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

Комбинаторная теория полугрупп

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

1 Цель работы и порядок ее выполнения

Цель работы — изучение основных понятий теории полугрупп.

Порядок выполнения работы:

1. Рассмотреть понятия полугруппы, подполугруппы и порождающего множества. Разработать алгоритм построения подполугрупп по таблице Кэли.
2. Разработать алгоритм построения полугруппы бинарных отношений по заданному порождающему множеству.
3. Рассмотреть понятия подгруппы, порождающего множества и определяющих соотношений. Разработать алгоритм построения полугруппы по порождающему множеству и определяющим соотношениям.

2 Теоретические сведения

Полугруппа — это алгебра $S = (S, \cdot)$ с одной ассоциативной бинарной операцией $\cdot$, т.е. выполняется $(x \cdot y) \cdot z = x \cdot (y \cdot z)$ для $\forall x, y, z \in S$. Если полугрупповая операция называется умножением (или сложением), то полугруппу называют мультипликативной (или аддитивной).

Подмножество X полугруппы S называется подполугруппой, если X устойчиво относительно операции умножения, т.е. $\forall x, y \in X$ выполняется свойство: $x \cdot y \in X$. В этом случае множество X с ограничением на нем операции умножения исходной полугруппы S образует полугруппу.

2.1 Алгоритм построения подполугруппы по заданному порождающему множеству и таблице Кэли

Вход Порождающее множество S длины n, подмножество множества $subset$, таблица Кэли $C = (a_{ij})$ размерности $n \times n$.

Выход Подполугруппа $\langle X \rangle \subset S$.

1. Создать переменную $x\_current = subset$.
2. Создать переменную $x\_previous = ''$.
3. Цикл пока $x\_current \neq x\_previous$.
   а) Создать пустой список x_l.
   б) Цикл по i от 1 до $|x\_current|$, цикл по j от 1 до $|subset|$.
      i. Создать переменную x и присвоить ей номер позиции $x\_current[i]$ в множестве S.
      ii. Создать переменную y и присвоить ей номер позиции $subset[j]$ в множестве S.
      iii. Добавить в список x_l значение $C[x][y]$.
   в) Присвоить $x\_previous$ значение $x\_current$.
   г) Присвоить $x\_current$ объединение полученных из переменных x_l и $x\_current$ множеств и сделать из $x\_current$ список.
   д) Отсортировать $x\_current$ по возрастанию.
   е) Если $x\_current = x\_previous$, то ответ — $x\_current$.

Трудоемкость алгоритма $O(n^3)$.

В силу общего свойства подалгебр пересечение любого семейства X_i $(i \in I)$ подполугрупп полугруппы S является подполугруппой S и, значит, множество $Sub(S)$ всех подполугрупп полугруппы S является системой замыканий. множество X. Такая полугруппа обозначается символом $\langle X\rangle$ и называется подполугруппой S, порождённой множеством X. При этом множество X называется также **порождающим множеством** подполугруппы $\langle X\rangle$. В частности, если $\langle X\rangle = S$, то X называется порождающим множеством полугруппы S и говорят, что множество X порождает полугруппу S.

2.2 Алгоритм построения полугруппы бинарных отношений по порождающему множеству

Для реализации алгоритма используется вспомогательный алгоритм $insert\_matrix$, представляющий из себя добавление в исходное матричное множество матрично перемноженных подмножеств множества.

Вход Порождающее множество S длины n, k матриц бинарных отношений $M_k = (m_{k_{ij}})$ размерности $n \times n$.

Выход Полугруппа S, таблица Кэли C_s полугруппы S.

1. Создать пустые отсортированные словари $in\_matrices$ и $out\_matrices$
2. Создать переменную $cur = 0$.
3. Создать пустое множество $sets$.
4. Цикл по i от 1 до k.
   а) Присвоить $in\_matrices$ по ключу $char(65 + cur)$ значение M_i.
   б) Переменной cur присвоить значение $cur + 1$.
   в) Присвоить $out\_matrices$ по ключу M_i значение $char(65 + cur)$.
   г) Добавить M_i во множество $sets$.
5. Создать множество $group$ и присвоить копию множества $sets$.
6. Создать переменную $flag$ и присвоить ей значение $True$ (данная переменная требуется для того, чтобы была выполнена хотя бы одна итерация следующего цикла).
7. Создать переменную $cur\_updated = 0$
8. Цикл пока $group \neq sets \mid\mid Flag$.
   а) Переменной $flag$ присвоить значение $False$.
   б) Цикл по i от 1 до $|sets|$, цикл по j от 1 до $|sets|$.
      i. Создать переменную new_matrix и присвоить ей умножение матрицы $sets[i]$ на матрицу $sets[j]$ (алгоритм перемножения матриц был рассмотрен в лабораторной работе №1).
      ii. Если new_matrix не находится в $group$, то:
         A. Присвоить $in\_matrices$ по ключу $char(65{+}cur{+}cur\_updated)$

значение new_matrix.

Б. Переменной $cur\_updated$ присвоить значение $cur\_updated + 1$.

В. Присвоить $out\_matrices$ по ключу new_matrix значение $char(65 + cur + cur\_updated)$.

Г. Добавить во множество $group$ матрицу new_matrix.

9. Если $group = sets$:

   а) Полугруппа получается в результате поиска по ключу в словаре $out\_matrices$, где ключами будут являться элементы множества $sets$.

   б) Множеству $sets$ присвоить значение $insert\_matrix(sets, n)$.

   в) Цикл по i от 1 до $|out\_matrices|$, цикл по j от 1 до $|out\_matrices|$.

      i. Таблица Кэли получается в результате перемножения матриц $in\_matrices$ по индексу $char(65 + i)$ и $in\_matrices$ по индексу $char(65 + j)$. Затем происходит поиск значения в словаре $out\_matrices$ по полученному в результате перемножения матриц ключу.

10. Ответ — полугруппа $sets$ и ее таблица Кэли.

Трудоемкость алгоритма $O(|out\_matrices|^2 n^3)$, так как алгоритм умножения матриц имеет трудоемкость $O(n^3)$.

Для любой конечной полугруппы S найдется такой конечный алфавит A, что для некоторого отображения $\phi : A \to S$ выполняется равенство $\langle \phi(A) \rangle = S$ и, значит, $S \cong A^+/ker\phi$ этом случае множество A называется множеством порождающих символов полугруппы S (относительно отображения $\phi : A \to S$). Если при этом для слов $w_1, w_2 \in A$ выполняется равенство $\phi(w_1) = \phi(w_2)$, т.е. $w_1 \equiv w_2(ker\phi)$, то говорят, что на S выполняется соотношение $w_1 = w_2$ (относительно отображения $\phi : A \to S$).

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

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

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. $(u, v) = (xw_1y, xw_2y)$ для некоторых $x, y \in A^*$;
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 $ker\phi$ полностью определяется отношением ρ и выражение $< A : w_1 = w_2 : (w_1, w_2) \in \rho >$ 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.

2.3 Алгоритм построения полугруппы по порождающему множеству и определяющим соотношениям

Вход Элементы алфавита $Alph$ длины m, определяющие соотношения $rules$ в количестве n штук.

Выход Полугруппа и ее таблица Кэли.

1. Создать пустое множество $new\_semigroup$.
2. Создать список $semigroup$ и заполнить его элементами алфавита $Alph$.
3. Цикл пока $new\_semigroup \neq semigroup$.

   а) Создать пустой список new_elems.

   б) Цикл по i от 1 до $|semigroup|$, цикл по j от 1 до $|semigroup|$.

i. Создать переменную new_elem и присвоить ей значение конкатенации слов $semigroup[i] + semigroup[j]$.

ii. Создать переменную $new\_elem\_copy = ''$

iii. Цикл пока $new\_elem\_copy \neq new\_elem$.

A. Цикл по $key,\ value$ из словаря $rules$.

— Если key находится в new_elem, то в new_elem заменить key на $value$.

iv. Добавить в список new_elems строку new_elem.

в) Множеству $new\_semigroup$ присвоить копию $semigroup$.

г) Цикл по i от 1 до $|new\_elems|$.

i. Если $new\_elems[i]$ не находится в $semigroup$, то добавить $new\_elems[i]$ в $semigroup$.

4. Создать пустой список $matrix$.
5. Цикл по i от 1 до $|semigroup|$.

а) Создать пустой список $matrix\_string$.

б) Цикл по j от 1 до $|semigroup|$.

i. Создать переменную $new\_elem = semigroup[i] + semigroup[j]$.

ii. Создать переменную $new\_elem\_copy =''$.

iii. Цикл пока $new\_elem\_copy \neq new\_elem$.

A. Переменной new_elem_copy присвоить копию new_elem.

Б. Цикл по $key,\ value$ из словаря $rules$.

— Если key находится в new_elem, то заменить в new_elem key на $value$.

в) Добавить new_elem в $matrix\_string$.

6. Добавить $matrix\_string$ в список $matrix$.
7. Ответ — элементы полугруппы $semigroup$ и таблица Кэли $matrix$ полугруппы $semigroup$.

Трудоемкость алгоритма составляет $O(n^n)$ (процесс построения соотношений является бесконечным).

3 Практическая часть

Задание 1

(4) $f = \begin{pmatrix} 1 & 2 & 3 \\ 1 & 1 & 1 \end{pmatrix}$ $\quad g = \begin{pmatrix} 1 & 2 & 3 \\ 3 & 3 & 2 \end{pmatrix}$

Известно, что мн-во преобразований f, g порождает полугруппу $S = \langle f, g \rangle$ преобразований мн-ва X, которая сост. из эл-ов $f, g, f^2, fg, gf, g^2, \dots$

$$f^2 = \begin{pmatrix} 1 & 2 & 3 \\ 1 & 1 & 1 \end{pmatrix} \cdot \begin{pmatrix} 1 & 2 & 3 \\ 1 & 1 & 1 \end{pmatrix} = \begin{matrix} 1 & 2 & 3 \\ \downarrow & \downarrow & \downarrow \\ 1 & 1 & 1 \\ \downarrow & \downarrow & \downarrow \\ 1 & 1 & 1 \end{matrix} = \begin{pmatrix} 1 & 2 & 3 \\ 1 & 1 & 1 \end{pmatrix}$$

$$fg = \begin{pmatrix} 1 & 2 & 3 \\ 1 & 1 & 1 \end{pmatrix} \begin{pmatrix} 1 & 2 & 3 \\ 3 & 3 & 2 \end{pmatrix} = \begin{pmatrix} 1 & 2 & 3 \\ 3 & 3 & 3 \end{pmatrix} \qquad gf = \begin{pmatrix} 1 & 2 & 3 \\ 3 & 3 & 2 \end{pmatrix} \begin{pmatrix} 1 & 2 & 3 \\ 1 & 1 & 1 \end{pmatrix} = \begin{pmatrix} 1 & 2 & 3 \\ 1 & 1 & 1 \end{pmatrix}$$

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

Задание 2

(4) $a = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 \\ 3 & 3 & 2 & 5 & 2 \end{pmatrix}$ $\quad aa = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 \\ 2 & 2 & 3 & 2 & 3 \end{pmatrix}$ $\quad aaa = \begin{matrix} 1 & 2 & 3 & 4 & 5 \\ \downarrow & \downarrow & \downarrow & \downarrow & \downarrow \\ 2 & 2 & 3 & 2 & 3 \\ \downarrow & \downarrow & \downarrow & \downarrow & \downarrow \\ 3 & 3 & 2 & 3 & 2 \end{matrix}$

$aaaa = \begin{matrix} 1 & 2 & 3 & 4 & 5 \\ 2 & 2 & 3 & 2 & 3 \end{matrix}$

Видно, что $aaaa \Rightarrow aa$, тогда на 4 преобразовании наблюдается цикличность. Тогда, если считать элементы, начиная с 2-х, то каждый $2k$-й элемент будет иметь преобразование $\begin{matrix} 1 & 2 & 3 & 4 & 5 \\ 2 & 2 & 3 & 2 & 3 \end{matrix}$, где $k \in N$, т.е. $T = 2$

Задание 3

$S = \langle x, y : xy = yx, x^2 = y, y^3 = y \rangle$

Выделим полную систему представителей классов конгруэнции ε, которая опр. соотношениями данного копредставления. Для этого последовательно рассмотрим словаданной длины и выделим те, которые не будут экв. м/д. собой отн. конг. ε.

Сначала рассм. слова длины 1: x, y – не экв. м/д. собой отн. конг. ε.

Длины 2: $x^2 = y$, xy, $yx = xy$, y^2, только слова y^2, xy не экв. отн. конг. ε и др. ранее выделенным словам.

Длины 3: $x^3 = x^2x = yx$, $xyx = x^2y = y^2$, $x^2y = y^2$, xy^2, $y^3 = y$

Из этих слов только слово xy^2 не экв отн конг ε и ранее выд. сл.

Длины 4: $xy^2x = x^2y^2 = yy^2 = y^3 = y$,

$xy^2y = xy^3 = xy$ – все эти слова экв. отн. конг ε ранее выделенным словам.

Значит $S = \{x, y, y^2, xy, xy^2\}$ – полная сист. предст. классов конг. ε. Операция • таких слов опр. с точностью до конг. ε по след. табл. Кэли:

*	x	y	y^2	xy	xy^2
x	y	xy	xy^2	y^2	y
y	xy	y^2	y	xy^2	xy
y^2	xy^2	y	y^2	xy	xy^2
xy	y^2	xy^2	xy	y	y
xy^2	y	xy	xy^2	y	y

4 Программная реализация рассмотренных алгоритмов

4.1 Результаты тестирования программы

```
Выберите действие:
Построить подполугруппу по таблице Кэли (1)
Построить полугруппу бинарных отношений по порождающему множеству (2)
Построить полугруппу по порождающему множеству и определяющим соотношениям (3)
1
Введите элементы множества: 1 2 3 4
Введите элементы подмножества: 1 2
Введите значения таблицы Кэли множества, для которой будет строится подполугруппа:
  1 2 3 4
1 1 2 2 3
2 3 4 4 3
3 1 2 2 2
4 1 2 3 4
Подполугруппа: {1, 2, 3, 4}
```

Рисунок 1

```
Построить полугруппу по порождающему множеству и определяющим соотношениям (3)
2
Введите количество элементов на множестве: 3
Введите элементы порождающего множества (3): 1 2 3
"Введите количество матриц в порождающем множестве: 2
Введите матрицу  A
  1 2 3
1 1 1 0
2 1 0 0
3 0 1 0
Введите матрицу  B
  1 2 3
1 1 1 1
2 0 0 0
3 0 0 0
Полученная полугруппа:
G :
1 1 0
1 1 0
1 1 0
```

Рисунок 2

```
F :
1 1 1
1 1 1
1 1 1


E :
1 1 0
0 0 0
0 0 0


D :
1 1 1
1 1 1
0 0 0


B :
1 1 1
0 0 0
0 0 0


C :
1 1 0
1 1 0
1 0 0
```

```
A :
1 1 0
1 0 0
0 1 0


H :
1 1 0
1 1 0
0 0 0


Таблица Кэли:
  A B C D E F G H
A C D G F H F G G

B E B E B E B E E

C G F G F G F G G

D H D H D H D H H

E E B E B E B E E

F G F G F G F G G

G G F G F G F G G
```

```
Построить полугруппу по порождающему множеству и определяющим соотношениям (3)
3
Введите элементы алфавита: a b
Введите количество определяющих соотношений: 3
Введите элементы 1-го соотношения:
ab = ba
Введите элементы 2-го соотношения:
aaa = a
Введите элементы 3-го соотношения:
bb = b
Полугруппа: {a, b, aa, ba, baa}
Таблица Кэли:
aa  ba  a   baa ba
ba  b   baa ba  baa
a   baa aa  ba  baa
baa ba  ba  baa ba
ba  baa baa ba  baa
```

Рисунок 5

4.2 Коды программ, реализующих рассмотренные алгоритмы

```
import sortedcontainers as s
import numpy as np


def multiply_matrices(a, b, n):
    ab = [[0 for _ in range(n)] for _ in range(n)]
    for i in range(n):
        for j in range(n):
            for k in range(n):
                if a[i][k] == 1 and b[k][j] == 1:
                    ab[i][j] = 1
                    break
    return ab


def insert_matrix(sets, n):
    for subset in sets:
        for subset1 in sets:
            sets.add(make_tuples(multiply_matrices(subset, subset1, n)))
    return sets


def make_tuples(matrix):
    return tuple(tuple(row) for row in matrix)

```

```python

def print_out(in_matrices, out_matrices, i , j, n):
    print(chr(ord(out_matrices[make_tuples(multiply_matrices(in_matrices[chr(65 + i)],
                                                             in_matrices[chr(65 + j)], n))]) - 1),
                                                             end=' ')


def task2():
    n = int(input('Введите количество элементов на множестве: '))
    st = [int(value) for value in input(f'Введите элементы порождающего множества ({n}): ').split()]
    m = int(input('"Введите количество матриц в порождающем множестве: '))

    in_matrices = s.SortedDict()
    out_matrices = s.SortedDict()
    sets = set()
    k = 0
    for i in range(0, m):
        print('Введите матрицу ', chr(65 + k))
        print(" ", *st)
        matrix = make_tuples([list(map(int, input(f"{st[i]} ").split())) for i in range(n)])
        sets.add(matrix)
        in_matrices[chr(65 + k)] = matrix
        k += 1
        out_matrices[matrix] = chr(65 + k)

    group = sets.copy()
    k_updated = 0
    while True:
        for value in sets:
            for value1 in sets:
                new_matrix = make_tuples(multiply_matrices(value, value1, n))
                if new_matrix not in group:
                    in_matrices[chr(65 + k + k_updated)] = new_matrix
                    k_updated += 1
                    out_matrices[new_matrix] = chr(65 + k + k_updated)
```

```
                    group.add(new_matrix)

        if group == sets:
            sets = insert_matrix(sets, n)
            print('Полученная полугруппа:')
            for subset in sets:
                print(chr(ord(out_matrices[subset]) - 1), ':')
                for i in range(len(subset)):
                    print(*subset[i])
                print('\n')
            print('Таблица Кэли: ')
            print(' ', *[chr(65 + i) for i in range(len(out_matrices))])
            for i in range(len(out_matrices)):
                print(chr(65 + i), end=' ')
                for j in range(len(out_matrices)):
                    print_out(in_matrices, out_matrices, i, j, n)
                print('\n')
            return
        else:
            sets = group.copy()


def check_associative(cayley, input_list):
    n = len(input_list)
    for a in range(n):
        for x in range(n):
            for z in range(n):
                if cayley[x, input_list.index(str(cayley[a, z]))] \
                        != cayley[input_list.index(str(cayley[x, a])), z]:
                    return False
    return True


def build_cayley():
    input_list = input('Введите элементы алфавита: ').split()
    n = int(input('Введите количество определяющих соотношений: '))
    rules = {}

    for i in range(n):
        print(f'Введите элементы {i + 1}-го соотношения:')
        elems = input().replace(" =", "").split()
```

```
        first_elem, second_elem = elems
        rules[first_elem] = second_elem

    semigroup = input_list.copy()
    while True:
        new_elems = []
        for elem_1 in semigroup:
            for elem_2 in semigroup:
                new_elem = elem_1 + elem_2
                while True:
                    new_elem_copy = str(new_elem)
                    for first_elem, second_elem in rules.items():
                        if first_elem in new_elem:
                            new_elem = new_elem.replace(first_elem,
                             ↪  second_elem)
                    if new_elem_copy == new_elem:
                        break
                new_elems.append(new_elem)
        new_semigroup = set(semigroup.copy())
        for new_elem in new_elems:
            if new_elem not in semigroup:
                semigroup.append(new_elem)
        if new_semigroup == set(semigroup):
            break

    semigroup = list(semigroup)
    matrix = []
    for elem_1 in semigroup:
        matrix_string = []
        for elem_2 in semigroup:
            new_elem = elem_1 + elem_2
            while True:
                new_elem_copy = str(new_elem)
                for first_elem, second_elem in rules.items():
                    if first_elem in new_elem:
                        new_elem = new_elem.replace(first_elem, second_elem)
                if new_elem_copy == new_elem:
                    break
            matrix_string.append(new_elem)
        matrix.append(matrix_string)

```

```
    group_length = len(semigroup)
    matrix = np.array(matrix).reshape(group_length, group_length)
    print('Полугруппа: {', end='')
    print(*semigroup, sep=', ', end='')
    print('}')
    print('Таблица Кэли:')
    for i in range(len(matrix)):
        print(*matrix[i], sep='\t')


def build_sub_semigroup():
    input_list = input("Введите элементы множества: ").split()
    n = len(input_list)

    subset = input("Введите элементы подмножества: ").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_current = subset.copy()
    while True:
        x_l = []
        for x in x_current:
            for y in subset:
                x_l.append(cayley_table[input_list.index(x)][input_list.index(y)])
        previous_x = x_current.copy()
        x_current = list(set(x_current).union(set(x_l)))
        x_current.sort()
        if previous_x == x_current:
            break

    print("Подполугруппа: {", end='')
    print(*x_current, sep=', ', end='}\n')


def main():
```

```
    print("Выберите действие: ")
    print("Построить подполугруппу по таблице Кэли (1)")
    print("Построить полугруппу бинарных отношений по порождающему множеству (2)")
    print("Построить полугруппу по порождающему множеству и определяющим соотношениям (3)")
    action = input()
    if action:
        if action == "1":
            build_sub_semigroup()
        if action == '2':
            task2()
        if action == '3':
            build_cayley()


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

ЗАКЛЮЧЕНИЕ

В лабораторной работе были рассмотрены теоретические сведения о подгруппах, полугруппах, подполугруппах и порождающем множестве. На их основе были составлены алгоритмы построения подполугруппы по таблице Кэли, построения полугруппы бинарных отношений и ее таблицы Кэли по заданному порождающему множеству, построения полугруппы и ее таблицы Кэли по порождающему множеству и определяющим соотношениям. Для всех алгоритмов была произведена оценка трудоемкости.