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

ИДЕАЛЫ ПОЛУГРУПП

ЛАБОРАТОРНАЯ РАБОТА

студента 3 курса 331 группы
специальности 100501 — Компьютерная безопасность
факультета КНиИТ
Окунькова Сергея Викторовича

Проверил
аспирант ____________ В. Н. Кутин

Саратов 2022

СОДЕРЖАНИЕ

1 Постановка задачи . . . 3

2 Теоретические сведения по рассмотренным темам с их обоснованием . . . 4

3 Результаты работы . . . 6

3.1 Алгоритм 1 – Построение правых идеалов полугруппы по таблице Кэли . . . 6

3.2 Алгоритм 2 – Построение левых идеалов полугруппы по таблице Кэли . . . 6

3.3 Алгоритм 3 – Построение двусторонних идеалов полугруппы по таблице Кэли . . . 6

3.4 Алгоритм 4 – Построение отношения Грина по таблице Кэли . . . 7

3.5 Алгоритм 5 – Построение «egg-box»-диаграммы . . . 7

3.6 Алгоритм 6 – Построение полугруппы по порождающему множеству и определяющим соотношениям . . . 8

3.7 Коды программ, реализующей рассмотренные алгоритмы . . . 8

3.8 Результаты тестирования программ . . . 15

3.9 Ответы на задачи . . . 16

ЗАКЛЮЧЕНИЕ . . . 20

1 Постановка задачи

Цель работы: изучение строения полугрупп с помощью отношений Грина.

Порядок выполнения работы:

1. Рассмотреть понятия идеалов полугруппы. Разработать алгоритмы построения идеалов полугруппы по таблице Кэли.
2. Рассмотреть понятия и свойства отношений Грина на полугруппах.
3. Разработать алгоритмы вычисления отношений Грина и построения «egg-box»-картины конечной полугруппы.

2 Теоретические сведения по рассмотренным темам с их обоснованием

Пусть S – произвольная полугруппа.

Определение 1. 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 любых $x, y, z \in S$.

Определение 2. Непустое подмножество $I \subset S$ называется правым (левым) идеалом полугруппы S, если для любых $x \in I, y \in S$ выполняется условие: $xy \in I$ ($yx \in I$), т.е. $I \cdot S \subset I$ ($S \cdot I \subset I$). Если I – одновременно левый и правый идеал полугруппы S, то I называется двусторонним идеалом (или просто идеалом) полугруппы S. Ясно, что в коммутативной полугруппе S все эти определения совпадают.

Лемма 1. Множество всех идеалов IdS (соответственно, левых идеалов $LIdS$ или правых идеалов $RIdS$) любой полугруппы S является системой замыкания. Пусть X – подмножество полугруппы S. Тогда наименьший правый идеал полугруппы S, содержащий подмножество X, равен $(X] = XS^1 = X \cup XS$, наименьший левый идеал полугруппы S, содержащий подмножество X, равен $[X) = S^1X = X \cup SX$ и наименьший идеал полугруппы S, содержащий подмножество X, равен $[X] = S^1XS^1 = X \cup XS \cup SX \cup SXS$.

В частности, любой элемент $a \in S$ определяет наименьшие правый, левый и двусторонний идеалы: $(a] = aS^1$, $[a) = S^1a$ и $[a] = S^1aS^1$, которые называются главными (соответственно, правыми, левыми и двусторонними) идеалами. Минимальные относительно теоретико-множественного включения идеалы (левые или правые идеалы) называются минимальными идеалами (минимальными левыми или правыми идеалами).

Лемма 2. Если полугруппа имеет минимальный идеал, то он является ее наименьшим идеалом и называется ядром полугруппы.

Пример: В полугруппе натуральных чисел с операцией сложения $\mathbf{N} = (\mathbf{N}, +)$ главные идеалы $(n] = n, n+1, n+2, \ldots$ образуют бесконечную последовательность с пустым пересечением.

Отображения $f : a \mapsto [a]$, $f_r : a \mapsto (a]$, $f_l : a \mapsto [a)$, $a \in S$ определяют ядра $\mathfrak{J} = ker f$, $\mathfrak{R} = ker f_r$, $\mathfrak{L} = ker f_l$ по формулам:

$$(a,b) \in \mathfrak{J} \Longleftrightarrow [a] = [b],$$
$$(a,b) \in \mathfrak{R} \Longleftrightarrow (a] = (b],$$
$$(a,b) \in \mathfrak{L} \Longleftrightarrow [a) = [b).$$

Все эти отношения, а также отношения $\mathfrak{D} = \mathfrak{R} \vee \mathfrak{L}$, $\mathfrak{H} = \mathfrak{R} \cap \mathfrak{L}$ являются эквивалентностями на множестве S, которые называются **отношениями Грина** полугруппы S. Классы этих эквивалентностей, порожденные элементом $a \in S$, обозначаются J_a, R_a, L_a, D_a и H_a, соответственно.

Лемма 3. Отношения Грина полугруппы S удовлетворяют следующим свойствам:

1. эквивалентность $\mathfrak{R}$ регулярна слева и эквивалентность $\mathfrak{L}$ регулярна справа, т.е. $(a,b) \in \mathfrak{R} \Rightarrow (xa, xb) \in \mathfrak{R}$ и $(a,b) \in \mathfrak{L} \Rightarrow (ax, bx) \in \mathfrak{L}$ для любых $x \in S$;
2. эквивалентности $\mathfrak{R}$, $\mathfrak{L}$ коммутируют;
3. $\mathfrak{D} = \mathfrak{R} \cdot \mathfrak{L} = \mathfrak{L} \cdot \mathfrak{R}$;
4. если полугруппа S конечна, то $\mathfrak{D} = \mathfrak{J}$;
5. любой класс $\mathfrak{D}$ эквивалентности $\mathfrak{D}$ можно изобразить с помощью следующей «egg-box»-диаграммы, клетки которой являются классами эквивалентности $\mathfrak{H}$, лежащими в $\mathfrak{D}$.

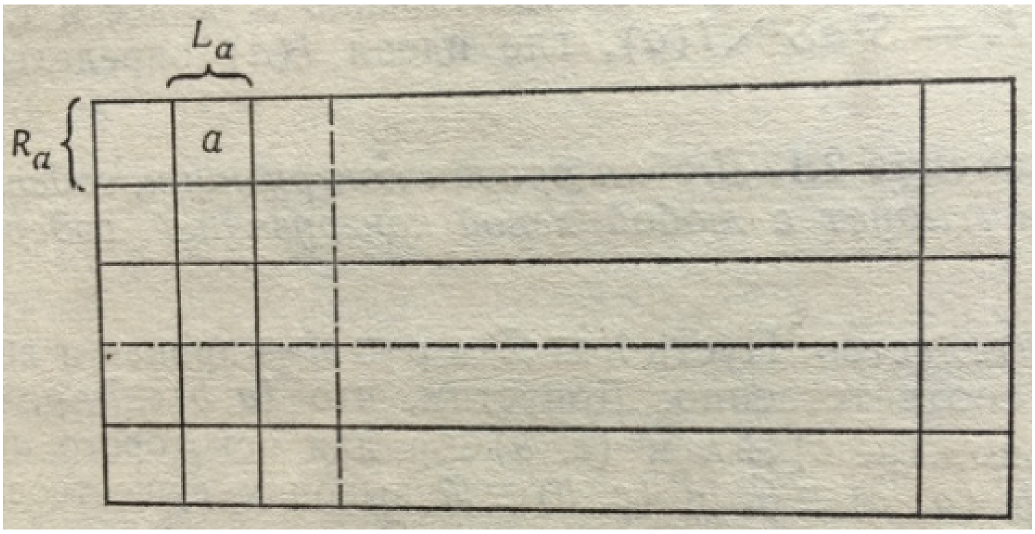


Рисунок 1 – «egg-box»-диаграмма

3 Результаты работы

3.1 Алгоритм 1 – Построение правых идеалов полугруппы по таблице Кэли

Вход: Конечная полугруппа S с таблицей Кэли $A = (a_{ij})$ размерности $n \times n$ и элементом $x \in S$.

Выход: Правый идеал Id_R полугруппы S, порожденный элементом $x \in S$.

Шаг 1. Инициализировать пустое множество $Id_R = \{\}$. Получить индекс $index$ заданного элемента $x \in semigroup$. Стоит отметить, что полугруппа S представлена в виде списка $semigroup$.

Шаг 2. Пройти по элементам строки таблицы Кэли $a[index][i]$, где $0 \leq i < n$. Если элемент $a[index][i]$ $(0 \leq i < n)$ еще не содержится в множестве Id_R, то он добавляется в это множество.

Шаг 3. Вернуть множество Id_R в качестве ответа.

Оценка сложности алгоритма равна $O(nlogn)$.

3.2 Алгоритм 2 – Построение левых идеалов полугруппы по таблице Кэли

Вход: Конечная полугруппа S с таблицей Кэли $A = (a_{ij})$ размерности $n \times n$ и элементом $x \in S$.

Выход: Левый идеал Id_L полугруппы S, порожденный элементом $x \in S$.

Шаг 1. Инициализировать пустое множество $Id_L = \{\}$. Получить индекс $index$ заданного элемента $x \in semigroup$. Стоит отметить, что полугруппа S представлена в виде списка $semigroup$.

Шаг 2. Пройти по элементам столбца таблицы Кэли $a[i][index]$, где $0 \leq i < n$. Если элемент $a[i][index]$ $(0 \leq i < n)$ еще не содержится в множестве Id_L, то он добавляется в это множество.

Шаг 3. Вернуть множество Id_L в качестве ответа.

Оценка сложности алгоритма равна $O(nlogn)$.

3.3 Алгоритм 3 – Построение двусторонних идеалов полугруппы по таблице Кэли

Вход: Конечная полугруппа S с таблицей Кэли $A = (a_{ij})$ размерности $n \times n$ и элементом $x \in S$.

Выход: Двусторонний идеал Id полугруппы S, порожденный элементом $x \in S$.

Шаг 1. Вызвать алгоитм 1 и 2 для таблицей Кэли $A = (a_{ij})$ и элемента x, в результате которых получаем множества RI и LI.

Шаг 2. Вернуть множество Id в качестве ответа, которое будет представлять из себя объединение множеств RI и LI.

Оценка сложности алгоритма равна $O(nlogn)$.

3.4 Алгоритм 4 – Построение отношения Грина по таблице Кэли

Вход: Конечная полугруппа S с таблицей Кэли $A = (a_{ij})$ размерности $n \times n$ и элементом $x \in S$.

Выход: Матрица $D = (d_{ij})$ отношения Грина.

Шаг 1. Используя алгоритм 1, построим список $right\_ideals$, элементы которого будут правые идеалы $Id_{R_0}, \dots, Id_{R_{n-1}}$ порожденные элементами $x_i \in S$, где $0 \leq i < n$. Аналогично, используя алгоритм 2, построим список $left\_ideals$, элементы которого будут правые идеалы $Id_{L_0}, \dots, Id_{L_{n-1}}$ порожденные элементами $x_i \in S$, где $0 \leq i < n$.

Шаг 2. Построим матрицу $R = (r_{ij})$. Пусть $i = 0$ и $el = right\_ideals[i]$. Необходимо пройти по всем элементам Id_{R_j} $(0 \leq j < n)$ списка $right\_ideals$, чтобы выполнить следующее условие: если $el = Id_{R_j}$ $(0 \leq j < n)$, то $d[i][j] = 1$, в противном случае – $d[i][j] = 0$. После того, как $j = n$, необходимо присвоить $i = i + 1$ и осуществить проход по элементам $Id_{R_0}, \dots, Id_{R_{n-1}}$ списка $right\_ideals$ повторно. В итоге получим матрицу R.

Шаг 3. Аналогично построим матрицу $L = (l_{ij})$, только уже используя список $left\_ideals$. В итоге получаем матрицу L.

Шаг 4. Построим матрицу $D = R + L$, т.е. $d[i][j] = r[i][j] + l[i][j]$, где $0 \leq i, j < n$. Учитывая, что если $r[i][j] = l[i][j] = 1$, то $d[i][j] = 1$.

Шаг 5. Вернуть в качестве ответа матрицу D, так как она, в свою очередь, является представлением отношения Грина.

Оценка сложности алгоритма равна $O(n^2)$.

3.5 Алгоритм 5 – Построение «egg-box»-диаграммы

Вход: Конечная полугруппа S с таблицей Кэли $A = (a_{ij})$ размерности $n \times n$ и элементом $x \in S$.

Выход: «egg-box»-картина конечной полугруппы S.

Шаг 1. Запустив последовательно алгоритмы 1, 2 и 4, получим отношение Грина, выраженного матрицей $D = (d_{ij})$ (обстрактно его можно представть как граф).

Шаг 2. Необходимо в матрице D найти все компоненты сильной связности. Для этого можно использовать алгоритм конденсации графа или алгоритм Тарьяна. В результате получаем список egg_box_list, состоящего из элементов $x_i \in S$, $(0 \leq i < n)$ и элемента 1, так как «egg-box»-картина строится по полугруппе с внешне присоединенной единицей $S^1 = S \cup \{1\}$.

Оценка сложности равна оценке сложность алгоритма 6, т.е. $O(n + n) = O(n)$

3.6 Алгоритм 6 – Построение полугруппы по порождающему множеству и определяющим соотношениям

Вход: Конечное множество символов A мощности n и конечное множество R определяющих соотношений мощности m.

Выход: Полугруппа $\langle A|R \rangle$.

Шаг 1. Необходимо инициализировать список $semigroup = []$, в который будут добавлены все элементы $a \in A$.

Шаг 2. Инициализировать список $elements_{new} = []$.

Шаг 3. Далее возьмем элемент $x \in semigroup$ и «умножим» его на $y \in semigroup$, получая новое слово $z = xy$. Далее полученное слово необходимо обработать, используя соотношение $r \in R$. После всех преобразований получим слово $z^{'}$. Добавим полученное слово в список $elements_{new}$.

Шаг 4. Инициализировать список $semigroup_{check} = semigroup$ (т.е. делается копия списка $semigroup$). Далее добавляем элементы $z^{'} \in elements_{new}$ в список $semigroup$, если их еще нет в списке $semigroup$.

Шаг 5. Если после после шага 4 переменная $semigroup = semigroup_{check}$, то завершить алгоритм, иначе вернуться к шагу 2.

Оценка сложности алгоритма примерно равна $O((n-1) \cdot n^n \cdot m)$, так как трудно оценить из-за наличия бесконеного цикла в реализации данного алгоритма.

3.7 Коды программ, реализующей рассмотренные алгоритмы

```
import numpy as np

order = []
component = []
```

```
def check_associative(set_list, a):
  n = len(set_list)
  for i in range(n):
    for j in range(n):
      for k in range(n):
        if a[i][set_list.index(str(a[j][k]))] != \
          a[set_list.index(str(a[i][j]))][k]:
          return False
  return True

def dfs(graph, used, v):
    used[v] = True
    for i in range(len(graph)):
        to = i
        if (not(used[to]) and graph[v][to]):
            dfs(graph, used, to)
    order.append(v)


def dfs2(graph, used, v):
    used[v] = True
    component.append(v)
    for i in range(len(graph)):
        to = i
        if (not(used[to]) and graph[v][to]):
            dfs2(graph, used, to)


def find_ideals(st, cayley_table):
    right_ideal = [set(i) for i in cayley_table]
    left_ideal = []
    for i in range(len(cayley_table)):
        s = set()
        for j in range(len(cayley_table)):
            s.add(cayley_table[j][i])
        left_ideal.append(s)
    full_ideal = []
    for i in range(len(cayley_table)):
        full_ideal.append(right_ideal[i].union(left_ideal[i]))
        print(f'Right ideal {st[i]} : {right_ideal[i]}')
        print(f'Left ideal {st[i]} : {left_ideal[i]}')
```

```
        print(f'Full ideal {st[i]} : {full_ideal[i]}')
    return right_ideal, left_ideal, full_ideal


def build_green_relation(right_ideal, left_ideal):
    mr = []
    for i in range(len(right_ideal)):
        m = []
        for j in range(len(right_ideal)):
            if right_ideal[i] == right_ideal[j]:
                m.append(1)
            else:
                m.append(0)
        mr.append(m)
    ml = []
    for i in range(len(left_ideal)):
        m = []
        for j in range(len(left_ideal)):
            m.append(int(left_ideal[i] == left_ideal[j]))
        ml.append(m)
    mr = np.array(mr)
    ml = np.array(ml)
    green_realtion = mr + ml
    for i in range(len(green_realtion)):
        for j in range(len(green_realtion)):
            if green_realtion[i][j] > 1:
                green_realtion[i][j] = 1
    print("Green relation: ")
    print(green_realtion)
    return green_realtion


def gen_cayley_table(set_list, presentation):
    semigroup = set_list.copy()
    while True:
        new_elements = []
        for el1 in semigroup:
            for el2 in semigroup:
                new_word = el1 + el2
                while True:
                    tmp = str(new_word)
```

```
                    for key, val in presentation.items():
                        if key in new_word:
                            new_word = new_word.replace(key, val)
                    if tmp == new_word:
                        break
                new_elements.append(new_word)
        check_semgr = set(semigroup.copy())
        for el in new_elements:
            if el not in semigroup:
                semigroup.append(el)
        if check_semgr == set(semigroup):
            break

    cayley_table = []
    for i in range(len(semigroup)):
        t = []
        for j in range(len(semigroup)):
            new_word = semigroup[i] + semigroup[j]
            while True:
                tmp = str(new_word)
                for key, val in presentation.items():
                    if key in new_word:
                        new_word = new_word.replace(key, val)
                if tmp == new_word:
                    break
            t.append(new_word)
        cayley_table.append(t)

    print("Semigroup:")
    print(*semigroup)
    print("Cayley table: ")
    print(cayley_table)
    return semigroup, cayley_table


def task1():
    print("Enter your set")
    st = input().split()
    print('Enter Cayley table')
    print("  ", *st)
    cayley_table = [input(f'{i}  ').split() for i in st]
```

```
    if check_associative(st, cayley_table):
        find_ideals(st, cayley_table)
    else:
        print('Your Cayley table is not associative!')


def task2():
    print("Enter your set")
    st = input().split()
    print('Enter Cayley table')
    print("  ", *st)
    cayley_table = [input(f'{i}  ').split() for i in st]
    if check_associative(st, cayley_table):
        right_ideal, left_ideal, _ = find_ideals(st, cayley_table)
        build_green_relation(right_ideal, left_ideal)
    else:
        print('Your Cayley table is not associative!')


def task3():
    print('Enter elements of set:')
    s = input()
    set_list = [i for i in s.split(' ')]
    print('Number of elements in presentation:')
    k = int(input())
    presentation = {}
    for i in range(k):
        print(f'Enter element №{i + 1}')
        key = input()
        print(f'Enter equivalent of element №{i + 1}')
        val = input()
        presentation[key] = val
    semigroup, cayley_table = gen_cayley_table(set_list, presentation)
    if check_associative(semigroup, cayley_table):
        right_ideal, left_ideal, _ = find_ideals(semigroup, cayley_table)
        green_relation = build_green_relation(right_ideal, left_ideal)
        used = [False for _ in range(len(green_relation))]
        for i in range(len(green_relation)):
            if (not(used[i])):
                dfs(green_relation, used, i)
        used = [False for _ in range(len(green_relation))]
```

```
        egg_box = [[1]]
        for i in range(len(green_relation)):
            v = order[len(green_relation) - 1 - i]
            if (not(used[v])):
                dfs2(green_relation.T, used, v)
                c = []
                for el in component:
                    c.append(semigroup[el])
                egg_box.append(c.copy())
                component.clear()
        print(egg_box)
    else:
        print('Your Cayley table is not associative!')


def task4():
    print("Enter your set")
    st = input().split()
    print('Enter Cayley table')
    print("  ", *st)
    cayley_table = [input(f'{i}  ').split() for i in st]
    right_ideal, left_ideal, _ = find_ideals(st, cayley_table)
    green_relation = build_green_relation(right_ideal, left_ideal)
    used = [False for _ in range(len(green_relation))]
    for i in range(len(green_relation)):
        if (not(used[i])):
            dfs(green_relation, used, i)
    used = [False for _ in range(len(green_relation))]
    egg_box = []
    for i in range(len(green_relation)):
        v = order[len(green_relation) - 1 - i]
        if (not(used[v])):
            dfs2(green_relation.T, used, v)
            egg_box.append(component.copy())
            component.clear()
    print(egg_box)

if __name__ == "__main__":
    print("What are you want? (1 - Find ideals, 2 - Build green relation, 3 - Build g
    f = int(input())
    if f == 1:
```

```
    task1()
elif f == 2:
    task2()
elif f == 3:
    task3()
else:
    print("Something going wrong! Enter a number from 1 to 3")
```

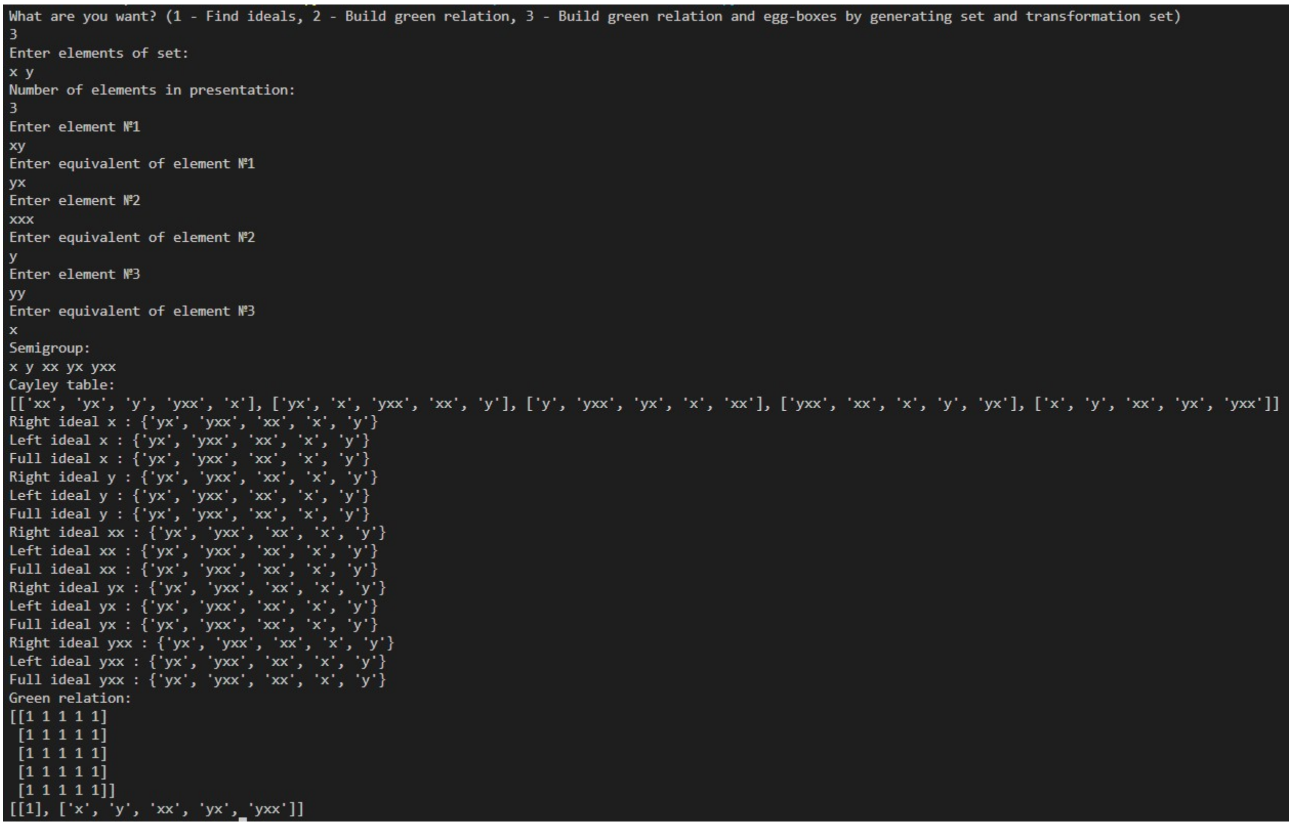
3.8 Результаты тестирования программ

```
What are you want? (1 - Find ideals, 2 - Build green relation, 3 - Build green relation and egg-boxes by generating set and transformation set)
1
Enter your set
a b c d
Enter Cayley table
   a b c d
a  a b c d
b  b c d a
c  c d a b
d  d a b c
Right ideal a : {'c', 'd', 'b', 'a'}
Left ideal a : {'c', 'd', 'b', 'a'}
Full ideal a : {'a', 'c', 'd', 'b'}
Right ideal b : {'c', 'd', 'b', 'a'}
Left ideal b : {'c', 'd', 'b', 'a'}
Full ideal b : {'a', 'c', 'd', 'b'}
Right ideal c : {'c', 'b', 'd', 'a'}
Left ideal c : {'c', 'b', 'd', 'a'}
Full ideal c : {'a', 'c', 'b', 'd'}
Right ideal d : {'b', 'c', 'd', 'a'}
Left ideal d : {'b', 'c', 'd', 'a'}
Full ideal d : {'a', 'c', 'b', 'd'}
```

Рисунок 2 – Тест алгоритма поиска идеалов

```
What are you want? (1 - Find ideals, 2 - Build green relation, 3 - Build green relation and egg-boxes by generating set and transformation set)
2
Enter your set
a b c d
Enter Cayley table
   a b c d
a  a b c d
b  b c d a
c  c d a b
d  d a b c
Right ideal a : {'b', 'd', 'a', 'c'}
Left ideal a : {'b', 'd', 'a', 'c'}
Full ideal a : {'a', 'd', 'b', 'c'}
Right ideal b : {'b', 'd', 'a', 'c'}
Left ideal b : {'b', 'd', 'a', 'c'}
Full ideal b : {'a', 'd', 'b', 'c'}
Right ideal c : {'d', 'b', 'a', 'c'}
Left ideal c : {'d', 'b', 'a', 'c'}
Full ideal c : {'a', 'd', 'b', 'c'}
Right ideal d : {'d', 'c', 'a', 'b'}
Left ideal d : {'d', 'c', 'a', 'b'}
Full ideal d : {'a', 'd', 'b', 'c'}
Green relation:
[[1 1 1 1]
 [1 1 1 1]
 [1 1 1 1]
 [1 1 1 1]]
```

Рисунок 3 – Тест алгоритма построения отношения Грина



```
What are you want? (1 - Find ideals, 2 - Build green relation, 3 - Build green relation and egg-boxes by generating set and transformation set)
3
Enter elements of set:
x y
Number of elements in presentation:
3
Enter element №1
xy
Enter equivalent of element №1
yx
Enter element №2
xxx
Enter equivalent of element №2
y
Enter element №3
yy
Enter equivalent of element №3
x
Semigroup:
x y xx yx yxx
Cayley table:
[['xx', 'yx', 'y', 'yxx', 'x'], ['yx', 'x', 'yxx', 'xx', 'y'], ['y', 'yxx', 'yx', 'x', 'xx'], ['yxx', 'xx', 'x', 'y', 'yx'], ['x', 'y', 'xx', 'yx', 'yxx']]
Right ideal x : {'yx', 'yxx', 'xx', 'x', 'y'}
Left ideal x : {'yx', 'yxx', 'xx', 'x', 'y'}
Full ideal x : {'yx', 'yxx', 'xx', 'x', 'y'}
Right ideal y : {'yx', 'yxx', 'xx', 'x', 'y'}
Left ideal y : {'yx', 'yxx', 'xx', 'x', 'y'}
Full ideal y : {'yx', 'yxx', 'xx', 'x', 'y'}
Right ideal xx : {'yx', 'yxx', 'xx', 'x', 'y'}
Left ideal xx : {'yx', 'yxx', 'xx', 'x', 'y'}
Full ideal xx : {'yx', 'yxx', 'xx', 'x', 'y'}
Right ideal yx : {'yx', 'yxx', 'xx', 'x', 'y'}
Left ideal yx : {'yx', 'yxx', 'xx', 'x', 'y'}
Full ideal yx : {'yx', 'yxx', 'xx', 'x', 'y'}
Right ideal yxx : {'yx', 'yxx', 'xx', 'x', 'y'}
Left ideal yxx : {'yx', 'yxx', 'xx', 'x', 'y'}
Full ideal yxx : {'yx', 'yxx', 'xx', 'x', 'y'}
Green relation:
[[1 1 1 1 1]
 [1 1 1 1 1]
 [1 1 1 1 1]
 [1 1 1 1 1]
 [1 1 1 1 1]]
[[1], ['x', 'y', 'xx', 'yx', 'yxx']]
```

Рисунок 4 – Тест алгоритма построения отношения Грина и egg-box-ов по порождающему множеству и определяющим соотношениям

3.9 Ответы на задачи

Задание 1. Найдите подполугруппу $\langle x \rangle$, правый $(x]$, левый $[x)$ и двусторонний $[x]$ идеалы полугруппы S, порожденные элементом x, и определите порядок элемента x для каждого элемента полугруппы, на которой бинарная операция задана следующей таблицей Кэли:

·	a	b	c	d
a	a	b	c	d
b	b	c	d	a
c	c	d	a	b
d	d	a	b	c

1. Нахождение подполугруппы:

   Подполугруппа строится по порождающему ее множеству. Допустим у нас есть подмножество $X \subset S$, где $X = \{a\}$. Тогда построим подполгруппу: Элемент a подмножества X определен в первой строчке таблицы Кэли. Поэтому мы должны пройти по элементам, находящихся в первой строке. Если еще такого элемента нет в подмножестве X, то он добавляется в данное подмножество. Т.е. пройдя по первой строке в данном случае получается подмножество $X = \{a, b, c, d\}$. Далее необходимо пройтись по строчкам, в котором определены новые элементы подмножества X (т.е. b, c, d). Рассмотрим элемент b и, пройдясь по второй строчке, видно, что новых элементов в подмножество X не добавилось. Значит, мы получили подполугруппу $\langle X \rangle = \{a, b, c, d\}$.

Аналогично строится подполугруппа для b, c, d полугруппы S:

Пусть $X \subset S$, где $X = \{b\}$. Тогда $\langle X \rangle = \{a, b, c, d\}$.

Пусть $X \subset S$, где $X = \{c\}$. Тогда $\langle X \rangle = \{a, b, c, d\}$.

Пусть $X \subset S$, где $X = \{d\}$. Тогда $\langle X \rangle = \{a, b, c, d\}$.

2. Нахождение идеалов:

Используя таблицу Кэли построим правые идеалы:

$$(a] = \{a, b, c, d\}$$
$$(b] = \{a, b, c, d\}$$
$$(c] = \{a, b, c, d\}$$
$$(d] = \{a, b, c, d\}$$

Теперь построим левые идеалы:

$$[a) = \{a, b, c, d\}$$
$$[b) = \{a, b, c, d\}$$
$$[c) = \{a, b, c, d\}$$
$$[d) = \{b, b, c, d\}$$

Также построим двусторонние идеалы:

$$[a] = \{a, b, c, d\}$$
$$[b] = \{a, b, c, d\}$$
$$[c] = \{a, b, c, d\}$$
$$[d] = \{a, b, c, d\}$$

Задание 2.

Найдем отношения Грина для полугруппы $S = \{a, b, c, d\}$ из задания 1:

Заполним матрицу $\mathfrak{R}$, элементы которой будут определяться следующим образом: Возьмем правый идеал $(a]$ и рассмотрим относительно него остальные правые идеалы. Если, например, $(a] = (b]$, то на месте пересечения элементов a и b в матрице $\mathfrak{R}$ будет стоять 1, в противном случае будет стоять 0.

Тогда матрица будет выглядеть следующим образом:

$$\mathfrak{R} = \begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \end{pmatrix}$$

Аналогично построим матрицу $\mathfrak{L}$ по левым идеалам:

$$\mathfrak{L} = \begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \end{pmatrix}$$

Тогда отношение Грина будет представлено матрицей $\mathfrak{D} = \mathfrak{R} \oplus \mathfrak{L}$:

$$\mathfrak{D} = \begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \end{pmatrix}$$

Задание 3.

Найдите полугруппу S по следующему ее копредставлению:

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

Выделим полную систему представителей классов конгруэнции ϵ, которая определяется соотношениями данного копредставления. Для этого последовательно рассмотрим слова фиксированной длины и выделим те, которые не будут эквивалентны между собой относительно конгруэнции ϵ.

Рассмотрим слова длины 1: x, y — эти слова не эквивалентны между собой относительно конгруэнции ϵ.

Рассмотрим слова длины 2, которые получаются из слов длины 1 путем последовательного умножения их справа на буквы x и y: $x^2, xy, yx = xy, y^2 = x$ – из этих слов только слова y^2, yx, не эквивалентны относительно конгруэнции ϵ другим ранее выделенным словам.

Теперь рассмотрим слова длины 3, которые получаются из выделенных слов длины 2 путем последовательного умножения их справа на буквы x и y:

$x^3 = x$, x^2y, xyx, $xy^2 = x^2$ — из этих слов только слово y^2x не эквивалентно относительно конгруэнции ε другим ранее выделенным словам.

Наконец рассмотрим слова длины 4, которые получаются из выделенного слова длины 3 путем последовательного умножения его справа на буквы x и y: $x^3y = xy$, $x^2y^2 = x^3 = x$ – все эти слова эквивалентны относительно конгруэнции ε ранее выделенным словам.

Значит, $S = \{xyyxy^2y^2x\}$ — полная система представителей классов конгруэнции ε. Операция умножения $\cdot$ таких слов определяется с точностью до конгруэнции ε по следующей таблице Кэли:

$\cdot$	x	y	yx	y^2	y^2x
x	y	yx	yy	y^2x	x
y	yx	y^2	y^2x	x	y
yx	y^2	y^2x	x	y	yx
y^2	y^2x	x	y	yx	y^2
y^2x	x	y	yx	y^2	y^2x

Соответственно правые идеалы будут иметь следующие значения:

$$(x] = \{y^2x, yx, x, y^2, y\}$$
$$(y] = \{y^2x, yx, x, y^2, y\}$$
$$(y^2] = \{y^2x, yx, x, y^2, y\}$$
$$(yx] = \{y^2x, yx, x, y^2, y\}$$
$$(y^2x] = \{y^2x, yx, x, y^2, y\}$$

Соответственно левые идеалы:

$$[x) = \{y^2x, yx, x, y^2, y\}$$
$$[y) = \{y^2x, yx, x, y^2, y\}$$
$$[y^2) = \{y^2x, yx, x, y^2, y\}$$
$$[yx) = \{y^2x, yx, x, y^2, y\}$$
$$[y^2x) = \{y^2x, yx, x, y^2, y\}$$

Тогда отношение Грина:

$$\mathfrak{D} = \mathfrak{R} \vee \mathfrak{L} = \begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \end{pmatrix} + \begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \end{pmatrix} = \begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \end{pmatrix}$$

ЗАКЛЮЧЕНИЕ

В рамках данной лабораторной работы были рассмотренны теоритические идеалов полугруппы, понятие и свойства отношений Грина на полугруппах и понятие egg-box. На основе этой теоретической части была смоделирована программа на языке Python с использованием средств библиотеки Numpy, которая способна построить найти идеалы полугрупп по по таблице Кэли, построить таблицу Грина, построить полугруппу по порождающему множеству и определяющим соотношениям и построить egg-box-ы.