

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.

подпись, дата

В. А. Молчанов

Саратов 2022

СОДЕРЖАНИЕ

1	Цель работы и порядок ее выполнения	3
2	Теоретические сведения	4
2.1	Алгоритм построения подполугруппы по заданному порождающему множеству и таблице Кэли	4
2.2	Алгоритм построения полугруппы бинарных отношений по порождающему множеству	5
2.3	Алгоритм построения полугруппы по порождающему множеству и определяющим соотношениям	7
3	Практическая часть	9
4	Программная реализация рассмотренных алгоритмов	13
4.1	Результаты тестирования программы	13
4.2	Коды программ, реализующих рассмотренные алгоритмы	16
	ЗАКЛЮЧЕНИЕ	22

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 замы-

каний. 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_{gr} , таблица Кэли C_{sgr} полугруппы S_{gr} .

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 $group \neq sets \parallel flag$.
 - 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:
 - А. 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 Sgr 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 $|out_matrices|$.

и. 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 Sgr и ее таблица Кэли C_{sgr} .

Трудоёмкость алгоритма $O(2^{n^2} n^3)$, так как алгоритм умножения матриц имеет трудоёмкость $O(n^3)$.

Для любой конечной полугруппы S найдется такой конечный алфавит A , что для некоторого отображения $\phi : A \rightarrow S$ 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 $\phi : A \rightarrow S$). 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 $\phi : A \rightarrow S$).

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 $\langle A : w_1 = w_2 : (w_1, w_2) \in \rho \rangle$ называется **копределением 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 $w_1 = a_1 \dots a_n$ и $w_2 = b_1 \dots b_m$ операция конкатенации ставит в соответствие слово $w_1 \cdot w_2 = a_1 \dots a_n b_1 \dots b_m$. 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 .
 - B. 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 $semigroup$.

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

3 Практическая часть

Задание 1

④ $f = \begin{pmatrix} 123 \\ 111 \end{pmatrix}, g = \begin{pmatrix} 123 \\ 332 \end{pmatrix}$

$$f^2 = \begin{pmatrix} 123 \\ 111 \end{pmatrix} \begin{pmatrix} 123 \\ 111 \end{pmatrix} = \begin{pmatrix} 123 \\ 111 \end{pmatrix} = f$$

$$g^2 = \begin{pmatrix} 123 \\ 332 \end{pmatrix} \begin{pmatrix} 123 \\ 332 \end{pmatrix} = \begin{pmatrix} 123 \\ 223 \end{pmatrix} \quad g^3 = \begin{pmatrix} 123 \\ 223 \end{pmatrix} \begin{pmatrix} 123 \\ 332 \end{pmatrix} = \begin{pmatrix} 123 \\ 332 \end{pmatrix} = g$$

$$gf = \begin{pmatrix} 123 \\ 332 \end{pmatrix} \begin{pmatrix} 123 \\ 111 \end{pmatrix} = \begin{pmatrix} 123 \\ 111 \end{pmatrix} = f$$

$$fg = \begin{pmatrix} 123 \\ 111 \end{pmatrix} \begin{pmatrix} 123 \\ 332 \end{pmatrix} = \begin{pmatrix} 123 \\ 333 \end{pmatrix} \quad fgg = \begin{pmatrix} 123 \\ 333 \end{pmatrix} \begin{pmatrix} 123 \\ 332 \end{pmatrix} = \begin{pmatrix} 123 \\ 222 \end{pmatrix}$$

$$fggg = \begin{pmatrix} 123 \\ 222 \end{pmatrix} \begin{pmatrix} 123 \\ 332 \end{pmatrix} = \begin{pmatrix} 123 \\ 333 \end{pmatrix} = fg$$

Таблица Кэли:

	\cdot	f	g	g^2	fg	fgg
\cdot	$\begin{pmatrix} 123 \\ 111 \end{pmatrix}$	$\begin{pmatrix} 123 \\ 111 \end{pmatrix}$	$\begin{pmatrix} 123 \\ 332 \end{pmatrix}$	$\begin{pmatrix} 123 \\ 223 \end{pmatrix}$	$\begin{pmatrix} 123 \\ 333 \end{pmatrix}$	$\begin{pmatrix} 123 \\ 222 \end{pmatrix}$
f	$\begin{pmatrix} 123 \\ 111 \end{pmatrix}$	$\begin{pmatrix} 123 \\ 111 \end{pmatrix}$	$\begin{pmatrix} 123 \\ 332 \end{pmatrix}$	$\begin{pmatrix} 123 \\ 223 \end{pmatrix}$	$\begin{pmatrix} 123 \\ 333 \end{pmatrix}$	$\begin{pmatrix} 123 \\ 222 \end{pmatrix}$
g	$\begin{pmatrix} 123 \\ 332 \end{pmatrix}$	$\begin{pmatrix} 123 \\ 332 \end{pmatrix}$	$\begin{pmatrix} 123 \\ 332 \end{pmatrix}$	$\begin{pmatrix} 123 \\ 223 \end{pmatrix}$	$\begin{pmatrix} 123 \\ 333 \end{pmatrix}$	$\begin{pmatrix} 123 \\ 222 \end{pmatrix}$
g^2	$\begin{pmatrix} 123 \\ 223 \end{pmatrix}$	$\begin{pmatrix} 123 \\ 223 \end{pmatrix}$	$\begin{pmatrix} 123 \\ 332 \end{pmatrix}$	$\begin{pmatrix} 123 \\ 223 \end{pmatrix}$	$\begin{pmatrix} 123 \\ 333 \end{pmatrix}$	$\begin{pmatrix} 123 \\ 222 \end{pmatrix}$
fg	$\begin{pmatrix} 123 \\ 333 \end{pmatrix}$	$\begin{pmatrix} 123 \\ 333 \end{pmatrix}$	$\begin{pmatrix} 123 \\ 332 \end{pmatrix}$	$\begin{pmatrix} 123 \\ 223 \end{pmatrix}$	$\begin{pmatrix} 123 \\ 333 \end{pmatrix}$	$\begin{pmatrix} 123 \\ 222 \end{pmatrix}$
fgg	$\begin{pmatrix} 123 \\ 222 \end{pmatrix}$	$\begin{pmatrix} 123 \\ 222 \end{pmatrix}$	$\begin{pmatrix} 123 \\ 332 \end{pmatrix}$	$\begin{pmatrix} 123 \\ 223 \end{pmatrix}$	$\begin{pmatrix} 123 \\ 333 \end{pmatrix}$	$\begin{pmatrix} 123 \\ 222 \end{pmatrix}$

Задача 2

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

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

Видно, что $aaaa \Rightarrow aa$, поэтому на 4

преобразования наблюдается цикличность

Тогда, если считать элементы, начиная

с 2-го, то каждый 2k-й элемент будет

состоять преобразование $\begin{pmatrix} 1 & 2 & 3 & 4 & 5 \\ 2 & 2 & 3 & 2 & 3 \end{pmatrix}$, где $k \in \mathbb{N}$,

$$\text{т.е. } T = 2$$

Индекс: 2 ~~2~~

Период: 2

Задача 3

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

Введем на множестве представителей классов конгруэнции ε , которая определена соотношениями данного представления. Для этого поинтерпретируем рассматриваемые слова длины 1 и выделим те, которые не будут экв. м.г. собой отн. к ε .

Сначала рассм. слова длины 1: x, y — не экв. м.г. собой отн. к ε .

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

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

Из этих слов только слово xy^2 не экв. отн. к ε ранее выдел.

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

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

Значит $S = \{x, y, y^2, xy, xy^2\}$ — полная

сист. предст. классов к ε : Отсюда • полная слов отн. с точностью до к ε по след. табл. к ε .

$*$	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 строиться 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

```

1 import sortedcontainers as s
2 import numpy as np
3
4
5 def multiply_matrices(a, b, n):
6     ab = [[0 for _ in range(n)] for _ in range(n)]
7     for i in range(n):
8         for j in range(n):
9             for k in range(n):
10                 if a[i][k] == 1 and b[k][j] == 1:
11                     ab[i][j] = 1
12                     break
13     return ab
14
15
16 def insert_matrix(sets, n):
17     for subset in sets:
18         for subset1 in sets:
19             sets.add(make_tuples(multiply_matrices(subset, subset1, n)))
20     return sets
21
22
23 def make_tuples(matrix):
24     return tuple(tuple(row) for row in matrix)
25

```



```

26
27 def print_out(in_matrices, out_matrices, i , j, n):
28
    ↪ print(chr(ord(out_matrices[make_tuples(multiply_matrices(in_matrices[chr(65
    ↪ + i)]),
29
                                                                    in_matrices[chr(65
                                                                    ↪ + j)], n)))] -
                                                                    ↪ 1),
                                                                    end= ' ')
30
31
32
33 def task2():
34     n = int(input('Введите количество элементов на множестве: '))
35     st = [int(value) for value in input(f'Введите элементы порождающего
    ↪ множества ({n}): ').split())
36     m = int(input('Введите количество матриц в порождающем множестве: '))
37
38     in_matrices = s.SortedDict()
39     out_matrices = s.SortedDict()
40     sets = set()
41     k = 0
42     for i in range(0, m):
43         print('Введите матрицу ', chr(65 + k))
44         print(" ", *st)
45         matrix = make_tuples([list(map(int, input(f" {st[i]} ").split())) for
    ↪ i in range(n)])
46         sets.add(matrix)
47         in_matrices[chr(65 + k)] = matrix
48         k += 1
49         out_matrices[matrix] = chr(65 + k)
50
51     group = sets.copy()
52     k_updated = 0
53     while True:
54         for value in sets:
55             for value1 in sets:
56                 new_matrix = make_tuples(multiply_matrices(value, value1, n))
57                 if new_matrix not in group:
58                     in_matrices[chr(65 + k + k_updated)] = new_matrix
59                     k_updated += 1
60                     out_matrices[new_matrix] = chr(65 + k + k_updated)

```

```

61             group.add(new_matrix)
62
63     if group == sets:
64         sets = insert_matrix(sets, n)
65         print('Полученная полугруппа:')
66         for subset in sets:
67             print(chr(ord(out_matrices[subset]) - 1), ':')
68             for i in range(len(subset)):
69                 print(*subset[i])
70             print('\n ')
71         print('Таблица Кэли: ')
72         print(' ', *[chr(65 + i) for i in range(len(out_matrices))])
73         for i in range(len(out_matrices)):
74             print(chr(65 + i), end=' ')
75             for j in range(len(out_matrices)):
76                 print_out(in_matrices, out_matrices, i, j, n)
77             print('\n ')
78         return
79     else:
80         sets = group.copy()
81
82
83 def check_associative(cayley, input_list):
84     n = len(input_list)
85     for a in range(n):
86         for x in range(n):
87             for z in range(n):
88                 if cayley[x, input_list.index(str(cayley[a, z]))] \
89                     != cayley[input_list.index(str(cayley[x, a])), z]:
90                     return False
91     return True
92
93
94 def build_cayley():
95     input_list = input('Введите элементы алфавита: ').split()
96     n = int(input('Введите количество определяющих соотношений: '))
97     rules = {}
98
99     for i in range(n):
100         print(f'Введите элементы {i + 1}-го соотношения:')
101         elems = input().replace(" =", "").split()

```

```

102         first_elem, second_elem = elems
103         rules[first_elem] = second_elem
104
105     semigroup = input_list.copy()
106     while True:
107         new_elems = []
108         for elem_1 in semigroup:
109             for elem_2 in semigroup:
110                 new_elem = elem_1 + elem_2
111                 while True:
112                     new_elem_copy = str(new_elem)
113                     for first_elem, second_elem in rules.items():
114                         if first_elem in new_elem:
115                             new_elem = new_elem.replace(first_elem,
116                                                         ↪ second_elem)
117                     if new_elem_copy == new_elem:
118                         break
119                     new_elems.append(new_elem)
120         new_semigroup = set(semigroup.copy())
121         for new_elem in new_elems:
122             if new_elem not in semigroup:
123                 semigroup.append(new_elem)
124         if new_semigroup == set(semigroup):
125             break
126
127     semigroup = list(semigroup)
128     matrix = []
129     for elem_1 in semigroup:
130         matrix_string = []
131         for elem_2 in semigroup:
132             new_elem = elem_1 + elem_2
133             while True:
134                 new_elem_copy = str(new_elem)
135                 for first_elem, second_elem in rules.items():
136                     if first_elem in new_elem:
137                         new_elem = new_elem.replace(first_elem, second_elem)
138                 if new_elem_copy == new_elem:
139                     break
140             matrix_string.append(new_elem)
141     matrix.append(matrix_string)

```

```

142     group_length = len(semigroup)
143     matrix = np.array(matrix).reshape(group_length, group_length)
144     print('Полугруппа: {', end='')
145     print(*semigroup, sep=', ', end='')
146     print('}')
147     print('Таблица Кэли:')
148     for i in range(len(matrix)):
149         print(*matrix[i], sep=' \t ')
150
151
152 def build_sub_semigroup():
153     input_list = input("Введите элементы множества: ").split()
154     n = len(input_list)
155
156     subset = input("Введите элементы подмножества: ").split()
157
158     print("Введите значения таблицы Кэли множества, для которой будет строится
159     ↪ подполугруппа: ")
159     print(" ", *input_list)
160     cayley = [list(map(str, input(f"{{input_list[i]}} ").split())) for i in
161     ↪ range(n)]
161     cayley_table = np.array(cayley).reshape(n, n)
162
163     x_current = subset.copy()
164     while True:
165         x_1 = []
166         for x in x_current:
167             for y in subset:
168
169                 ↪ x_1.append(cayley_table[input_list.index(x)][input_list.index(y)])
169         previous_x = x_current.copy()
170         x_current = list(set(x_current).union(set(x_1)))
171         x_current.sort()
172         if previous_x == x_current:
173             break
174
175     print("Подполугруппа: {", end='')
176     print(*x_current, sep=', ', end='} \n ')
177
178
179 def main():

```

```

180     print("Выберите действие: ")
181     print("Построить подполугруппу по таблице Кэли (1)")
182     print("Построить полугруппу бинарных отношений по порождающему множеству  

    ↪ (2)")
183     print("Построить полугруппу по порождающему множеству и определяющим  

    ↪ соотношениям (3)")
184     action = input()
185     if action:
186         if action == "1":
187             build_sub_semigroup()
188         if action == '2':
189             task2()
190         if action == '3':
191             build_cayley()
192
193
194 if __name__ == "__main__":
195     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.