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

ИДЕАЛЫ ПОЛУГРУПП

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 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, ·) с одной ассоциативной бинарной операцией ·, т.е. выполняется

$$(x \cdot y) \cdot z = x \cdot (y \cdot z)$$

для любых $x, y, z \in S$.

Определение 2. Полугруппа с единичным элементом называется моноидом. Другими словами, моноид $M = (M, \cdot, 1)$ – это алгебра с ассоциативной бинарной операцией и выделенным единичным элементом 1. При этом полугруппа $(M, \cdot)$ называется полугруппой моноида $M = (M, \cdot, 1)$ и гомоморфизмом моноидов называется гомоморфизм их полугрупп, сохраняющий выделенные единичные элементы. Для любой полугруппы $S = (S, \cdot)$ канонически определяется моноид $M(S)$ по следующему правилу: $M(S) = S \subset \{1\}$ для некоторого элемента $1 \notin S$ и умножение в $M(S)$ на новый элемент 1 определяется по формуле: $x \cdot 1 = 1 \cdot x = x$ (а умножение элементов из S совпадает с умножением этих элементов в полугруппе S). Полугруппа моноида $M(S)$ обозначается символом S^1 и называется полугруппой с внешне присоединенной единицей.

Определение 3. 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 $\mathfrak{J} = kerf$, $\mathfrak{R} = kerf_r$, $\mathfrak{L} = kerf_l$ 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 следующей «egg-box»-диаграммы, клетки которой являются классами эквивалентности $\mathfrak{H}$, лежащими в $\mathfrak{D}$.

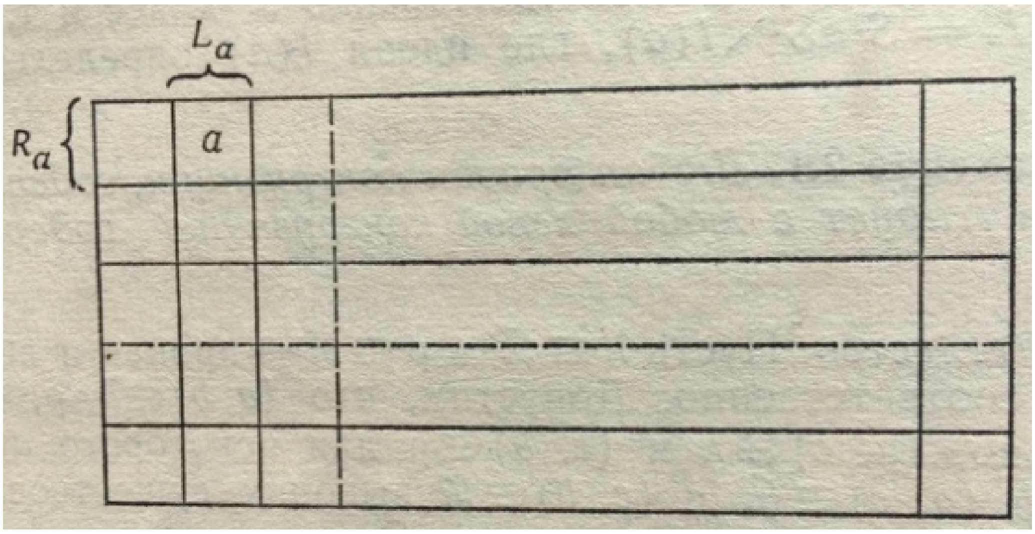

Рисунок 1 – «egg-box»-диаграмма

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 $0 \le i < n$. Если элемент $a[index][i]$ $(0 \le i < n)$ еще не содержится в множестве Id_R, то он добавляется в это множество. В противном случае переходим к элементу $a[index][i+1]$.

Шаг 3. Вернуть множество Id_R в качестве выхода функции.

Оценка сложности алгоритма равна $O(n)$.

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 $0 \le i < n$. Если элемент $a[i][index]$ $(0 \le i < n)$ еще не содержится в множестве Id_L, то он добавляется в это множество. В противном случае переходим к элементу $a[i+1][index]$.

Шаг 3. Вернуть множество Id_L в качестве выхода функции.

Оценка сложности алгоритма равна $O(n)$.

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. Инициализировать множество $Id = \{\}$. 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, то он добавляется в это множество. В противном случае переходим к элементу $a[index][i+1]$. Затем 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, то он добавляется в это множество. В противном случае переходим к элементу $a[i+1][index]$.

Шаг 3. Вернуть множество Id в качестве выхода функции.

Оценка сложности алгоритма равна $O(n)$.

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 ссписок $right\_ideals$, элементы которого будут правые идеалы $Id_{R_0}, \ldots, Id_{R_{n-1}}$ порожденные элементами $x_i \in S$, где $0 \leq i < n$. Аналогично, используя алгоритм 2, построим список $left\_ideals$, элементы которого будут правые идеалы $Id_{L_0}, \ldots, Id_{L_{n-1}}$ 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 $r[i][j] = 1$, в противном случае – $r[i][j] = 0$. После того, как $j = n$, необходимо присвоить $i = i + 1$ и осуществить проход по элементам $Id_{R_0}, \ldots, Id_{R_{n-1}}$ 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 выхода функции 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 $D = (d_{ij})$.

Шаг 2. Необходимо в матрице D найти все компоненты связности. Они находятся с помощью алгоритма 6. 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 – Нахождение компонент связности

Вход: Матрица $D = (d_{ij})$ отношения Грина размерности $n \times n$.

Выход: Список egg_box_list, который содержит элементы $egg-box$-картины.

Шаг 1. Инициализировать списки $order = []$, $component = []$ и $visited = []$. Присвоить элементам $visited[i]$ $(0 \leq i < n)$ значения $false$.

Шаг 2. Выполнить обход в глубину начиная с элемента $x_i \in S$, если $visited[i] = False$ $(0 \leq i < n)$, и подавая на вход список $order$, матрицу D. Когда $\forall visited[i] = True$, обход в глубину закончится и будет получен список $order$, элементы которого являются временем выхода из текущего элемента x_i.

Шаг 3. Далее получим матрицу D^T, путем транспонирования матрицы D. Присвоить элементам $visited[i]$ $(0 \leq i < n)$ значения $false$. Далее выполним обход в глубину уже начиная с элемента $order[n-1-i]$, если $visited[order[n-1-i]] = False$ $(0 \leq i < n)$, и подавая на вход список $component$, матрицу D^T. После очередного $i-$ $(0 \leq i < n)$ обхода в глубину добавим список $component$ в список egg_box_list. Очистить список $component = []$ и перейти к $i+1$ элементу.

Шаг 4. Вернуть список egg_box_list в качестве выхода функции.

Оценка сложности алгоритма равна O(n).

3.7 Алгоритм 7 – 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.

3.8 Коды программ, реализующей рассмотренные алгоритмы

```
import numpy as np


def print_set(s):
    print('{', end=' ')
    k = 1
    n = len(s)
    for el in s:
        if k == n:
            print(el, '}')
        else:
            print(str(el) + ',', end=' ')
        k += 1
```

```
# Проверка операции на ассоциативность
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 get_right_ideal(x, set_list, c_tbl):
    right_ideal = set()
    indx = set_list.index(x)
    for el in c_tbl[indx]:
        right_ideal.add(el)
    return right_ideal


def get_left_ideal(x, set_list, c_tbl):
  left_ideal = set()
  indx = set_list.index(x)
  for i in range(len(c_tbl)):
    left_ideal.add(c_tbl[i][indx])
  return left_ideal


def dfs(gr, visited, v, order):
    visited[v] = True
    for i in range(len(gr)):
        u = i
        if (not(visited[u]) and gr[v][u]):
            dfs(gr, visited, u, order)
    order.append(v)


def dfs1(t_gr, visited, v, component):
    visited[v] = True
```

```
        component.append(v)
        for i in range(len(t_gr)):
            u = i
            if (not(visited[u]) and t_gr[v][u]):
                dfs1(t_gr, visited, u, component)


def print_egg_boxes(semigroup, egg_box):
    print('Your egg-box-diagram:')
    print('{* 1 }')
    for box in egg_box:
        print('{*', end=' ')
        for i in range(len(box)):
            print(semigroup[box[i]], end=' ')
        print('}')


def get_egg_boxes(semigroup, d):
    n = len(d)
    order = []
    component = []
    visited = [False for _ in range(n)]
    for i in range(n):
        if (not(visited[i])):
            dfs(d, visited, i, order)
    visited = [False for _ in range(n)]
    egg_box = []
    for i in range(n):
        v = order[n - 1 - i]
        if (not(visited[v])):
            dfs1(d.T, visited, v, component)
            egg_box.append(component.copy())
            component.clear()
    order.clear()
    return egg_box


def create_Grin_relation(semigroup, right_ideals_dict, left_ideals_dict):
    r = []
    l = []
    for el1 in semigroup:
```

```
        tmp_a = []
        tmp_b = []
        for el2 in semigroup:
            if right_ideals_dict[el1] == right_ideals_dict[el2]:
                tmp_a.append(1)
            else:
                tmp_a.append(0)
            if left_ideals_dict[el1] == left_ideals_dict[el2]:
                tmp_b.append(1)
            else:
                tmp_b.append(0)
        r.append(tmp_a)
        l.append(tmp_b)
    n = len(semigroup)
    d = np.zeros((n, n))
    for i in range(n):
        for j in range(n):
            if r[i][j] == l[i][j] == 1:
                d[i][j] = 1
            else:
                d[i][j] = r[i][j] + l[i][j]

    print('Your Grin\'s relation:')
    for el in d:
        print(el)

    print('Would you like to get egg-box-diagram? (1 - "yes")')
    bl = input()
    if bl == '1':
        egg_box_list = get_egg_boxes(semigroup, d)

    print_egg_boxes(semigroup, egg_box_list)

def get_and_set_ideals(semigroup, c_tbl):
    right_ideals_dict = {}
    left_ideals_dict = {}
    for el in semigroup:
        print('-----------------------------------------')
        print(f'Right ideal ({el}]:', end=' ')
        right_ideals_dict[el] = get_right_ideal(el, semigroup, c_tbl)
        tmp_set = list(right_ideals_dict[el])
```

```
        tmp_set.sort()
        print_set(tmp_set)
        print(f'Left ideal [{el}):', end=' ')
        left_ideals_dict[el] = get_left_ideal(el, semigroup, c_tbl)
        tmp_set = list(left_ideals_dict[el])
        tmp_set.sort()
        print_set(tmp_set)
        print(f'Ideal [{el}]:', end=' ')
        tmp_set = list(left_ideals_dict[el].union(right_ideals_dict[el]))
        tmp_set.sort()
        print_set(tmp_set)
    print('-----------------------------------------')
    return right_ideals_dict, left_ideals_dict


def create_ideals():
    print('Enter set values:')
    s = input()
    semigroup = [i for i in s.split(' ')]
    n = len(semigroup)
    print('Enter Cayley table values:')
    c_tbl = []
    for i in range(n):
      c_tbl.append([j for j in input().split()])

    if check_associative(semigroup, c_tbl) == False:
        print('Cayley table isn\'t associative!')
        return choose_mode()

    right_ideals_dict, left_ideals_dict = get_and_set_ideals(semigroup, c_tbl)

    print('Would you to create Grin\'s relation? (1 - "yes")')
    bl = input()
    if bl == '1':
        create_Grin_relation(semigroup, right_ideals_dict, left_ideals_dict)
    return choose_mode()


def create_table(semigroup, n, presentation):
    a = []
    for i in range(n):
```

```
        tmp_a = []
        for j in range(n):
            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
            tmp_a.append(new_word)
        a.append(tmp_a)
    return a


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

```python
                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

    print("Your semigroup:")
    print(semigroup)
    tbl = create_table(semigroup, len(semigroup), presentation)
    print('Cayley table:')
    for line in tbl:
        print(line)

    if check_associative(semigroup, tbl) == False:
        print('Cayley table isn\'t associative!')
        return choose_mode()

    right_ideals_dict, left_ideals_dict = get_and_set_ideals(semigroup, tbl)

    print('Would you to create Grin\'s relation? (1 - "yes")')
    bl = input()
    if bl == '1':
        create_Grin_relation(semigroup, right_ideals_dict, left_ideals_dict)

    choose_mode()


# Главное меню
def choose_mode():
    print('Choose mode:')
    print('Press 1 to create Grin\'s relations')
    print('Press 2 to create Grin\'s relations via subset')
    print('Press 3 to exit')
    bl = input()
    if bl == '1':
        create_ideals()
    elif bl == '2':
        create_semigroup_via_subset()
    elif bl == '3':
```

```
        return
    else:
        print('Incorrect output')
        return choose_mode()


choose_mode()
```

3.9 Результаты тестирования программ

На рисунке 1 показана работа алгоритма построения идеалов по следующей таблице Кэли:

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

Из рисунка видна, что ассоциативность текущей таблицы не выполняется.

```
Choose mode:
Press 1 to create Grin's relations
Press 2 to create Grin's relations via subset
Press 3 to exit
1
Enter set values:
a b c d
Enter Cayley table values:
a b c d
b d a c
c a d b
d a b c
Cayley table isn't associative!
Choose mode:
Press 1 to create Grin's relations
Press 2 to create Grin's relations via subset
Press 3 to exit
```

Рисунок 2 – Проверка свойства ассоциативности таблицы

На рисунке 2 показано построение идеалов таблицы Кэли:

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

```
Choose mode:
Press 1 to create Grin's relations
Press 2 to create Grin's relations via subset
Press 3 to exit
1
Enter set values:
a b c d
Enter Cayley table values:
a b a b
a b a b
a b c d
a b c d
-----------------------------------------
Right ideal (a]: { a, b }
Left ideal [a): { a }
Ideal [a]: { a, b }
-----------------------------------------
Right ideal (b]: { a, b }
Left ideal [b): { b }
Ideal [b]: { a, b }
-----------------------------------------
Right ideal (c]: { a, b, c, d }
Left ideal [c): { a, c }
Ideal [c]: { a, b, c, d }
-----------------------------------------
Right ideal (d]: { a, b, c, d }
Left ideal [d): { b, d }
Ideal [d]: { a, b, c, d }
-----------------------------------------
Would you to create Grin's relation? (1 - "yes")
```

Рисунок 3 – Тест алгоритма построения идеалов

На рисунке 3 показано построение отношения Грина, которого представлено в виде матрицы, а также построение «egg-box»-картины.

```
-----------------------------------------
Would you to create Grin's relation? (1 - "yes")
1
Your Grin's relation:
[1. 1. 0. 0.]
[1. 1. 0. 0.]
[0. 0. 1. 1.]
[0. 0. 1. 1.]
Would you like to get egg-box-diagram? (1 - "yes")
1
Your egg-box-diagram:
{* 1 }
{* c d }
{* a b }
Choose mode:
Press 1 to create Grin's relations
Press 2 to create Grin's relations via subset
Press 3 to exit
```

Рисунок 4 – Тест алгоритма построения отношения Грина и «egg-box»-картины

На рисунке 4 показано построение полугруппы по копредставлению.

```
Choose mode:
Press 1 to create Grin's relations
Press 2 to create Grin's relations via subset
Press 3 to exit
2
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
x
Enter element №3
yy
Enter equivalent of element №3
x
Your semigroup:
['x', 'y', 'xx', 'yx', 'yxx']
Cayley table:
['xx', 'yx', 'x', 'yxx', 'yx']
['yx', 'x', 'yxx', 'xx', 'x']
['x', 'yxx', 'xx', 'yx', 'yxx']
['yxx', 'xx', 'yx', 'x', 'xx']
['yx', 'x', 'yxx', 'xx', 'x']
```

Рисунок 5 – Построение полугруппы по копредставлению

На рисунке 5 показано построение идеалов, отношения Грина и «egg-box»-картины.

```
Right ideal (x]: { x, xx, yx, yxx }
Left ideal [x): { x, xx, yx, yxx }
Ideal [x]: { x, xx, yx, yxx }
----------------------------------------
Right ideal (y]: { x, xx, yx, yxx }
Left ideal [y): { x, xx, yx, yxx }
Ideal [y]: { x, xx, yx, yxx }
----------------------------------------
Right ideal (xx]: { x, xx, yx, yxx }
Left ideal [xx): { x, xx, yx, yxx }
Ideal [xx]: { x, xx, yx, yxx }
----------------------------------------
Right ideal (yx]: { x, xx, yx, yxx }
Left ideal [yx): { x, xx, yx, yxx }
Ideal [yx]: { x, xx, yx, yxx }
----------------------------------------
Right ideal (yxx]: { x, xx, yx, yxx }
Left ideal [yxx): { x, xx, yx, yxx }
Ideal [yxx]: { x, xx, yx, yxx }
----------------------------------------
Would you to create Grin's relation? (1 - "yes")
1
Your Grin's 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.]
Would you like to get egg-box-diagram? (1 - "yes")
1
Your egg-box-diagram:
{* 1 }
{* x y xx yx yxx }
Choose mode:
Press 1 to create Grin's relations
Press 2 to create Grin's relations via subset
Press 3 to exit
```

Рисунок 6 – Тест алгоритма по построению отношения Грина по порождающему множеству и определяющим соотношениям

3.10 Решение задач

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

1. Нахождение подполугруппы:
   Сначала необходимо проверить данную таблицу Кэли на ассоциативность. В данном варианте (6 варианте) таблице Кэли не ассоциативна, поэтому я возьму для первого задания таблицу Кэли из 3-й лабораторной работы моего варианта. Т.е. таблицу:

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

   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 $X = \{a, b\}$. 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 $\langle X \rangle = \{a, b\}$.
   Аналогично строится подполугруппа для b, c, d полугруппы S:
   Пусть $X \subset S$, где $X = \{b\}$. Тогда $\langle X \rangle = \{a, b\}$.
   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:

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

   Теперь построим левые идеалы:

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

Также построим двусторонние идеалы:

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

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:

$$\mathfrak{R} = \begin{pmatrix} 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 \end{pmatrix}$$

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

$$\mathfrak{L} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

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

$$\mathfrak{D} = \begin{pmatrix} 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 \end{pmatrix}$$

«Egg-box»-картина будет выглядеть следующим образом:

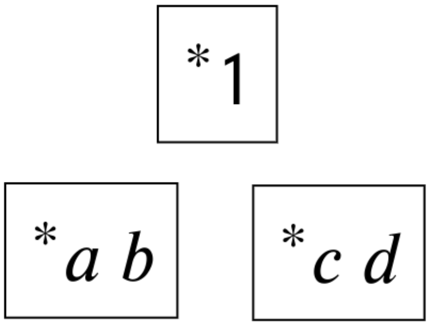

Рисунок 7 – «Egg-box»-картина

«Egg-box»-диаграмма строится по матрице $\mathfrak{D}$. Если элементы находятся в какой-либо одной компоненте связности, то они помещаются в один так называемый «box». В лекции используется алгоритм Тарьяна для нахождения компонент связности. В данной работе используется алгоритм Косарайю и Шарира, оценка сложности которого $O(n+m)$. Стоит отметить, что в отдельный «egg-box» помещается элемент 1, так как осуществляется работа с полугруппой с внешне присоединенной единицей S^1.

Задание 3.

Найдите полугруппу S по следующему ее копредставлению:

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

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

$\cdot$	x	y	x^2	xy	x^2y
x	x^2	xy	x	x^2y	xy
y	xy	x	x^2y	x^2	x
x^2	x	x^2y	x^2	xy	x^2y
xy	x^2y	x^2	xy	x	x^2
x^2y	xy	x	x^2y	x^2	x^2

Соответственно правые идеалы будут иметь следующие значения:

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

Соответственно левые идеалы:

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

Тогда отношение Грина:

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

«Egg-box»-картина будет выглядеть следующим образом:

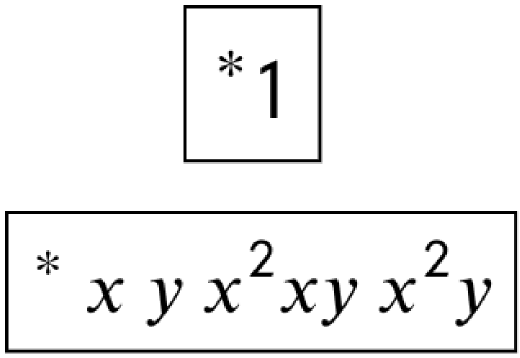

Рисунок 8 – «Egg-box»-картина

ЗАКЛЮЧЕНИЕ

В результате лабораторной работы были рассмотрены теоретические сведения о полугруппах, идеалах, понятия и свойства отношениий Грина на полугруппах, построения «egg-box»-картин. Опираясь на изложенную выше теорию, были разработаны алгоритмы построения идеалов полугруппы по таблице Кэли, вычисления отношений Грина и построения «egg-box»-картины конечной полугруппы. Была произведена оценка сложности каждого из построенных алгоритмов. Была реализована программа, написанная на языке Python с использованием библиотеки Numpy для работы с большими массивами данных.