

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 языки и конечные распознаватели»

Выполнил: ст. группы ПВ-223  
Дмитриев Андрей  
Александрович

Проверил:  
Рязанов Юрий Дмитриевич

Белгород 2024 г.

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

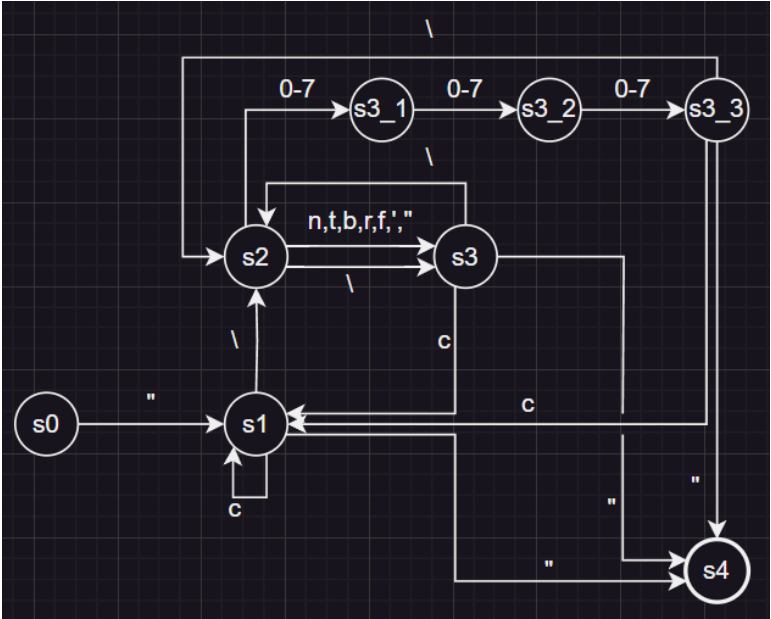
Вариант 2: (по старому пособию)

Язык строковых констант.

Строковая константа — последовательность символов, заключённая в кавычки. Последовательности символов `\n` , `\t` , `\b` , `\r` , `\f` , `\'` , `\"` , `\\` и символ `\` , за которым следует трёхразрядное восьмеричное число, считаются одним символом, а одиночный символ `\` недопустим.

**Задание 1. (по старому пособию)** Построить минимальный детерминированный конечный распознаватель заданного языка (см.варианты заданий).

Граф распознавателя:



Распознаватель:

$A = \langle S, X, s_0, o, F \rangle$

$S = \{s_0, s_1, s_2, s_3, s_{3\_1}, s_{3\_2}, s_{3\_3}, s_4\}$

$X = \{“, \backslash, n, t, b, r, f, ', 0-7\} + \text{другие символы}$

$s_0 = s_0$

$F = \{s_4\}$

Распознаватель в виде таблицы:

	↓							1
	s0	s1	s2	s3	s3_1	s3_2	s3_3	s4
c`		s1		s1			s1	
“	s1	s4		s4			s4	
\		s2	s3	s2			s2	
{“,n,t,b,r,f,`}			s3					
0-7			s3_1		s3_2	s3_3		

c` – все символы, кроме ‘, “ и \

Классы 0-эквивалентных преобразований:

	K1								K2
	s0	s1	s2	s3	s3_1	s3_2	s3_3		s4
c`	K1	K1 s1	K1	K1 s1	K1	K1	K1 s1	K1	K1
“	K1 s1	K2 s4	K1	K2 s4	K1	K1	K2 s4	K1	K1
\	K1	K1 s2	K1 s3	K1 s2	K1	K1	K1 s2	K1	K1
{“,n,t,b,r,f,”}	K1	K1	K1 s3	K1	K1	K1	K1	K1	K1
0-7	K1	K1	K1 s3_1	K1	K1 s3_2	K1 s3_3	K1	K1	K1

Классы 1-эквивалентных преобразований:

	K7	K6	K5	K4	K1	K2			K3
	s0	s2	s3_1	s3_2		s1	s3	s3_3	s4
c`	K1	K1	K1	K1	K1	K2 s1	K2 s1	K2 s1	K1
“	K2 s1	K1	K1	K1	K1	K3 s4	K3 s4	K3 s4	K1
\	K1	K1 s3	K1	K1	K1	K1 s2	K1 s2	K1 s2	K1
{“,n,t,b,r,f,”}	K1	K2 s3	K1	K1	K1	K1	K1	K1	K1
0-7	K1	K5 s3_1	K4 s3_2	K2 s3_3	K1	K1	K1	K1	K1

Получим распознаватель:

$A = \langle S, X, s_0, o, F \rangle$

$S = \{s_0, s_1, s_2, s_{3\_1}, s_{3\_2}, s_4\}$

$X = \{“, \backslash, n, t, b, r, f, ', 0-7\} + \text{другие символы}$

$s_0 = s_0$

$F = \{s_4\}$

Окончательная таблица переходов минимального распознавателя:

	↓					1
	s0	s1	s2	s3_1	s3_2	s4
c`		s1				
“	s1	s4				
\		s2	s3			
{“,n,t,b,r,f,”}			s1			
0-7			s3_1	s3_2	s1	

**Задание 1.** Язык  $L_1$  в алфавите  $\{a, b\}$ , представляющий собой множество цепочек, в которых символ  $a$  встречается не менее одного раза, а символ  $b$  — не более одного раза, задан грамматикой:

$S \rightarrow aAbA$

$S \rightarrow baA$

$A \rightarrow Aa$

$A \rightarrow \varepsilon$

Построить детерминированный конечный распознаватель языка  $L_1$ .

Устраним пустое правило:

$S \rightarrow aAbA$

$S \rightarrow aAb$

$S \rightarrow abA$

$S \rightarrow ab$

$S \rightarrow baA$   
 $S \rightarrow ba$   
 $A \rightarrow Aa$   
 $A \rightarrow a$

Устраним левую рекурсию:

$S \rightarrow aAbA$   
 $S \rightarrow aAb$   
 $S \rightarrow abA$   
 $S \rightarrow ab$   
 $S \rightarrow baA$   
 $S \rightarrow ba$   
 $A \rightarrow aB1$   
 $B1 \rightarrow aB1$   
 $B1 \rightarrow \varepsilon$

Устраним пустое правило:

$S \rightarrow aAbA$   
 $S \rightarrow aAb$   
 $S \rightarrow abA$   
 $S \rightarrow ab$   
 $S \rightarrow baA$   
 $S \rightarrow ba$   
 $A \rightarrow aB1$   
 $A \rightarrow a$   
 $B1 \rightarrow aB1$   
 $B1 \rightarrow a$

Правила начинаются с терминала, поэтому приведём грамматику к правосторонней грамматике:

$N1 \rightarrow Ab$  ( $N1 \rightarrow aB1b$ ,  $N1 \rightarrow ab$ )  
 $N2 \rightarrow B1b$  ( $N2 \rightarrow aN2$ ,  $N2 \rightarrow ab$ )

$N3 \rightarrow N1A$  ( $N3 \rightarrow aN2A$ ,  $N3 \rightarrow abA$ )  
 $N4 \rightarrow N2A$  ( $N4 \rightarrow aN4$ ,  $N4 \rightarrow abA$ )

$S \rightarrow aN3$   
 $S \rightarrow aN1$   
 $S \rightarrow abA$   
 $S \rightarrow ab$   
 $S \rightarrow baA$   
 $S \rightarrow ba$   
 $A \rightarrow aB1$   
 $A \rightarrow a$   
 $B1 \rightarrow aB1$   
 $B1 \rightarrow a$   
 $N1 \rightarrow aN2$   
 $N1 \rightarrow ab$   
 $N2 \rightarrow aN2$

$N_2 \rightarrow ab$   
 $N_3 \rightarrow aN_4$   
 $N_3 \rightarrow abA$   
 $N_4 \rightarrow aN_4$   
 $N_4 \rightarrow abA$

Приводим к автоматной правосторонней грамматике:

$S \rightarrow aN_3$   
 $S \rightarrow aN_1$   
 $S \rightarrow aS_1$   
 $S_1 \rightarrow bA$   
 $S \rightarrow aS_5$   
 $S_5 \rightarrow bE$   
 $S \rightarrow bS_2$   
 $S_2 \rightarrow aA$   
 $S \rightarrow bS_6$   
 $S_6 \rightarrow aE$   
 $A \rightarrow aB_1$   
 $A \rightarrow aE$   
 $B_1 \rightarrow aB_1$   
 $B_1 \rightarrow aE$   
 $N_1 \rightarrow aN_2$   
 $N_1 \rightarrow aS_7$   
 $S_7 \rightarrow bE$   
 $N_2 \rightarrow aN_2$   
 $N_2 \rightarrow aS_8$   
 $S_8 \rightarrow bE$   
 $N_3 \rightarrow aN_4$   
 $N_3 \rightarrow aS_3$   
 $S_3 \rightarrow bA$   
 $N_4 \rightarrow aN_4$   
 $N_4 \rightarrow aS_4$   
 $S_4 \rightarrow bA$   
 $E \rightarrow \epsilon$

Приведём к детерминированному автомату:

	S	S1	S2	S3	S4	S5	S6	S7	S8	A	B1	N1	N2	N3	N4	E
a	N3 N1 S1 S5		A				E			B1 E	B1 E	N2 S7	N2 S8	N4 S3	N4 S4	
b	S2 S6	A		A	A	E		E	E							

	{S}	{N3, N1, S1, S5}	{N4, S3, N2, S7}	{N4, S4, N2, S8}	{S2, S6}	{A, E}	{B1, E}
a	{N3, N1, S1, S5}	{N4, S3, N2, S7}	{N4, S4, N2, S8}	{N4, S4, N2, S8}	{A, E}	{B1, E}	{B1, E}
b	{S2, S6}	{A, E}	{A, E}	{A, E}			

Получим новый детерминированный распознаватель:

	↓					1	1
	S	S1	S2	S3	S4	A	B1
a	S1	S2	S3	S3	A	B1	B1
b	S4	A	A	A			

Получим минимизированный детерминированный распознаватель (потребуется в 3-ем задании):

	K1						K2	
	S	S1	S2	S3	S4		A	B1
a	K1 S1	K1 S2	K1 S3	K1 S3	K2 A	K1	K2 B1	K2 B1
b	K1 S4	K2 A	K2 A	K2 A	K1	K1	K1	K1

Класс 1-эквивалентности:

	K1						K2	
	S	S1	S2	S3	S4		A	B1
a	K1 S1	K1 S2	K1 S3	K1 S3	K2 A	K1	K2 B1	K2 B1
b	K1 S4	K2 A	K2 A	K2 A	K1	K1	K1	K1

Класс 2-эквивалентности:

	K1			K2			K3	
	S	S4		S1	S2	S3	A	B1
a	K2 S1	K3 A	K1	K2 S2	K2 S3	K2 S3	K3 B1	K3 B1
b	K1 S4	K1	K1	K3 A	K3 A	K3 A	K1	K1

Класс 3-эквивалентности:

	K1			K2			K3	
	S	S4		S1	S2	S3	A	B1
a	K2 S1	K3 A	K1	K2 S2	K2 S3	K2 S3	K3 B1	K3 B1
b	K1 S4	K1	K1	K3 A	K3 A	K3 A	K1	K1

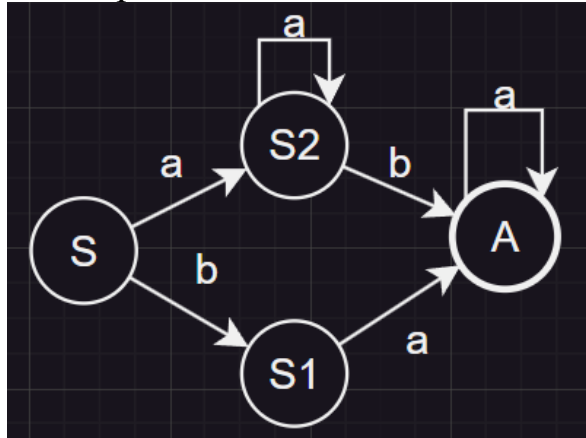
Класс 4-эквивалентности:

	K5	K4	K1	K2			K3	
	S	S4		S1	S2	S3	A	B1
a	K2 S1	K3 A	K1	K2 S2	K2 S3	K2 S3	K3 B1	K3 B1
b	K4 S4	K1	K1	K3 A	K3 A	K3 A	K1	K1

Распознаватель:

	↓			1
	S	S1	S2	A
a	S2		S2	A
b	S1	A	A	

Схема распознавателя:

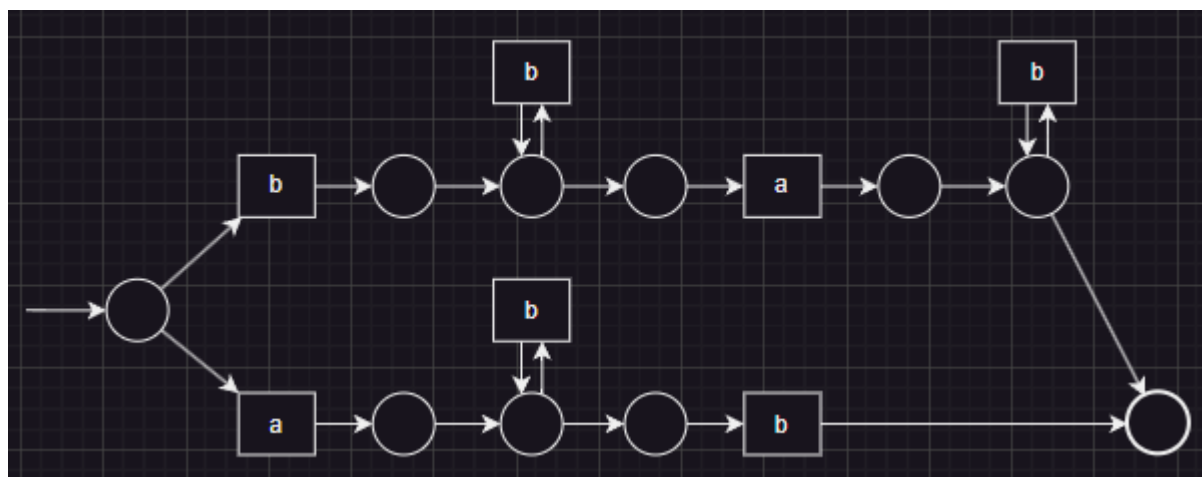
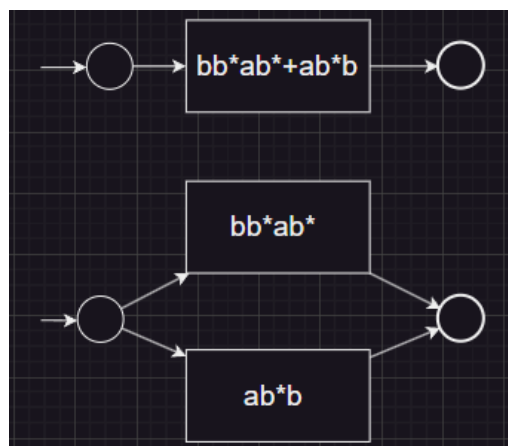


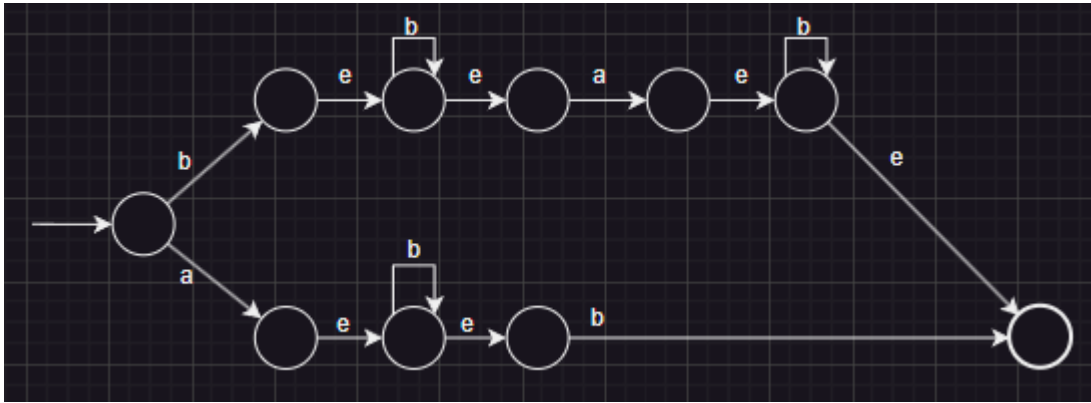
**Задание 2.** Язык  $L_2$  в алфавите  $\{a,b\}$ , представляющий собой множество цепочек, в которых символ  $b$  встречается не менее одного раза, а символ  $a$  — не более одного раза, задан регулярным выражением:

$bb^*ab^*+ab^*b$

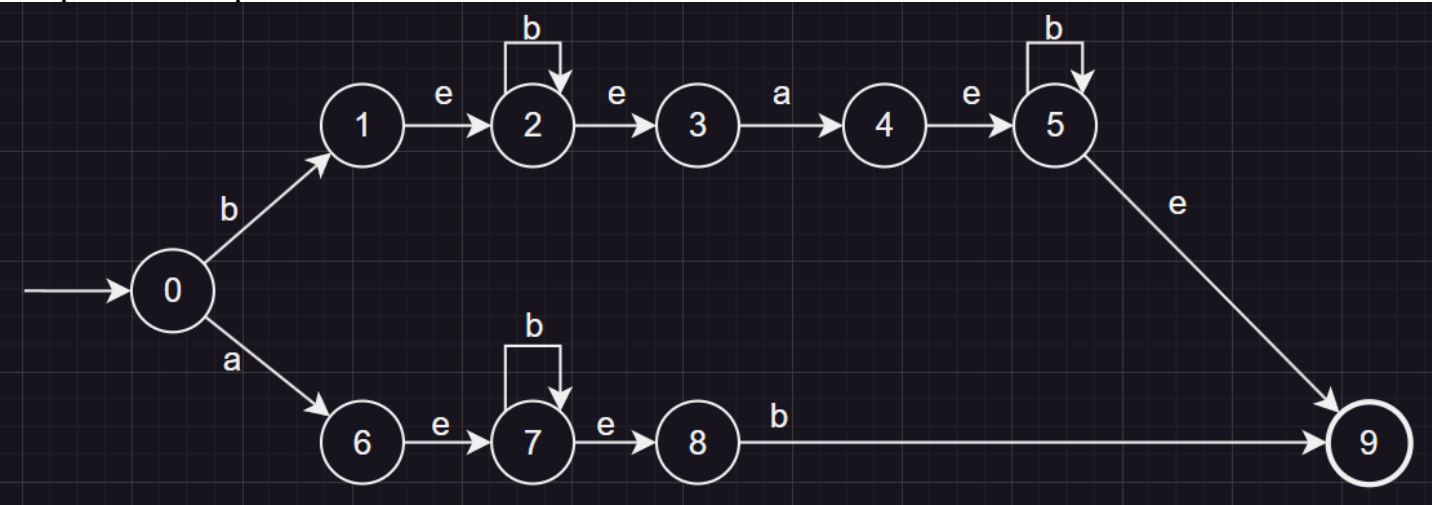
Построить детерминированный конечный распознаватель языка  $L_2$ .

Вывод недетерминированного распознавателя по шагам:





Устраним е-переходы:



Недетерминированный конечный распознаватель:

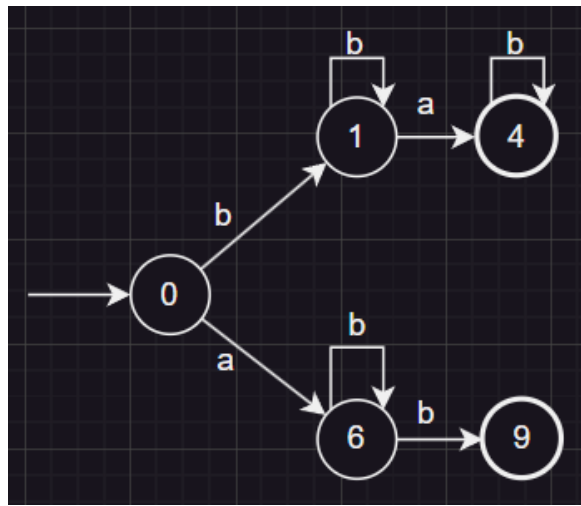
	S0	S1	S2	S3	S4	S5	S6	S7	S8	S9
a	S6			S4						
b	S1		S2			S5		S7	S9	
e		S2	S3		S5	S9	S7	S8		

Устраним е-переходы:

	e(S0) {S0}	e(S1) {S1,S2,S3}	e(S2) {S2,S3}	e(S3) {S3}	e(S4) {S4,S5,S9}	e(S5) {S5,S9}	e(S6) {S6,S7,S8}	e(S7) {S7,S8}	e(S8)	e(S9)
a	e(S6)	e(S4)	e(S4)	e(S4)						
b	e(S1)	e(S2)	e(S2)		e(S5), e(S9)	e(S5), e(S9)	e(S7), e(S9)	e(S7), e(S9)	e(S9)	

	↓		1		1
	S0	S1	S4	S6	S9
a	S6	S4			
b	S1	S1	S4	S6,S9	





Приведём к детерминированному конечному распознавателю:

	↓		1		1
	{S0}	{S1}	{S4}	{S6}	{S6,S9}
a	{S6}	{S4}			
b	{S1}	{S1}	{S4}	{S6,S9}	{S6,S9}

	↓		1		1
	S0	S1	S4	S6	S6_1
a	S6	S4			
b	S1	S1	S4	S6_1	S6_1

Получим минимизированный детерминированный конечный распознаватель (потребуется в 3-ем задании):

	K1				K2	
	S0	S1	S6		S6_1	S4
a	K1 S6	K2 S4	K1	K1	K1	K1
b	K1 S1	K1 S1	K2 S6_1	K1	K2 S6_1	K2 S4

Класс 1-эквивалентности:

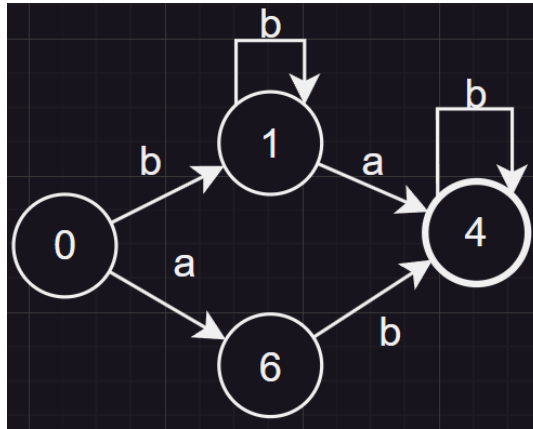
	K1				K2	
	S0	S1	S6		S6_1	S4
a	K1 S6	K2 S4	K1	K1	K1	K1
b	K1 S1	K1 S1	K2 S6_1	K1	K2 S6_1	K2 S4

Класс 2-эквивалентности:

	K5	K4	K3	K1	K2	
	S0	S1	S6		S6_1	S4
a	K1 S6	K2 S4	K1	K1	K1	K1
b	K1 S1	K1 S1	K2 S6_1	K1	K2 S6_1	K2 S4

Получим минимизированный детерминированный конечный распознаватель:

	↓			1
	S0	S1	S6	S4
a	S6	S4		
b	S1	S1	S4	S4

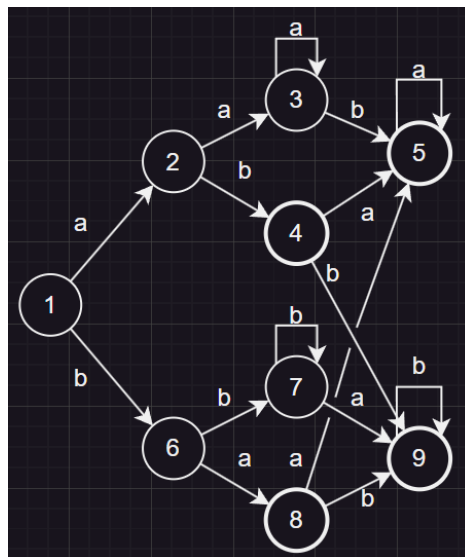


**Задание 3.** Построить минимальный детерминированный конечный распознаватель языка L3 в алфавите {a,b}, представляющий собой множество цепочек, в которых символ a встречается не менее одного раза, а символ b — не более одного раза, или символ b встречается не менее одного раза, а символ a — не более одного раза.

Заметим, что язык L1 и L2 схожи за исключением того, что символы инвертированы. В языке L3 нужно учесть верные строки 'ab' и 'ba' и также неоднозначность после 'ab' и 'ba', где могут продолжиться строчка из a или строчка из b.

Детерминированный конечный распознаватель имеет вид:

	↓			1	1			1	1
	S1	S2	S3	S4	S5	S6	S7	S8	S9
a	S2	S3	S3	S5	S5	S8	S9	S5	
b	S6	S4	S5	S9		S7	S7	S9	S9



Минимизируем, используя эквивалентные преобразования:

	K1						K2			
	S1	S2	S3	S6	S7		S4	S8	S5	S9
a	K1	K1	K1	K1	K2	K1	K2	K2	K2	K1
b	K1	K1	K2	K1	K1	K1	K2	K2	K1	K2
	S2	S3	S3	S8	S9		S5	S5	S5	
	S6	S4	S5	S7	S7		S9	S9		S9

Классы 1-эквивалентности:

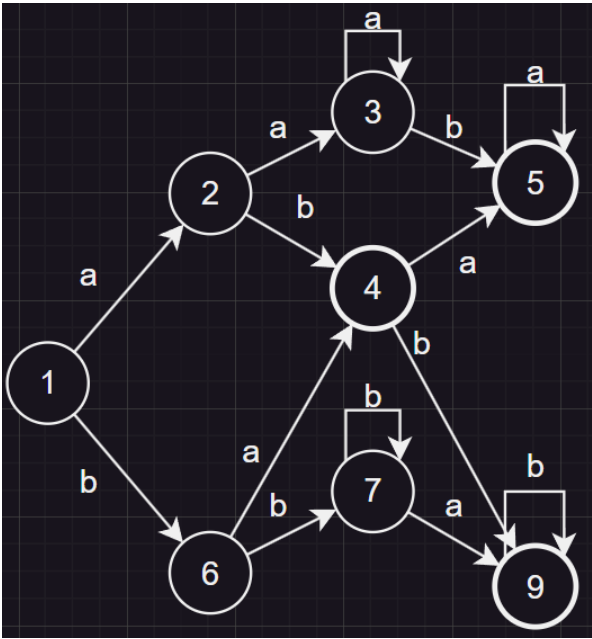
	K1						K3		K2	
	S1	S2	S3	S6	S7		S4	S8	S5	S9
a	K1	K1	K1	K3	K2	K1	K2	K2	K2	K1
	S2	S3	S3	S8	S9		S5	S5	S5	
b	K1	K3	K2	K1	K1	K1	K2	K2	K1	K2
	S6	S4	S5	S7	S7		S9	S9		S9

Классы 2-эквивалентности:

	K8	K7	K6	K5	K4	K1	K3		K2	
	S1	S2	S3	S6	S7		S4	S8	S5	S9
a	K7	K6	K6	K3	K2	K1	K2	K2	K2	K1
	S2	S3	S3	S8	S9		S5	S5	S5	
b	K6	K3	K2	K4	K4	K1	K2	K2	K1	K2
	S6	S4	S5	S7	S7		S9	S9		S9

Получим минимизированный детерминированный конечный распознаватель:

	↓			1	1			1
	S1	S2	S3	S4	S5	S6	S7	S9
a	S2	S3	S3	S5	S5	S4	S9	
b	S6	S4	S5	S9		S7	S7	S9



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

Код программы:

```
MESSAGES = {
    -1: "Отвергнуть. Последовательность пуста",
    -2: "Отвергнуть. Входной символ невалидный",
    -3: "Отвергнуть. Символ b должен быть введён не более 1 раза",
    -4: "Отвергнуть. Символ a должен быть введён не более 1 раза",
    0: "Допустить."
}

def L3validator_comp(input: str):
    S = 1
    while (True):
        if S == 1:
            if len(input) == 0:
                S = -1
            elif input[0] == 'a':
                S = 2
            elif input[0] == 'b':
                S = 6
            else:
                S = -2
        elif S == 2:
            if len(input) == 0:
                S = -5
            elif input[0] == 'a':
                S = 3
            elif input[0] == 'b':
                S = 4
            else:
                S = -2
        elif S == 3:
            if len(input) == 0:
                S = -5
            elif input[0] == 'a':
                S = 2
            elif input[0] == 'b':
                S = 5
            else:
                S = -2
        elif S == 4:
            if len(input) == 0:
                S = 0
            elif input[0] == 'a':
                S = 5
            elif input[0] == 'b':
                S = 9
            else:
                S = -2
        elif S == 5:
            if len(input) == 0:
                S = 0
            elif input[0] == 'a':
                S = 5
            elif input[0] == 'b':
                S = -3
            else:
                S = -2
        elif S == 6:
            if len(input) == 0:
                S = -6
            elif input[0] == 'a':
                S = 4
            elif input[0] == 'b':
                S = 7
```

```

        else:
            S = -2
    elif S == 7:
        if len(input) == 0:
            S = -6
        elif input[0] == 'a':
            S = 9
        elif input[0] == 'b':
            S = 7
        else:
            S = -2
    elif S == 9:
        if len(input) == 0:
            S = 0
        elif input[0] == 'a':
            S = -4
        elif input[0] == 'b':
            S = 9
        else:
            S = -2
    else:
        raise Exception("Undefined state.")

    if S <= 0:
        return S

    input = input[1:]

```

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

Код программы:

```

CONVERT = {
    2: -5,
    3: -5,
    6: -6,
    7: -6,
    -1: -1,
    -2: -2,
    -3: -3,
    -4: -4,
    -5: -5,
    -6: -6,
}

MESSAGES = {
    -1: "Отвергнуть. Последовательность пуста",
    -2: "Отвергнуть. Входной символ невалидный",
    -3: "Отвергнуть. Символ b должен быть введён не более 1 раза",
    -4: "Отвергнуть. Символ a должен быть введён не более 1 раза",
    -5: "Отвергнуть. Символ b должен быть введён хотя бы 1 раз",
    -6: "Отвергнуть. Символ a должен быть введён хотя бы 1 раз",
    0: "Допустить",
}

PERMITTING = [4, 5, 8]

MATRIX = {
    'a': [2, 3, 3, 5, 5, 4, 8, -4],
    'b': [6, 4, 5, 8, -3, 7, 7, 8],
}

def L3validator_interp(input):
    if len(input) == 0:
        return -1

    S = 1

```

```

while len(input) > 0 and S > 0:
    if not input[0] in MATRIX.keys():
        return -2
    S = MATRIX[input[0]][S - 1]
    input = input[1:]

if S in PERMITTING:
    return 0

return S

```

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

Тестовые данные:

"aaab"	Допустить
"aab"	Допустить
"aaba"	Допустить
"ab"	Допустить
"aba"	Допустить
"abaa"	Допустить
"ba"	Допустить
"bab"	Допустить
"babb"	Допустить
"bbba"	Допустить
"bba"	Допустить
"bbbab"	Допустить
"bbab"	Допустить

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

Тестовые данные:

""	S1	Отвергнуть
"c"	S1	Отвергнуть
"a"	S2	Отвергнуть
"aaa"	S3	Отвергнуть
"ab"	S4	Допустить
"ba"	S4	Допустить
"aab"	S5	Допустить
"aba"	S5	Допустить
"baa"	S5	Допустить
"aabb"	S5	Отвергнуть
"b"	S6	Отвергнуть
"bb"	S7	Отвергнуть
"bba"	S9	Допустить
"bab"	S9	Допустить
"abb"	S9	Допустить
"bbaa"	S9	Отвергнуть

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

```
check_set = [
    ("aaab", 1),
    ("aab", 1),
    ("aaba", 1),
    ("ab", 1),
    ("aba", 1),
    ("abaa", 1),
    ("ba", 1),
    ("bab", 1),
    ("babb", 1),
    ("bbba", 1),
    ("bba", 1),
    ("bbbab", 1),
    ("bbab", 1),
    ("", 0), # 1
    ("c", 0), # 1
    ("a", 0), # 2
    ("aaa", 0), # 3
    ("ab", 1), # 4
    ("ba", 1), # 4
    ("aab", 1), # 5
    ("aba", 1), # 5
    ("baa", 1), # 5
    ("aabb", 0), # 5
    ("b", 0), # 6
    ("bb", 0), # 7
    ("bba", 1), # 9
    ("bab", 1), # 9
    ("abb", 1), # 9
    ("bbaa", 0), # 9
]

for check in check_set:
    res = L3validator(check[0])
    if check[1]:
        print(check[0], res >= 0, MESSAGES[res], sep='\t')
    else:
        print(check[0], res < 0, MESSAGES[res], sep='\t')
```

**Вывод программы:**

aaab	True	Допустить.
aab	True	Допустить.
aaba	True	Допустить.
ab	True	Допустить.
aba	True	Допустить.
abaa	True	Допустить.
ba	True	Допустить.
bab	True	Допустить.
babb	True	Допустить.
bbba	True	Допустить.
bba	True	Допустить.
bbbab	True	Допустить.
bbab	True	Допустить.
	True	Отвергнуть. Последовательность пуста
c	True	Отвергнуть. Входной символ невалидный
a	True	Отвергнуть. Символ b должен быть введен хотя бы 1 раз
aaa	True	Отвергнуть. Символ b должен быть введен хотя бы 1 раз
ab	True	Допустить.

ba	True	Допустить.
aab	True	Допустить.
aba	True	Допустить.
baa	True	Допустить.
aabb	True	Отвергнуть. Символ b должен быть введен не более 1 раза
b	True	Отвергнуть. Символ a должен быть введен хотя бы 1 раз
bb	True	Отвергнуть. Символ a должен быть введен хотя бы 1 раз
bba	True	Допустить.
bab	True	Допустить.
abb	True	Допустить.
bbaa	True	Отвергнуть. Символ a должен быть введен не более 1 раза



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