

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ
УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

**«БЕЛГОРОДСКИЙ ГОСУДАРСТВЕННЫЙ
ТЕХНОЛОГИЧЕСКИЙ УНИВЕРСИТЕТ им. В. Г. ШУХОВА»
(БГТУ им. В.Г. Шухова)**

Кафедра программного обеспечения вычислительной техники и автоматизированных
систем



Лабораторная работа №3

по дисциплине: Теория автоматов и формальных языков
тема: «Регулярные и конечные распознаватели»

Выполнил: ст. группы ПВ-223

Игнатьев Артур Олегович

Проверил:

Рязанов Юрий Дмитриевич

Белгород 2024 г.

Цель работы: изучить основные способы задания регулярных языков, способы построения, алгоритмы преобразования, анализа и реализации конечных распознавателей.

Вариант 3

1. Язык L_1 в алфавите $\{0,1\}$, представляющий собой множество цепочек, которые состоят из повторяющихся один или более раз фрагментов 01, задан регулярным выражением: $(01)^*01$

Построить детерминированный конечный распознаватель языка L_1 .

2. Язык L_2 в алфавите $\{0,1\}$, представляющий собой множество цепочек, которые состоят из повторяющихся один или более раз фрагментов 010, задан грамматикой:

$S \rightarrow AS$

$S \rightarrow 010$

$A \rightarrow B0$

$B \rightarrow 01$

Построить детерминированный конечный распознаватель языка L_2 .

3. Построить минимальный детерминированный конечный распознаватель языка L_3 в алфавите $\{0,1\}$, представляющий собой множество цепочек, которые состоят либо из повторяющихся один или более раз фрагментов 01, либо из повторяющихся один или более раз фрагментов 010.

4. Написать программу компиляционного типа для реализации минимального детерминированного конечного распознавателя языка L_3 .

5. Написать программу интерпретационного типа для реализации минимального детерминированного конечного распознавателя языка L_3 .

6. Подобрать наборы тестовых данных так, чтобы в процессе тестирования сработал каждый переход конечного распознавателя.

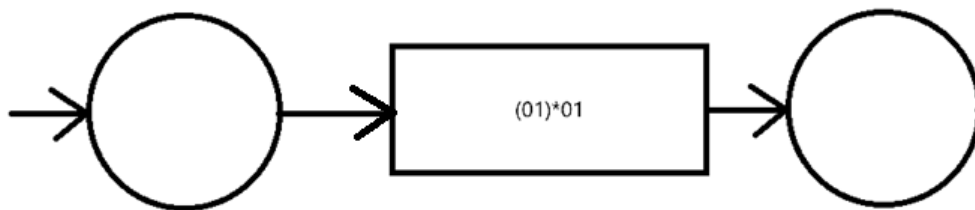
7. Подобрать наборы тестовых данных так, чтобы в процессе тестирования распознаватель закончил обработку цепочек в каждом состоянии конечного распознавателя.

8. Выполнить тестирование программ для реализации минимального детерминированного конечного распознавателя языка L_3 .

Решение заданий

Задание 1. Язык L_1 в алфавите $\{0,1\}$, представляющий собой множество цепочек, которые состоят из повторяющихся один или более раз фрагментов 01 , задан регулярным выражением: $(01)^*01$. Построить детерминированный конечный распознаватель языка L_1 .

1. Конечный распознаватель, допускающий множество цепочек, определяемых регулярным выражением r , представим моделью:

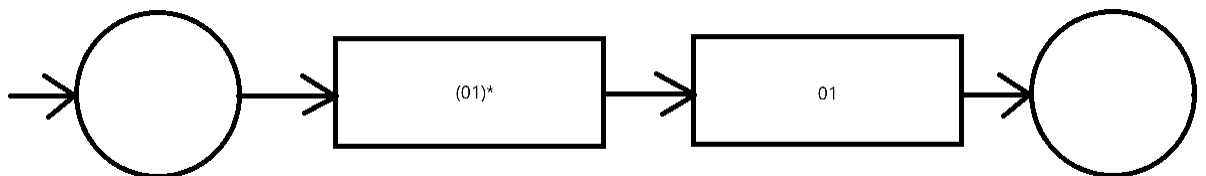


2. Регулярное выражение $r = (01)^*01$ представим как $r = r_1r_2$.

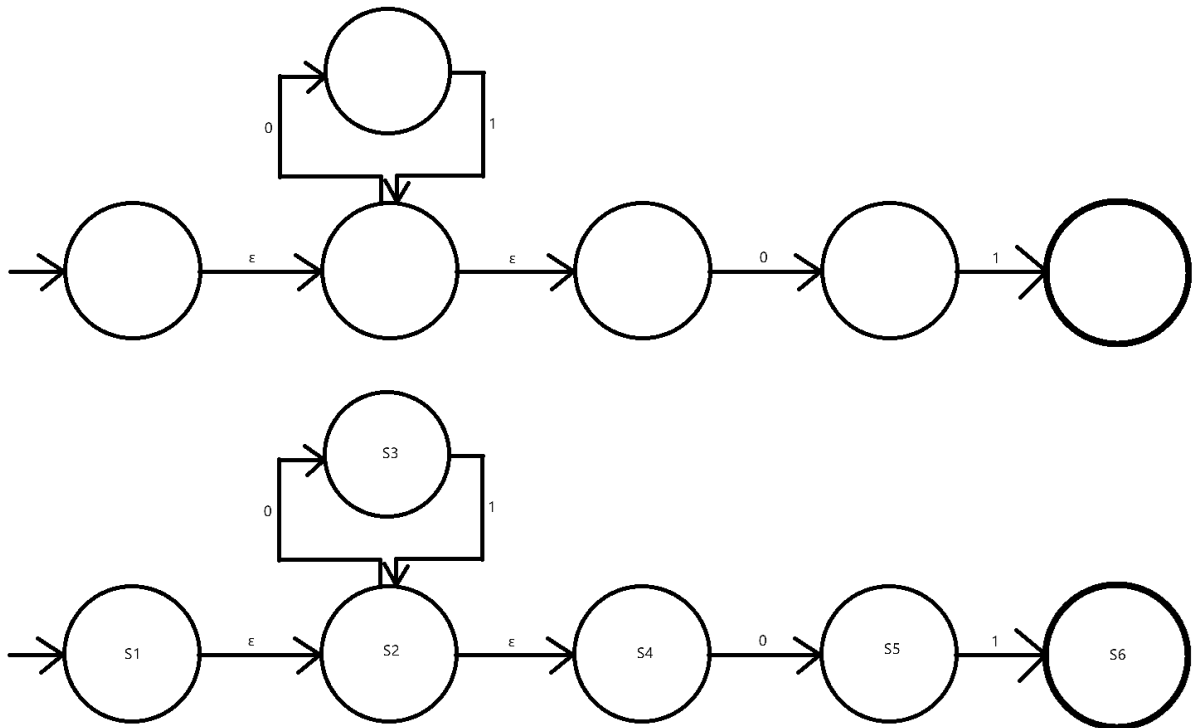
$$r_1 = (01)^*$$

$$r_2 = 01$$

По 5 правилу получаем:



3. Применим 3 и 6 правила к четырехугольнику с маркировкой r1 и 3 правило к четырехугольнику с маркировкой r2. Получим недетерминированный конечный распознаватель с эпсилон переходами:



Преобразуем НКР в ДКР, путем устранения эпсилон-переходов:

	↓					1
	S1	S2	S3	S4	S5	S6
1			S2		S6	
0		S3		S5		
ϵ	S2	S4				

Эпсилон-замыкания:

$$\varepsilon(S1)=\{S1,S2,S4\}$$

$$\varepsilon(S2)=\{S2,S4\}$$

$$\varepsilon(S3)=\{S3\}$$

$$\varepsilon(S4)=\{S4\}$$

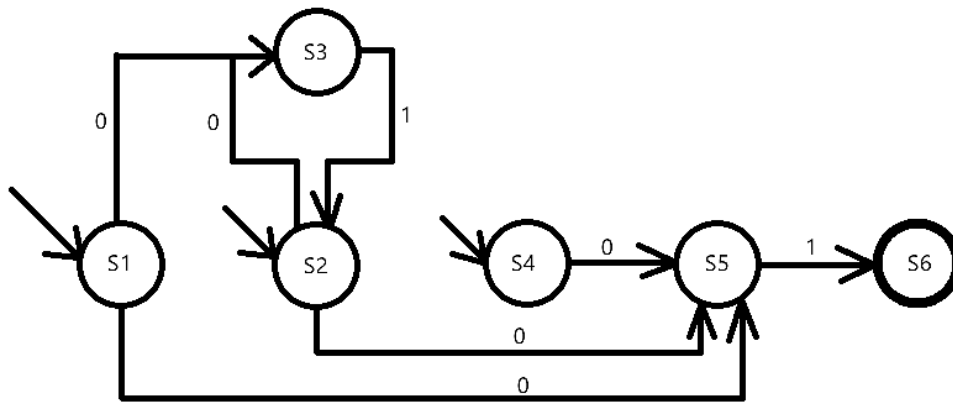
$$\varepsilon(S5)=\{S5\}$$

$$\varepsilon(S6)=\{S6\}$$

	↓	↓		↓		1
	$\varepsilon(S1)$ {S1,S2,S4}	$\varepsilon(S2)$ {S2,S4}	$\varepsilon(S3)$ {S3}	$\varepsilon(S4)$ {S4}	$\varepsilon(S5)$ {S5}	$\varepsilon(S6)$ {S6}
1			$\varepsilon(S2)$		$\varepsilon(S6)$	
0	$\varepsilon(S3), \varepsilon(S5)$	$\varepsilon(S3), \varepsilon(S5)$		$\varepsilon(S5)$		

Устранение эпсилон-переходов:

	↓	↓		↓		1
	S1	S2	S3	S4	S5	S6
1			S2		S6	
0	S3, S5	S3, S5		S5		



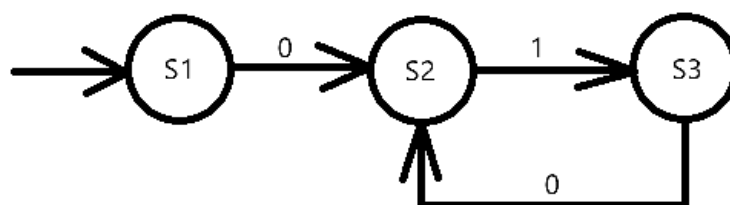
Преобразуем НКР в ДКР:

	{S1, S2, S4}	{S3, S5}	{S2, S6}
1		{S2, S6}	
0	{S3, S5}		{S3, S5}

Обозначим множества состояний как S1, S2, S3. S1 обозначим как начальное состояние, а S3 как допускающее состояние:

	↓		1
	S1	S2	S3
1		S3	
0	S2		S2

Детерминированный конечный распознаватель языка L1:



Задание 2. Язык L_2 в алфавите $\{0,1\}$, представляющий собой множество цепочек, которые состоят из повторяющихся один или более раз фрагментов 010, задан грамматикой:

$$S \rightarrow AS$$

$$S \rightarrow 010$$

$$A \rightarrow B0$$

$$B \rightarrow 01$$

Построить детерминированный конечный распознаватель языка L_2 .

Преобразуем КС-грамматику в автоматную правостороннюю. Лишних символов в грамматике нет. Цепных правил нет. Эпсилон-правил нет. Левой рекурсии нет.

Приведем грамматику к виду, в котором каждое правило будет начинаться с терминала:

$$S \rightarrow 010S$$

$$S \rightarrow 010$$

$$A \rightarrow 010$$

$$B \rightarrow 01$$

Исключаем правила A и B как лишние:

$$S \rightarrow 010S$$

$$S \rightarrow 010$$

Получили правостороннюю грамматику. Теперь используя её построим автоматную правостороннюю:

1)

$$S \rightarrow 010S$$

$$S \rightarrow 010$$

2)

$$S \rightarrow 010S$$

$$S \rightarrow 010E$$

$$E \rightarrow \varepsilon$$

3)

$$S \rightarrow 0N_1$$

$$S \rightarrow 0N_2$$

$$N_1 \rightarrow 10S$$

$$N_2 \rightarrow 10E$$

$$E \rightarrow \varepsilon$$

4)

$$S \rightarrow 0N_1$$

$$S \rightarrow 0N_2$$

$$N_1 \rightarrow 1N_3$$

$$N_2 \rightarrow 1N_4$$

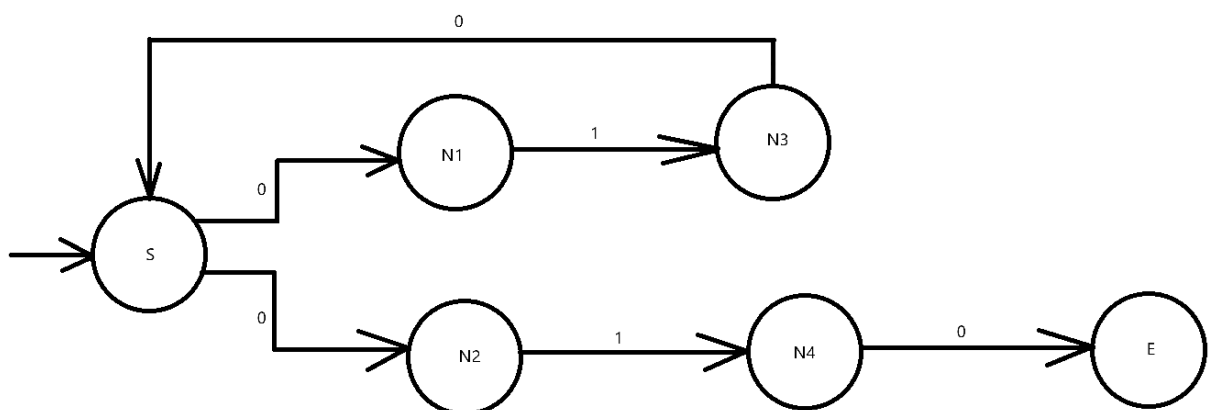
$$N_3 \rightarrow 0S$$

$$N_4 \rightarrow 0E$$

$$E \rightarrow \varepsilon$$

Получили автоматную правостороннюю грамматику. Построим по ней конечный распознаватель.

	↓					1
	S	N1	N2	N3	N4	E
1		N3	N4			
0	N1, N2			S	E	

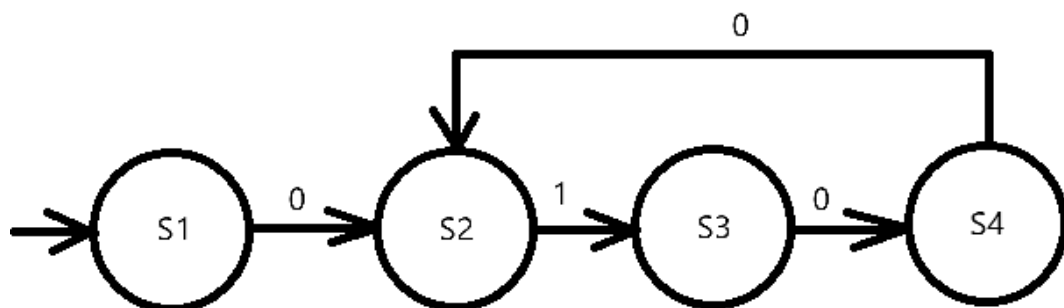


Распознаватель недетерминированный. Необходимо преобразовать его в детерминированный. Эпсилон-переходов нет.

	↓			1
	{S}	{N1, N2}	{N3, N4}	{S, E}
1		{N3, N4}		
0	{N1, N2}		{S, E}	{N1, N2}

Обозначим множества состояний как S1, S2, S3, S4. S1 – начальное состояние, S4 – допускающее.

	↓			1
	S1	S2	S3	S4
1		S3		
0	S2		S4	S2



ДКР языка L2 получен.

Задание 3. Построить минимальный детерминированный конечный распознаватель языка L_3 в алфавите $\{0,1\}$, представляющий собой множество цепочек, которые состоят либо из повторяющихся один или более раз фрагментов 01, либо из повторяющихся один или более раз фрагментов 010.

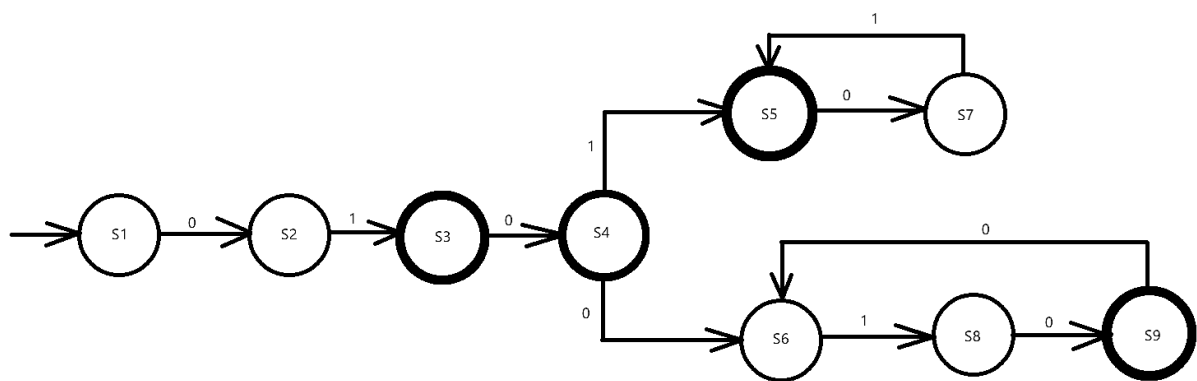
Чтобы получить язык L_3 необходимо объединить ДКР языков L_1 и L_2 и привести результирующий КР к МДКР.

	↓			1	↓		1
	S1	S2	S3	S4	S'1	S'2	S'3
1		S3				S'3	
0	S2		S4	S2	S'2		S'2

Преобразуем недетерминированный распознаватель в детерминированный:

	↓		1	1	1				1
	{S1, S'1}	{S2, S'2}	{S3, S'3}	{S4, S'2}	{S'3}	{S2}	{S'2}	{S3}	{S4}
1		{S3, S'3}		{S'3}		{S3}	{S'3}		
0	{S2, S'2}		{S4, S'2}	{S2}	{S'2}			{S4}	{S2}

	↓		1	1	1				1
	S1	S2	S3	S4	S5	S6	S7	S8	S9
1		S3		S5		S8	S5		
0	S2		S4	S6	S7			S9	S6



Проведем минимизацию ДКР. Недостижимые состояния из начального отсутствуют. Переходим к поиску и исключению эквивалентных состояний.

Таблица переходов в подмножества первого разбиения

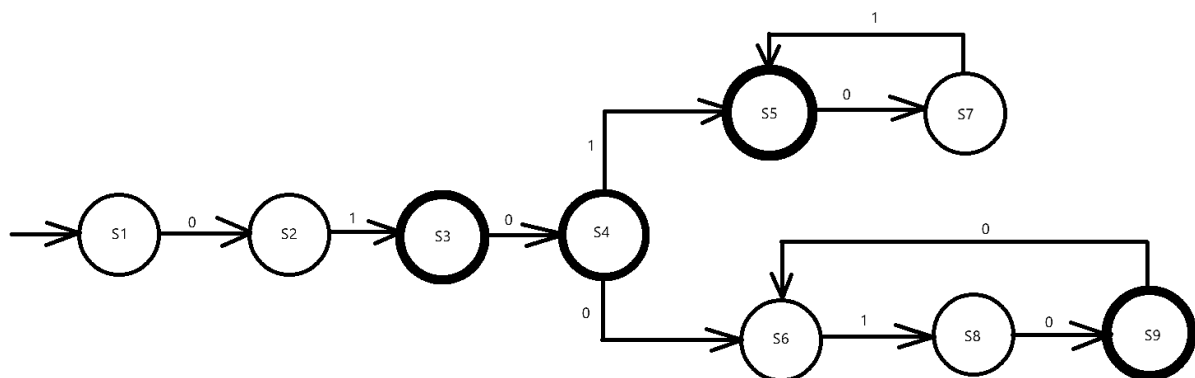
	K1		K2			K3			K5
	S1	S8	S2	S6	S7	S3	S5	S9	S4
1			K3	K1	K3				K3
0	K2	K3				K5	K2	K2	K2

Строим таблицу переходов в классы 1-эквивалентных состояний.

	K1	K2	K3		K4	K5	K6		K5
	S1	S8	S2	S7	S6	S3	S5	S9	S4
1			K5	K6	K2				
0	K3	K6				K5	K3	K4	K4

Из таблицы делаем вывод, что в таблице переходов в классы 2-эквивалентных состояний, каждое состояние будет иметь свой класс. Значит исходный ДКР уже является минимальным детерменированным конечным распознавателем.

L3:



Задание 4. Написать программу компиляционного типа для реализации минимального детерминированного конечного распознавателя языка L3.

Программа выполнена на языке программирования Python.

```
compilation_conditions = {
    1: "Отвергнуть (Начальный символ не 0)",
    2: "Отвергнуть (Второй символ не 1)",
    -3: "Отвергнуть (Третий символ не 0)",
    -4: "Отвергнуть (Некорректный символ начала следующей цепочки)",
    -5: "Отвергнуть (Для цепочки из 01 первый символ не 0)",
    6: "Отвергнуть (Для цепочки из 010 первый символ не 0)",
    7: "Отвергнуть (Для цепочки из 01 второй символ не 1)",
    8: "Отвергнуть (Для цепочки 010 второй символ не 1)",
    -9: "Отвергнуть (Для цепочки 010 третий символ не 0)",
    0: "Принять"
}

session_type_of_compilation_states = [3, 4, 5, 9]

def compilation_type_l3(str_l3):
    str_copy_l3 = str_l3
    s = 1

    while len(str_copy_l3) > 0 and s >= 0:
        current_symbol = str_copy_l3[0]

        if s == 1:
            if current_symbol == "0":
                s = 2
            else:
                break
        elif s == 2:
            if current_symbol == "1":
                s = 3
            else:
                break
        elif s == 3:
            if current_symbol == "0":
                s = 4
            else:
                s = -3
                break
        elif s == 4:
            if current_symbol == "1":
                s = 5
            elif current_symbol == "0":
```

```
        s = 6
    else:
        s = -4
        break
elif s == 5:
    if current_symbol == "0":
        s = 7
    else:
        s = -5
        break
elif s == 6:
    if current_symbol == "1":
        s = 8
    else:
        break
elif s == 7:
    if current_symbol == "1":
        s = 5
    else:
        break
elif s == 8:
    if current_symbol == "0":
        s = 9
    else:
        break
elif s == 9:
    if current_symbol == "0":
        s = 6
    else:
        s = -9
        break

    str_copy_l3 = str_copy_l3[1:]

if s in session_type_of_compilation_states:
    s = 0

print(str_l3, compilation_conditions[s])
return s
```


Задание 5. Написать программу интерпретационного типа для реализации минимального детерминированного конечного распознавателя языка L3.

Программа выполнена на языке программирования Python.

```
conditions_interpretive = {
    0: "Пустая строка",
    -1: "Отвергнуть (Некорректный символ или не соответствует видам цепочек)",
    1: "Принять",
}

transition_table = {
    "1": [-1, 2, -1, 4, -1, 7, 4, -1, -1],
    "0": [1, -1, 3, 5, 6, -1, -1, 8, 5]
}

permitting_states = [2, 3, 4, 8]

def interpretive_type_l3(str_l3):
    str_copy_l3 = str_l3
    s = 0

    while len(str_copy_l3) > 0 and s >= 0:
        current_symbol = str_copy_l3[0]

        if current_symbol in transition_table:
            s = transition_table[current_symbol][s]
        else:
            s = -1
            break

        str_copy_l3 = str_copy_l3[1:]

    if s in permitting_states:
        s = 1
    else:
        s = -1

    print(str_l3, conditions_interpretive[s])
    return s
```

Задание 6. Подобрать наборы тестовых данных так, чтобы в процессе тестирования сработал каждый переход конечного распознавателя.

“010101” – использование всех переходов для цепочек символов “01”

“010010” – использование всех переходов для цепочки символов “010”

“а” – переход в ошибку

Задание 7. Подобрать наборы тестовых данных так, чтобы в процессе тестирования распознаватель закончил обработку цепочек в каждом состоянии конечного распознавателя.

S1 – “” – пустая строка

S2 – “0”

S3 – “01”

S4 – “010”

S5 – “0101”

S6 – “0100”

S7 – “01010”

S8 – “01001”

S9 – “010010”

Задание 8. Выполнить тестирование программ для реализации минимального детерминированного конечного распознавателя языка L3.

Тесты программы компиляционного варианта:

```
if __name__ == '__main__':  
    compilation_type_l3("")  
    compilation_type_l3("0")  
    compilation_type_l3("01")  
    compilation_type_l3("010")  
    compilation_type_l3("0101")  
    compilation_type_l3("0100")  
    compilation_type_l3("01010")  
    compilation_type_l3("01001")  
    compilation_type_l3("010010")
```

Результат выполнения тестирования:

```
PS C:\Users\Mi> & C:/Users/Mi/AppData/Local/Programs/Python/Python312/python.exe  
Отвергнуть (Начальный символ не 0)  
0 Отвергнуть (Второй символ не 1)  
01 Принять  
010 Принять  
0101 Принять  
0100 Отвергнуть (Для цепочки из 010 первый символ не 0)  
01010 Отвергнуть (Для цепочки из 01 второй символ не 1)  
01001 Отвергнуть (Для цепочки 010 второй символ не 1)  
010010 Принять
```

Тесты программы интерпретационного варианта:

```
if __name__ == '__main__':  
    interpretive_type_l3("")  
    interpretive_type_l3("0")  
    interpretive_type_l3("01")  
    interpretive_type_l3("010")  
    interpretive_type_l3("0101")  
    interpretive_type_l3("0100")  
    interpretive_type_l3("01010")  
    interpretive_type_l3("01001")  
    interpretive_type_l3("010010")
```

Результат выполнения тестирования:

```
PS C:\Users\Mi> & C:/Users/Mi/AppData/Local/Programs/Python/Python312/python.exe
Отвергнуть (Некорректный символ или не соответствует видам цепочек
0 Отвергнуть (Некорректный символ или не соответствует видам цепочек
01 Принять
010 Принять
0101 Принять
0100 Отвергнуть (Некорректный символ или не соответствует видам цепочек
01010 Отвергнуть (Некорректный символ или не соответствует видам цепочек
01001 Отвергнуть (Некорректный символ или не соответствует видам цепочек
010010 Принять
```

Вывод: в ходе выполнения лабораторной работы изучили основные способы задания регулярных языков, способы построения, алгоритмы преобразования, анализа и реализации конечных распознавателей.