 1 0
0 0 0 0 1

4
1 1 0 1
0 1 1 0
0 0 1 1
0 0 0 1

4
0 0 1 0
1 0 0 1
0 0 0 0
0 1 0 0
```

```
'''
```

ЗАКЛЮЧЕНИЕ

В ходе лабораторной работы были рассмотрены понятия эквивалентного замыкания бинарного отношения и получения представителей фактор-множества. 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.