

Министерство образования и науки Российской Федерации

Севастопольский государственный университет

Кафедра ИС

Отчет

По дисциплине: “Теоретические основы построения компиляторов”

Лабораторная работа №1

“Исследование детерминированного конечного автомата”

Выполнил:

ст.гр. ИС/б-20-1-о

Галенин А. К.

Проверил:

Карлусов В.Ю.

Севастополь

2023

1 ЦЕЛЬ РАБОТЫ

Научиться производить построение детерминированных конечных автоматов (ДКА), допускающих определённые цепочки символов языка. Освоить приёмы описания конечных автоматов (КА) в виде графов, таблиц переходов и регулярных выражений. Научиться выполнять построения ДКА по недетерминированным конечным автоматам (НКА). Научиться проводить построение минимальных детерминированных конечных автоматов (МДКА).

2 ПОСТАВНОВКА ЗАДАЧИ

Вариант 11 – $A \vee B \vee C$

3 ХОД РАБОТЫ

1. Была выполнена разметка регулярного выражения, результат представлен на рисунке 1.

	a	b	{	b	v	a	}	c	{	a	b	}	b	{	c	}	v	c	a	{	b	}
0	1	2		3		4		5		6	7		8		9			0	10	11		12
			2,3,4		2,3,4		2,3,4		5,7				6,7		8,9		8,9			11,12		11,12

Рисунок 1 – Разметка регулярного выражения

2. Была построена таблица переходов ДКА

Таблица 1 – Таблица переходов ДКА

Q	0	1	2	3	4	5	6	7	8	9	10	11	12
a	1		4	4	4	6		6			11		
b		2	3	3	3	8	7	8				12	12
c	10		5	5	5				9	9			

Из таблицы видно, что состояния 2 и 3, 4 и 6, а также 5, 7, и 8, 9, и 11, 12 неразличимы ни по входному сигналу, ни по выходному. Объединив $\{2, 3, 4\} \rightarrow 2$, $\{5, 7\} \rightarrow 3$, $\{8, 9\} \rightarrow 5$, $\{11, 12\} \rightarrow 7$, была получена таблица переходов МДКА.

Таблица 2 – Таблица переходов МДКА

Q	0	1	2	3	4	5	6	7
a	1	99	2	4	99	99	7	99
b	99	2	2	5	3	99	99	7
c	6	99	3	99	99	5	99	99

Был построен соответствующий таблице граф конечного автомата

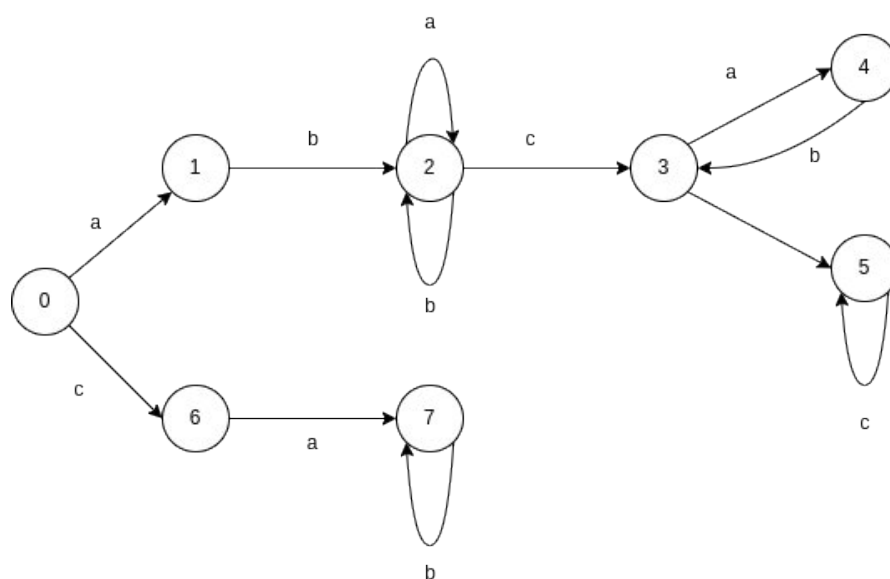


Рисунок 2 – Граф МДКА

3. Код программы

```
import numpy

regex = numpy.array([
    [1, 99, 2, 4, 99, 99, 7, 99],
    [99, 2, 2, 5, 3, 99, 99, 7],
    [6, 99, 3, 99, 99, 5, 99, 99],
])

end_nodes = (3, 4, 5, 7)

test_sequences = open("test_sequence.txt", "r").read().splitlines()
for test_sequence in test_sequences:
    node = 0
    print('\n')
    for symbol in test_sequence:
        match symbol:
            case 'a':
                node = regex[0][node]
            case 'b':
                node = regex[1][node]
            case 'c':
                node = regex[2][node]
            case _:
                node = 999
    if node == 999:
        print('\nНе символ алфавита')
        break
    elif node == 99:
        print('\nНеправильная вершина цепочки')
        break
    else:
        print(f'-{node}-', end="")

    if (node not in end_nodes) and (node != 999) and (node != 99):
        print('\nНеполная цепочка')
    elif (node != 999 and node != 99):
        print(f'\n{test_sequence}')
```

4. Были разработаны тестовые последовательности минимальной длины, допускаемые и отвергаемые КА:

ba – отвергаемая последовательность

ca – допускаемая последовательность

cab – допускаемая последовательность

abab – отвергаемая последовательность

ababc – допускаемая последовательность

ababca – допускаемая последовательность

ab – отвергаемая последовательность

abca – отвергаемая последовательность

```
Неправильная вершина цепочки

-6--7-
ca

-6--7--7-
cab

-1--2--2--2-
Неполная цепочка

-1--2--2--2--3-
ababc

-1--2--2--2--3--4--3-
ababscab

-1--2--3--4-
Неправильная вершина цепочки
```

Рисунок 3 – Результат выполнения программы

ВЫВОДЫ

В ходе выполнения лабораторной работы были изучены принципы построения детерминированных конечных автоматов (ДКА), допускающих определённые цепочки символов языка. Освоены приёмы описания конечных автоматов (КА) в виде графов, таблиц переходов и регулярных выражений. Были изучены принципы построения ДКА по недетерминированным конечным автоматам (НКА). Были изучены методы построения минимальных детерминированных конечных автоматов (МДКА).