МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное образовательное учреждение высшего образования

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

Кафедра математической кибернетики и компьютерных наук

# Отношение порядка и упорядоченные множества

ОТЧЕТ ПО ДИСЦИПЛИНЕ

«Комбинаторная теория полугрупп»

студента 3 курса 311 группы

специальности 02.03.02 Фундаментальная информатика и информационные технологии

факультета компьютерных наук и информационных технологий

Данилина Даниила Ивановича

Преподаватель

профессор ___________ В. А. Молчанов

подпись, дата

Саратов 2023

## СОДЕРЖАНИЕ

# 1 ПОСТАНОВКА ЗАДАЧИ

**Цель работы** - изучение основных понятий теории полугрупп.

**Порядок выполнения работы:**

1. Рассмотреть понятия полугруппы разработать алгоритм теста Лайта проверки ассоциативности бинарной операции. **Вход:** таблица Кэли бинарной операции на множестве X. **Выход:** операция ассоциативна или неассоциативна.
2. Рассмотреть понятия подполугруппы и порождающего множества. Разработать алгоритм построения подполугрупп по по таблице Кэли. **Вход:** полугруппа S с таблицей Кэли и подмножество $X \subset S$. **Выход:** подполугруппа $\langle X \rangle \subset S$.
3. Разработать алгоритм построения полугруппы бинарных отношений по заданному порождающему множеству. **Вход:** конечное множество X бинарных отношений (булевых матриц). **Выход:** полугруппа $\langle X \rangle$.

# 2 ТЕОРЕТИЧЕСКИЕ СВЕДЕНИЯ ПО РАССМАТРИВАЕМЫМ ТЕМАМ С ИХ ОБОСНОВАНИЕМ

## 2.1 Алгебраические операции

**Опр.** 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 $\cdot$, т.е. вместо f(x, y) писать

$x \cdot y$.

**Опр.** Бинарная операция $\cdot$ на множестве A называется:

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

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

3. идемпотентной, если ∀x ∈ A выполняется равенство x · x = x;

4. обратимой, если ∀x, y ∈ A уравнения x · a = y и b · x = y имеют решение, причем единственное;

5. дистрибутивной относительно операции +, если ∀x, y, z ∈ A выполняются равенства x · (y + z) = (x · y) + (x · z), (y + z) · x = (y · x) + (z · x);

## 2.2 Теория полугрупп Определение.

Полугруппа — это алгебра S = (S, ·) с одной ассоциативной бинарной операцией ·, т.е. выполняется (x·y)· z = x·(y · z) для ∀x, y, z ∈ S. Если полугрупповая операция называется умножением (или сложением), то полугруппу называют мультипликативной (или аддитивной). Определение. Подмножество X полугруппы S называется подполугруппой, если X устойчиво относительно операции умножения, т.е. ∀x, y ∈ X выполняется свойство: x · y ∈ X. В этом случае множество X с ограничением на нем операции умножения исходной полугруппы S образует полугруппу. В силу общего свойства подалгебр пересечение любого семейства $X_i$ (i ∈ I) подполугрупп полугруппы S является подполугруппой S и, значит, множество Sub(S) всех подполугрупп полугруппы S является системой замыканий. множество X. Такая полугруппа обозначается символом ⟨X⟩ и называется подполугруппой S, порождённой множеством X. При этом множество X называется также **порождающим множеством** подполугруппы ⟨X⟩. В частности, если ⟨X⟩ = S, то X называется порождающим множеством полугруппы S и говорят, что множество X порождает полугруппу S.

Для любой конечной полугруппы S найдется такой конечный алфавит A, что для некоторого отображения φ : A → S выполняется равенство ⟨φ(A)⟩ = S и, значит, S ∼= $A^{+}$/ker φ  в этом случае множество A называется множеством порождающих символов полугруппы S (относительно отображения φ : A → S ). Если при этом для слов $w_1$, $w_2$∈ A выполняется равенство φ($w_1$) = φ($w_2$), т.е. $w_1 \equiv w_2$ (ker φ) , то говорят, что на S выполняется соотношение $w_1 = w_2$ (относительно отображения φ : A → S).

Очевидно, что в общем случае множество таких соотношений $w_1 = w_2$ для всех пар ($w_1$, $w_2$) ∈ ker φ будет бесконечным и не представляется возможности эффективно описать полугруппу S в виде полугруппы классов конгруэнции ker φ . Однако в некоторых случаях можно выбрать такое сравнительно простое подмножество ρ ⊂ ker φ , которое однозначно определяет конгруэнцию ker φ как наименьшую конгруэнцию полугруппы $A^+$, содержащую отношение ρ, т.е. ker φ = $f_{con}$(ρ) = $f_{eq}$ ($f_{reg}$ (ρ)).

Так как в случае ($w_1$, $w_2$) ∈ ρ по-прежнему выполняется равенство φ($w_1$) = φ($w_2$), то будем писать $w_1 = w_2$ и называть такие выражения **определяющими соотношениями**. Из таких соотношений конгруэнция ker φ строится с помощью применения следующих процедур к словам u, v ∈ $A^+$:

1. слово v непосредственно выводится из слова u, если v получается из u заменой некоторого подслова $w_1$ на слово $w_2$, удовлетворяющее определяющему соотношению $w_1 = w_2$, т.е.(u, v) = (x$w_1$y, x$w_2$y) для некоторых x, y ∈ $A^*$ ;

2. слово v выводится из слова u, если v получается из u с помощью конечного числа применения процедуры 1.

Если все выполняющиеся на S соотношения выводятся из определяющих соотношений совокупности ρ, то конгруэнция ker φ полностью определяется отношением ρ и выражение < A : $w_1 = w_2$ : ($w_1$, $w_2$) ∈ ρ > называется **копредставлением полугруппы** S.

Обозначим символом $A^+$ множество всех непустых слов над алфавитом и символом $A^*$– множество слов $A^* = A^+ \cup \{\Lambda\}$. На этих множествах слов определена операция умножения, которая называется операцией конкатенации слов и определяется по правилу: любым словам $w_1 = a_1 \ldots a_n$ и $w_2 = b_1 \ldots b_m$ операция конкатенации ставит в соответствие слово $w_1 \cdot w_2 = a_1 \ldots a_n b_1 \ldots b_n$. В результате множество слов $A^+$ с операцией конкатенации образует полугруппу, которая называется полугруппой слов над алфавитом A, и множество слов $A^*$ с операцией конкатенации образует полугруппу с единичным элементом Λ, которая называется моноидом слов над алфавитом A.

# 3 РЕЗУЛЬТАТЫ РАБОТЫ

## 3.1 Описание алгоритма теста Лайта

**Вход:** Таблица Кэли A = $(a_{ij})$ размерности n × n операции ·, множество элементов S над операцией ·.

**Выход:** ”Ассоциативность операции выполняется!” или ”Ассоциативность операции не выполняется!”.

1. Проверим ∀x, a, z элементов множества S элементы $a_{ij}$ таблицы Кэли следующим образом: если $a_{it} = a_{pk}$ (где t = $a_{jk}$, p = $a_{ij}$ для всех 0 ≤ i, j, k ≤ n − 1, где i соответствует номеру строки таблицы Кэли для элемента x, a j и k — для a и z соответственно), то вернуть ”Ассоциативность операции выполняется!”, иначе — ”Ассоциативность операции не выполняется!”. Трудоемкость алгоритма O($n^3$).

## 3.2 Алгоритм построения построения подполугруппы по заданному порождающему множеству

Вход: Полугруппа S с таблицей Кэли A = $(a_{ij})$ размерности n × n и подмножество X ⊂ S. Выход: Подполугруппа ⟨X⟩ ⊂ S.

Шаг 1. Положим i = 0, X = $X_0$.

Шаг 2. Для $X_i$ вычислим $X_1$ = {x·y : x ∈ $X_i$∧y ∈ X} и положим $X_i+1 = X_i \cup X_1$ (выражение x · y означает $a_{xy}$ в таблице Кэли A).

Шаг 3. Вычисляем ⟨X⟩ = $\bigcup_{i=0}^{\infty} X_i$.

Трудоемкость алгоритма O($n^3$).

### 3.3 Алгоритм построения полугруппы бинарных отношений по заданному порождающему множеству

**Вход.** Конечное множество X бинарных отношений (|X| = m), заданное булевыми матрицами размерности n × n.

**Выход.** Полугруппа ⟨X⟩.

Шаг 1. Пусть каждому бинарному отношению $x_i$ ∈ X (0 ≤ i ≤ n − 1) соответствует матрица $A_i$. Зададим список L так, что $L_i = A_i$ (0 ≤ i ≤ n − 1). Полученный список L является полугруппой ⟨X⟩.

Шаг 2. Зададим новый список S так, что $S_i = L_j \odot L_k$ (∀0 ≤ i ≤ $|L|^2$ − 1) (∀0 ≤ j ≤ |L| − 1) (∀0 ≤ k ≤ |L| − 1), где |L| — количество элементов в списке L.

Шаг 3. Если (∃$S_i$ ∈ S) (∀$L_j$ ∈ L) $S_i \neq L_j$ , то добавим все элементы S в список L и переходим к шагу 2. Иначе заканчиваем алгоритм.

Трудоёмкость алгоритма — O($n^m$)

### 3.4 Коды программ, реализующей рассмотренные алгоритмы

```
import numpy as np
import math
from itertools import product

# For Light test
# ==============================Task 1 Block================================================
def check_associative(cayley_table, n, input_list):
```

```python
    check = True
    for a in range(n):
        for x in range(n):
            for z in range(n):
                if cayley_table[x, input_list.index(str(cayley_table[a, z]))]\
                        != cayley_table[input_list.index(str(cayley_table[x, a])),
                                        z]:
                    check = False
                    break
    if check:
        print("Ассоциативность операции выполняется!")
    else:
        print("Ассоциативность операции не выполняется!")

def light_test():
    print("Введите элементы множества: ")
    input_list = input().replace(",", "").split()
    n = len(input_list)
    print("Введите значения таблицы Кэли некоторой операции: ")
    cayley = [list(map(int, input().split())) for i in range(n)]
    cayley_table = np.array(cayley).reshape(n, n)

    check_associative(cayley_table, n, input_list)
# ==============================Task 2 Block================================================
def build_subsemigroup():
    print("Введите элементы множества: ")
    input_list = input().replace(",", "").split()
    n = len(input_list)

    print("Введите элементы подмножества: ")
    subset = input().replace(",", "").split()

    print("Введите значения таблицы Кэли множества, "
          "для которой будет строится подполугруппа: ")
    print(" ", *input_list)
    cayley = [list(map(str, input(f"{input_list[i]} ").split()))
              for i in range(n)]
    cayley_table = np.array(cayley).reshape(n, n)
    x_i = subset.copy()
    # print("x:", subset)
    while True:
        x_l = []
```

```
        for x in x_i:
            for y in subset:
                x_l.append(cayley_table[input_list.index(x)][input_list.index(y)])
        last_x_i = x_i.copy()
        x_i = list(set(x_i).union(set(x_l)))
        x_i.sort()
        if last_x_i == x_i:
            break

    print("Подполугруппа: ", x_i)


def find_correlation(ans):
    result = {}
    correlations = {}
    for word, matrix in ans.items():
        if not any(np.array_equal(matrix, i) for i in result.values()):
            result[word] = matrix
        else:
            for k, v in result.items():
                if np.array_equal(v, matrix):
                    correlations[word] = k

    print("Копредставление: ")
    for word, matrix in result.items():
        print(word, ":\n", matrix)

    print("Полученные соотношения: ")
    for repeated_word, word in correlations.items():
        print(repeated_word, "->", word)

# ================================Task 3 Block============================================
def build_semigroup():
    print("Введите элементы множества: ")
    input_list = input().replace(",", "").split()
    n = len(input_list)
    print("Введите количество бинарных отношений")
    bin_relation_amount = int(input())
    bin_relation_matrices = {}
    for i in range(1, bin_relation_amount + 1):
        print(f"Введите значения булевой матрицы {i}-го "
              "бинарного отношения: ")
```

```
        print(" ", *input_list)
        matrix = [list(map(int, input(f"{input_list[i]} ").split()))
                  for i in range(n)]
        matrix = np.array(matrix).reshape(n, n)
        bin_relation_matrices[str(i)] = matrix

    while True:
        new_matrices = {}
        for key_i, value_i in bin_relation_matrices.items():
            for key_j, value_j in bin_relation_matrices.items():
                new_matrices[key_i + key_j] = value_i * value_j

        already_exists = []
        for mat in new_matrices.values():
            flags = []
            for mat_old in bin_relation_matrices.values():
                flags.append(np.array_equal(mat, mat_old))
            already_exists.append(any(flags))
            if not any(flags):
                break

        if all(already_exists):
            break

        bin_relation_matrices.update(new_matrices)

    find_correlation(bin_relation_matrices)
# ==============================Main Block================================================
def main():
    print("""--- Лабораторная работа № 3 ---\nВыберите действие:
    \n1 - тест Лайта;\n2 - построение подполугруппы по заданному порождающему
    множеству;\n3 - построение полугруппы бинарных отношений по заданному
    порождающему множеству.\nВаш выбор: """)
    what_action = input()
    if what_action:
        if what_action == "1":
            light_test()
        elif what_action == "2":
            build_subsemigroup()
        elif what_action == "3":
            build_semigroup()
        elif what_action == _:
```

```python
        print("Ошибка")


if __name__ == '__main__':
    main()
```

## 3.5 Результаты тестирования программ

Тест Лайта

```
--- Лабораторная работа № 3 ---
Выберите действие:

1 - тест Лайта;
2 - построение подполугруппы по заданному порождающему
    множеству;
3 - построение полугруппы бинарных отношений по заданному
    порождающему множеству.
Ваш выбор:
1
Введите элементы множества:
0, 1, 2
Введите значения таблицы Кэли некоторой операции:
0 0 0
0 1 2
0 2 1
Ассоциативность операции выполняется!
```

Построение подполугруппы по заданному порождающему множеству

```
--- Лабораторная работа № 3 ---
Выберите действие:

1 - тест Лайта;
2 - построение подполугруппы по заданному порождающему
    множеству;
3 - построение полугруппы бинарных отношений по заданному
    порождающему множеству.
Ваш выбор:
2
Введите элементы множества:
1, 2, 3
Введите элементы подмножества:
1, 2
Введите значения таблицы Кэли множества, для которой будет строится подполугруппа:
  1 2 3
1 1 2 1
2 1 3 1
3 1 1 1
Подполугруппа:  ['1', '2', '3']
```

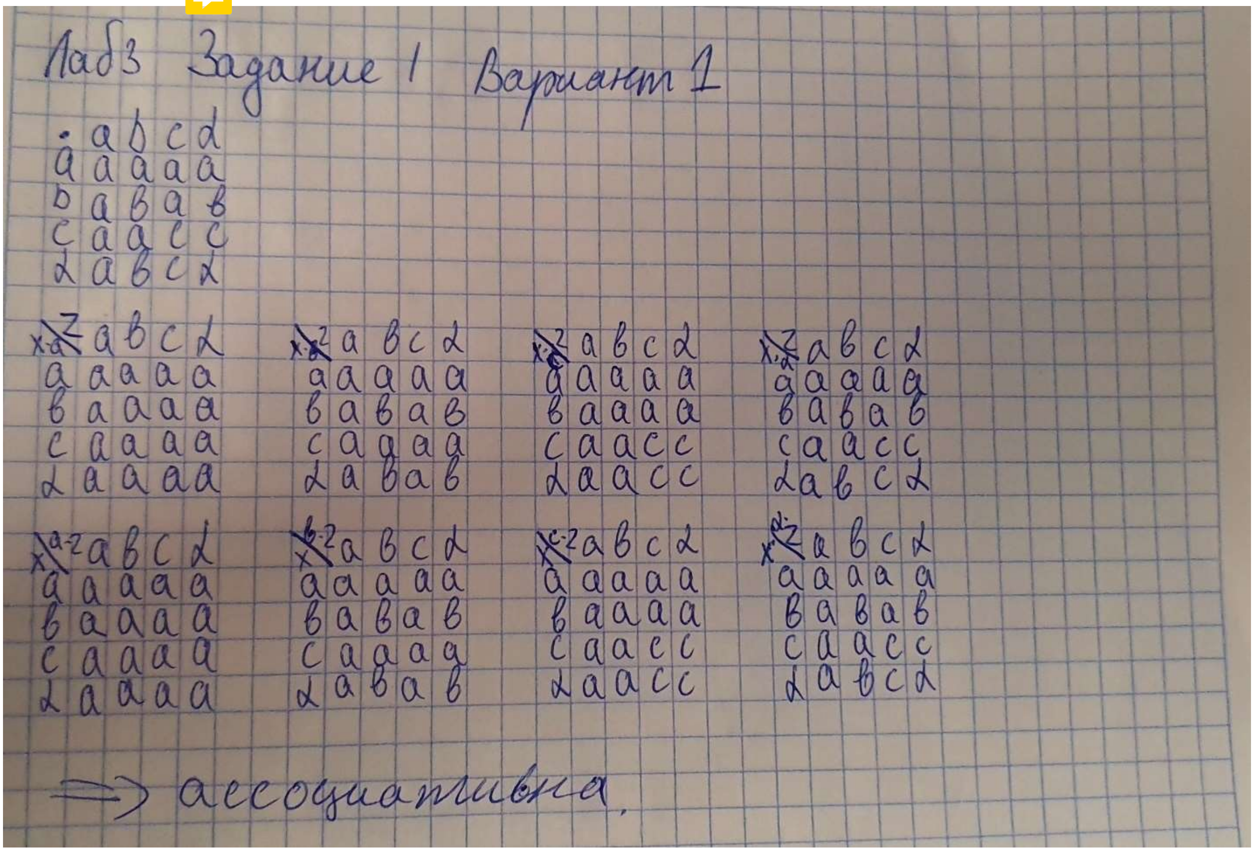
Построение полугруппы бинарных отношений по заданному порождающему множеству.

```
3 - построение полугруппы бинарных отношений по заданному
    порождающему множеству.
Ваш выбор:
3
Введите элементы множества:
a, b, c
Введите количество бинарных отношений
2
Введите значения булевой матрицы 1-го бинарного отношения:
  a b c
a 1 0 1
b 0 1 0
c 1 0 1
Введите значения булевой матрицы 2-го бинарного отношения:
  a b c
a 0 1 0
b 1 1 1
c 0 0 0
Копредставление:
1 :
 [[1 0 1]
 [0 1 0]
 [1 0 1]]
2 :
 [[0 1 0]
 [1 1 1]
 [0 0 0]]
12 :
 [[0 0 0]
 [0 1 0]
 [0 0 0]]
Полученные соотношения:
11 -> 1
21 -> 12
22 -> 2
```

## 3.6 Решение задач

Студент Данилин Д.И: порядковый номер в СДО IpsilonUni - 7. 7 mod 6 = вариант 1.

Задание 1.




Задание 2.

$T(\{1,2,3\}) \quad S = \{\underbrace{x, y}_{X_1}, \underbrace{x^2, yx}_{X_2}, \underbrace{x^3, yx^2}_{X_3}\}$

$x \cdot x = \begin{pmatrix} 1 & 2 & 3 \\ 2 & 3 & 1 \end{pmatrix} \begin{pmatrix} 1 & 2 & 3 \\ 2 & 3 & 1 \end{pmatrix} = \begin{pmatrix} 1 & 2 & 3 \\ 3 & 1 & 2 \end{pmatrix}$

$y \cdot x = \begin{pmatrix} 1 & 2 & 3 \\ 2 & 2 & 2 \end{pmatrix} \begin{pmatrix} 1 & 2 & 3 \\ 2 & 3 & 1 \end{pmatrix} = \begin{pmatrix} 1 & 2 & 3 \\ 3 & 3 & 3 \end{pmatrix}$

$y \cdot y = y$

$x \cdot y = y$

$x^2 \cdot x = \begin{pmatrix} 1 & 2 & 3 \\ 3 & 1 & 2 \end{pmatrix} \begin{pmatrix} 1 & 2 & 3 \\ 2 & 3 & 1 \end{pmatrix} = \begin{pmatrix} 1 & 2 & 3 \\ 1 & 2 & 3 \end{pmatrix}$

$yx \cdot x = \begin{pmatrix} 1 & 2 & 3 \\ 3 & 3 & 3 \end{pmatrix} \begin{pmatrix} 1 & 2 & 3 \\ 2 & 3 & 1 \end{pmatrix} = \begin{pmatrix} 1 & 2 & 3 \\ 1 & 1 & 1 \end{pmatrix}$

$yx \cdot y = y$

$x^2 \cdot y = y$

$x^3 \cdot x = x$

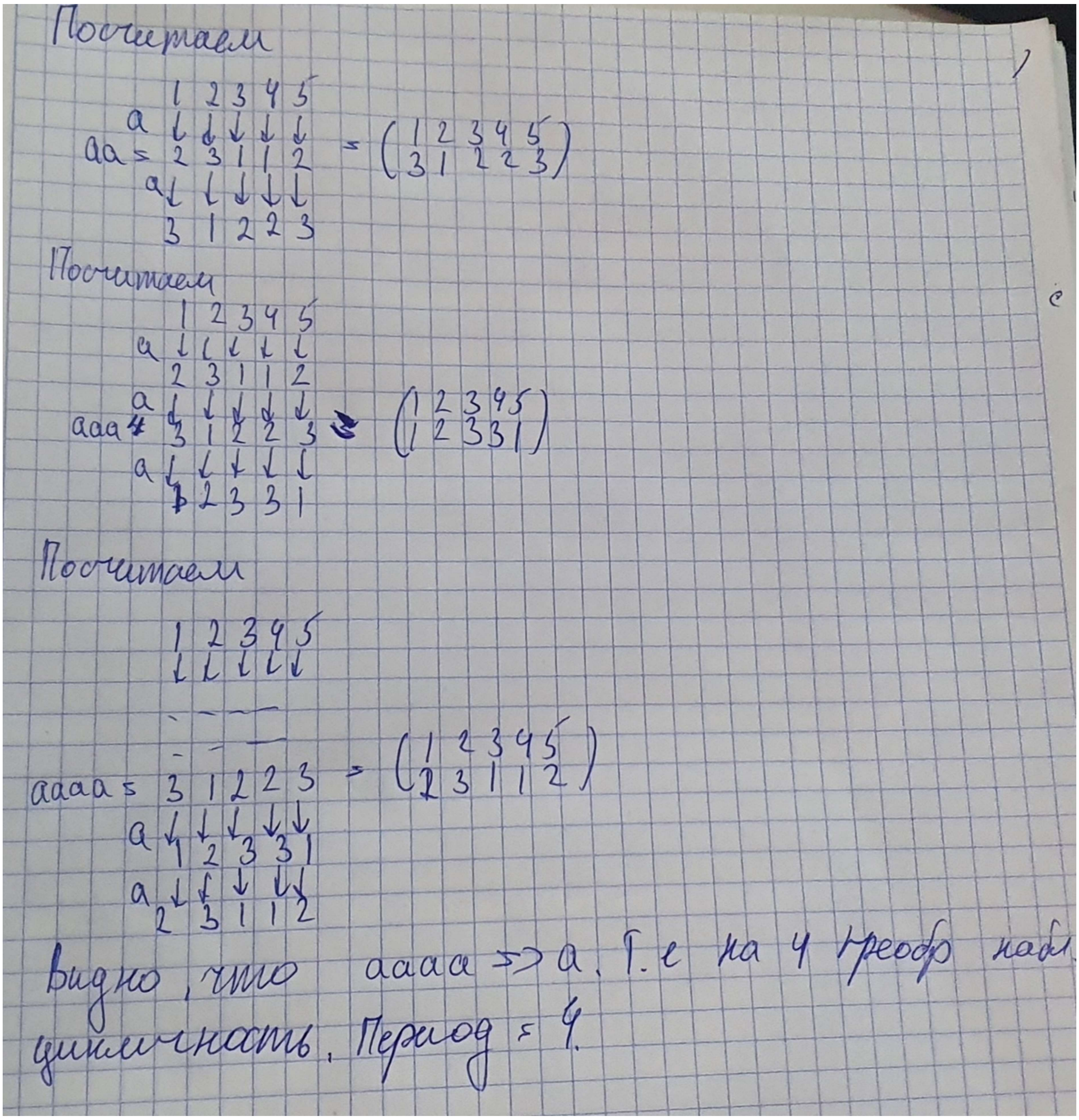
$yx^2 \cdot x = \begin{pmatrix} 1 & 2 & 3 \\ 1 & 1 & 1 \end{pmatrix} \begin{pmatrix} 1 & 2 & 3 \\ 2 & 3 & 1 \end{pmatrix} = \begin{pmatrix} 1 & 2 & 3 \\ 2 & 3 & 1 \end{pmatrix} = x$

$yx^2 \cdot y = y$

$x^3 \cdot y = y$

| $\cdot$ | $x$ | $y$ | $x^2$ | $yx$ | $x^3$ | $yx^2$ |
|---|---|---|---|---|---|---|
| $x$ | $x^2$ | $y$ | $x^3$ | $yx$ | $x$ | $yx^2$ |
| $y$ | $yx$ | $y$ | $yx^2$ | $yx$ | $y$ | $yx^2$ |
| $x^2$ | $x^3$ | $y$ | $x$ | $yx$ | $x^2$ | $yx^2$ |
| $yx$ | $yx^2$ | $y$ | $y$ | $yx$ | $yx$ | $yx^2$ |
| $x^3$ | $x$ | $y$ | $x^2$ | $yx$ | $x^3$ | $yx^2$ |
| $yx^2$ | $y$ | $y$ | $yx$ | $yx$ | $yx^2$ | $yx^2$ |

Задание 3.




## 4. ВЫВОДЫ ПО РАБОТЕ.

В данной лабораторной работе были рассмотрены теоретические сведения об алгебраической операции, полугруппах, подполугруппах. На их основе были составлены алгоритмы построения подполугруппы по таблице Кэли, построения полугруппы бинарных отношений по заданному порождающему множеству, тест Лайта. Была произведена оценка сложности созданных алгоритмов. Они послужили фундаментом для программной реализации, которая впоследствии успешно прошла тестирование, результаты которого были прикреплены к отчету вместе с листингом программы, написанной на языке Python.