МИНОБРНАУКИ РОССИИ
ФГБОУ ВО «СГУ ИМЕНИ Н. Г. ЧЕРНЫШЕВСКОГО»

УНИВЕРСАЛЬНЫЕ АЛГЕБРЫ И АЛГЕБРА ОТНОШЕНИЙ

ЛАБОРАТОРНАЯ РАБОТА

студента 3 курса 331 группы
направления 10.05.01 — Компьютерная безопасность
факультета КНиИТ
Токарева Никиты Сергеевича

Проверил
аспирант _____ В. Н. Кутин

Саратов 2022

1 Постановка задачи

Цель работы – изучение основных понятий универсальной алгебры и операций над бинарными отношениями.

Порядок выполнения работы:

1. Рассмотреть понятие алгебраической операции и классификацию свойств операций. Разработать алгоритмы проверки свойств операций: ассоциативность, коммутативность, идемпотентность, обратимость, дистрибутивность.
2. Рассмотреть основные операции над бинарными отношениями. Разработать алгоритмы выполнения операции над бинарными отношениями.
3. Рассмотреть основные операции над матрицами. Разработать алгоритмы выполнения операций над матрицами.

2 Теоретические сведения по рассмотренным темам с их обоснованием

2.1 Понятие алгебраической операции

Отображение $f : A^n \to A$ называется алгебраической n-арной операцией или просто **алгебраической операцией** на множестве A. При этом n называется порядком или арностью алгебраической операции f.

Далее для бинарной операции f по возможности будем использовать мультипликативную запись с помощью символа «·», т.е.вместо $f(x, y)$ писать $x \cdot y$. При необходимости для бинарной операции f используется также аддитивная запись с помощью символа «+», т.е. вместо $f(x, y)$ записывается $x + y$.

2.2 Классификация свойств операций

Бинарная операция · на множестве A называется:

— идемпотентной, если $\forall x \in A$ выполняется равенство $x \cdot x = x$;

— коммутативной, если $\forall x, y \in A$ выполняется равенство $x \cdot y = y \cdot x$;

— ассоциативной, если $\forall x, y, z \in A$ выполняется равенство $x \cdot (y \cdot z) = (x \cdot y) \cdot z$;

— обратимой, если $\forall x, y \in A$, если уравнения $x \cdot a = y$ и $b \cdot x = y$ имеют решение, причем единственное;

— дистрибутивной относительно операции +, если $\forall x, y, z \in A$ выполняются равенства

$$x \cdot (y + z) = (x \cdot y) + (x \cdot z),$$
$$(y + z) \cdot x = (y \cdot x) + (z \cdot x);$$

2.3 Основные операции над бинарными отношениями

— Над бинарными отношениями можно выполнять любые теоретико-множественные операции, в частности операции объединения $\cup$ и пересечения $\cap$;

— Обратным для бинарного отношения $\rho \subset A \times B$ называется бинарное отношение $\rho^{-1} \subset B \times A$, определяющееся по формуле:

$$\rho^{-1} = \{(b, a) : (a, b) \in \rho\};$$

— Композицией бинарных отношений $\rho \subset A \times B$ и $\sigma \subset B \times C$ называется бинарное отношение $\rho \circ \sigma \subset A \times C$, определяющееся по формуле:

$$\rho \circ \sigma = \{(a, c) : (a, b) \in \rho \text{ и } (b, c) \in \sigma \text{ для некоторого } b \in B\};$$

2.4 Основные операции над матрицами

— Сложение и вычитание матриц.

Суммой $A + B$ матриц $A_{m\times n} = (a_{ij})$ и $B_{m\times n} = (b_{ij})$ называется матрица $C_{m\times n} = (c_{ij})$, где $c_{ij} = a_{ij} + b_{ij}$ для всех $i = \overline{1, m}$ и $j = \overline{1, n}$.

Разностью $A - B$ матриц $A_{m\times n} = (a_{ij})$ и $B_{m\times n} = (b_{ij})$ называется матрица $C_{m\times n} = (c_{ij})$, где $c_{ij} = a_{ij} - b_{ij}$ для всех $i = \overline{1, m}$ и $j = \overline{1, n}$.

— Умножение матрицы на число.

Произведением матрицы $A_{m\times n} = (a_{ij})$ на число α называется матрица $C_{m\times n} = (c_{ij})$, где $c_{ij} = \alpha a_{ij}$ для всех $i = \overline{1, m}$ и $j = \overline{1, n}$.

— Произведение двух матриц.

Произведением матриц $A_{m\times n} = (a_{ij})$ на матрицу $B_{m\times n} = (b_{ij})$ называется матрица $C_{m\times n} = (c_{ij})$, где $c_{ij} = \sum\limits_{k=1}^{n} a_{ik}b_{kj}$ для всех $i = \overline{1, m}$ и $j = \overline{1, n}$.

— Транспонирование матрицы.

Транспонированной по отношению к матрице $A_{m\times n} = (a_{ij})$ называется матрица $A^T_{n\times m} = (a^T_{ij})$ для элементов которой $a^T_{ij} = a_{ji}$.

— Обращение матрицы.

Обращение матрицы $A_{m\times n}$ - получение матрицы A^{-1}, обратной к исходной матрице A. Обратная матрица A^{-1} — такая, при умножении которой на исходную матрицу A получается единичная матрица E. Это такая матрица, которая удовлетворяет равенству

$$AA^{-1} = A^{-1}A = E$$

3 Результаты работы

3.1 Коды программ, реализующей рассмотренные алгоритмы

```
import numpy as np

# Построение матрицы по бинарному отношению
def go_to_matrix(br, n):
    n = -1
    for pair in br:
        tmp = max(pair)
        n = max(tmp, n)
    n += 1
    a = []
    for i in range(n):
        a.append([0 for j in range(n)])
    for pair in br:
        a[pair[0]][pair[1]] = 1
    return np.array(a)


# Построение бинарного отношения по матрице
def go_to_set(a, n):
    s = []
    for i in range(n):
        for j in range(n):
            if a[i][j] == 1:
                s.append((i, j))
    return s


# Построение объединение бинарных отношений
def get_union_bin_rel(br1, br2=[]):
  if br2 == []:
    union = br1.copy()
    for pair in br1:
      if pair not in union:
        union.append(pair)
    print('Answer:')
    print_binary_relation(union, len(union))
  else:
    union = br1.copy()
    for pair in br2:
```

```
      if pair not in union:
        union.append(pair)
    print('Answer:')
    print_binary_relation(union, len(union))


# Построение пересечения бинарных отношений
def get_intersection_bin_rel(br1, br2=[]):
  if br2 == []:
    intersec = []
    for x1, y1 in br1:
      for x2, y2 in br1:
        if x1 == x2 and y1 == y2:
          intersec.append((x1, y1))
    print('Answer:')
    print_binary_relation(intersec, len(intersec))
  else:
    intersec = []
    for x1, y1 in br1:
      for x2, y2 in br2:
        if x1 == x2 and y1 == y2:
          intersec.append((x1, y1))
    print('Answer:')
    print_binary_relation(intersec, len(intersec))


# Построение обратного бинарного отношения
def get_reverse_bin_rel(br):
  rev_br = []
  for x, y in br:
    rev_br.append((y, x))
  print('Answer:')
  print_binary_relation(rev_br, len(rev_br))


# Построение композиции бинарных отношений
def get_composition_bin_rel(br1, br2=[]):
  if br2 == []:
    comp_br = []
    for x1, y1 in br1:
      for x2, y2 in br1:
```

```
        if y1 == x2:
          comp_br.append((x1, y2))
    print('Answer:')
    print_binary_relation(comp_br, len(comp_br))
  else:
    comp_br = []
    for x1, y1 in br1:
      for x2, y2 in br2:
        if y1 == x2:
          comp_br.append((x1, y2))
    print('Answer:')
    print_binary_relation(comp_br, len(comp_br))


# Вывод матрицы
def print_matrix(a, n):
    cnt = 0
    print('Your matrix:')
    for i in a:
        if (cnt < n):
            print(f'{i}' + ' ')
        else:
            print(f'{i}' + '\n')
            cnt = 0
        cnt += 1


# Вывод бинарного отношения
def print_binary_relation(br, n):
    print('{ ', end='')
    for i in range(n):
        if i == n - 1:
            print('(' + str(br[i][0]) + ', ' + str(br[i][1]), end=')')
        else:
            print('(' + str(br[i][0]) + ', ' + str(br[i][1]), end='), ')
    print(' }')


#
def check_all_bin_rel_properties(br1, br2=[]):
  if br2 == []:
```

```
    print('Your binary relation:')
    print_binary_relation(br1, len(br1))
    print('Choose operation:')
    print('Press 1 to get union of binary relation')
    print('Press 2 to get intersection of binary relation')
    print('Press 3 to get reverse binary relation')
    print('Press 4 to get composition of binary relation')
    bl = input()
    if bl == '1':
      get_union_bin_rel(br1)
      check_all_bin_rel_properties(br1)
    elif bl == '2':
      get_intersection_bin_rel(br1)
      check_all_bin_rel_properties(br1)
    elif bl == '3':
      get_reverse_bin_rel(br1)
      check_all_bin_rel_properties(br1)
    elif bl == '4':
      get_composition_bin_rel(br1)
      check_all_bin_rel_properties(br1)
    else:
      choose_mode()
  else:
    print('Your binary relations:')
    print('First:', end='')
    print_binary_relation(br1, len(br1))
    print('Second', end='')
    print_binary_relation(br2, len(br2))
    print('Choose operation:')
    print('Press 1 to get union of binary relations')
    print('Press 2 to get intersection of binary relations')
    print('Press 3 to get reverse binary relations')
    print('Press 4 to get composition of binary relations')
    bl = input()
    if bl == '1':
      get_union_bin_rel(br1, br2)
      check_all_bin_rel_properties(br1, br2)
    elif bl == '2':
      get_intersection_bin_rel(br1, br2)
      check_all_bin_rel_properties(br1, br2)
    elif bl == '3':
```

```
      print('First:', end='')
      get_reverse_bin_rel(br1)
      print('Second', end='')
      get_reverse_bin_rel(br2)
      check_all_bin_rel_properties(br1, br2)
    elif bl == '4':
      get_composition_bin_rel(br1, br2)
      check_all_bin_rel_properties(br1, br2)
    else:
      print('Exit...')
#
def construction_of_binary_relation():
    print('Enter numbers of binary relation:')
    s = input()
    tmp = [i for i in s.split(' ')]
    if len(tmp) % 2 != 0:
      print('Incorrect input!')
      choose_mode()
    else:
      br1 = []
      k = 0
      for i in range(1, len(tmp)):
        if i % 2 != 0:
          br1.append((tmp[i - 1], tmp[i]))
          k += 1
      print('Do you want to enter other binary relation? (0 - no, 1 - yes):')
      bl = input()
      print('Enter numbers of binary relation:')
      if bl == '1':
        s = input()
        tmp = [i for i in s.split(' ')]
        if len(tmp) % 2 != 0:
          print('Incorrect input!')
          choose_mode()
        else:
          br2 = []
          k = 0
          for i in range(1, len(tmp)):
            if i % 2 != 0:
              br2.append((tmp[i - 1], tmp[i]))
              k += 1
```

```
            check_all_bin_rel_properties(br1, br2)
        elif bl == '0':
            check_all_bin_rel_properties(br1)

    return choose_mode()


# Главное меню
def choose_mode():
    print('Choose mode:')
    print('Press 1 to check properties')
    print('Press 2 to execute operations for binary relations')
    print('Press 3 to execute operations for matrix')
    print('Press 4 to exit')
    bl = input()
    if bl == '1':
        print('chmod1')
    elif bl == '2':
        construction_of_binary_relation()
    elif bl == '3':
        print('chmod3')
        a = []
        print('Enter the number of verticies')
        n = int(input())
        print('Enter the matrix values')
        for i in range(n):
            tmp = input().split()
            for j in range(len(tmp)):
                tmp[j] = int(tmp[j])
            a.append(tmp)
    elif bl == '4':
        return
    else:
        print('Incorrect output')
        return choose_mode()


choose_mode()
```

3.2 Результаты тестирования программ

ЗАКЛЮЧЕНИЕ

В результате лабораторной работы были рассмотрены теоретические сведения об отношении эквивалентности, разобраны определения фактор-множества, отношения порядка и диаграммы Хассе, контекста и концепта. Опираясь на изложенную выше теорию, были разработаны алгоритмы построения эквивалентного замыкания бинарного отношения и системы представителей фактор-множества, алгоритмы вычисления минимальных (максимальных) и наименьших (наибольших) элементов и построения диаграммы Хассе, а также алгоритмы построения решетки концептов. Была произведена оценка сложности каждого из построенных алгоритмов. Была реализована программа, написанная на языке Python.