

**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)**



**ИНСТИТУТ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ И УПРАВЛЯЮЩИХ СИСТЕМ**

**Лабораторная работа №3**

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

Выполнил: ст. группы ПВ-223  
Степанов Дмитрий Сергеевич

Проверили:  
Рязанов Юрий Дмитриевич

Белгород 2024 г.

**Лабораторная работа №3**  
**Регулярные языки и конечные распознаватели**  
**Вариант 1**

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

1. Язык  $L_1$  в алфавите  $\{a,b,c\}$ , представляющий собой множество цепочек, которые содержат хотя бы один символ  $a$ , задан грамматикой:

$S \rightarrow RA$

$R \rightarrow Rb$

$R \rightarrow Rc$

$R \rightarrow \epsilon$

$A \rightarrow aP$

$P \rightarrow aP$

$P \rightarrow bP$

$P \rightarrow cP$

$P \rightarrow \epsilon$

Построить детерминированный конечный распознаватель языка  $L_1$ .

Чтобы получить распознаватель необходимо преобразовать грамматику в автоматную правую.

Удалим  $\epsilon$ -правила:

$S \rightarrow RA$

$S \rightarrow A$

$R \rightarrow Rb$

$R \rightarrow b$

$R \rightarrow Rc$

$R \rightarrow c$

$A \rightarrow aP$

$A \rightarrow a$

$P \rightarrow aP$

$P \rightarrow a$

$P \rightarrow bP$

$P \rightarrow b$

$P \rightarrow cP$

$P \rightarrow c$

Избавимся от цепных правил:

$S \rightarrow RA$

$S \rightarrow aP$

$S \rightarrow a$

$R \rightarrow Rb$

$R \rightarrow Rc$

$R \rightarrow b$

$R \rightarrow c$   
 $A \rightarrow aP$   
 $A \rightarrow a$   
 $P \rightarrow aP$   
 $P \rightarrow a$   
 $P \rightarrow bP$   
 $P \rightarrow b$   
 $P \rightarrow cP$   
 $P \rightarrow c$

Избавимся от левой рекурсии:

$S \rightarrow RA$   
 $S \rightarrow aP$   
 $S \rightarrow a$   
 $R \rightarrow bB1$   
 $R \rightarrow cB1$   
 $A \rightarrow aP$   
 $A \rightarrow a$   
 $P \rightarrow aP$   
 $P \rightarrow a$   
 $P \rightarrow bP$   
 $P \rightarrow b$   
 $P \rightarrow cP$   
 $P \rightarrow c$   
 $B1 \rightarrow bB1$   
 $B1 \rightarrow cB1$   
 $B1 \rightarrow \varepsilon$

Избавимся от  $\varepsilon$ -правил:

$S \rightarrow RA$   
 $S \rightarrow aP$   
 $S \rightarrow a$   
 $R \rightarrow bB1$   
 $R \rightarrow b$   
 $R \rightarrow cB1$   
 $R \rightarrow c$   
 $A \rightarrow aP$   
 $A \rightarrow a$   
 $P \rightarrow aP$   
 $P \rightarrow a$   
 $P \rightarrow bP$   
 $P \rightarrow b$   
 $P \rightarrow cP$   
 $P \rightarrow c$   
 $B1 \rightarrow bB1$   
 $B1 \rightarrow cB1$   
 $B1 \rightarrow b$   
 $B1 \rightarrow c$

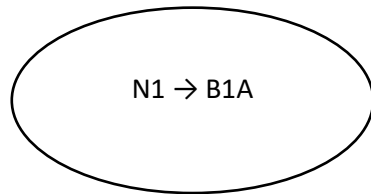
Преобразуем грамматику к виду, когда все правые части правил начинаются с терминала:

$$S \rightarrow bB1A$$
$$S \rightarrow cB1A$$
$$S \rightarrow bA$$
$$S \rightarrow cA$$
$$S \rightarrow aP$$
$$S \rightarrow a$$
$$R \rightarrow bB1$$
$$R \rightarrow b$$
$$R \rightarrow cB1$$
$$R \rightarrow c$$
$$A \rightarrow aP$$
$$A \rightarrow a$$
$$P \rightarrow aP$$
$$P \rightarrow a$$
$$P \rightarrow bP$$
$$P \rightarrow b$$
$$P \rightarrow cP$$
$$P \rightarrow c$$
$$B1 \rightarrow bB1$$
$$B1 \rightarrow cB1$$
$$B1 \rightarrow b$$
$$B1 \rightarrow c$$

Удалим недостижимые символы:

$$S \rightarrow bB1A$$
$$S \rightarrow cB1A$$
$$S \rightarrow bA$$
$$S \rightarrow cA$$
$$S \rightarrow aP$$
$$S \rightarrow a$$
$$A \rightarrow aP$$
$$A \rightarrow a$$
$$P \rightarrow aP$$
$$P \rightarrow a$$
$$P \rightarrow bP$$
$$P \rightarrow b$$
$$P \rightarrow cP$$
$$P \rightarrow c$$
$$B1 \rightarrow bB1$$
$$B1 \rightarrow cB1$$
$$B1 \rightarrow b$$
$$B1 \rightarrow c$$

Приведём грамматику к правосторонней грамматике:



$N1 \rightarrow bB1A$ , то же что и  $N1 \rightarrow bN1$

$N1 \rightarrow cB1A$ , то же что и  $N1 \rightarrow cN1$

$N1 \rightarrow bA$

$N1 \rightarrow cA$

Привели грамматику к правосторонней:

$S \rightarrow bN1$

$S \rightarrow cN1$

$S \rightarrow aP$

$S \rightarrow a$

$S \rightarrow bA$

$S \rightarrow cA$

$A \rightarrow aP$

$A \rightarrow a$

$P \rightarrow aP$

$P \rightarrow a$

$P \rightarrow bP$

$P \rightarrow b$

$P \rightarrow cP$

$P \rightarrow c$

$B1 \rightarrow bB1$

$B1 \rightarrow cB1$

$B1 \rightarrow b$

$B1 \rightarrow c$

$N1 \rightarrow bN1$

$N1 \rightarrow cN1$

$N1 \rightarrow bA$

$N1 \rightarrow cA$

Приведём грамматику к правосторонней автоматной. Введём правило  $N2 \rightarrow \varepsilon$ :

$S \rightarrow bN1$

$S \rightarrow cN1$

$S \rightarrow aP$

$S \rightarrow aN2$

$S \rightarrow bA$

$S \rightarrow cA$

$A \rightarrow aP$

$A \rightarrow aN2$

$P \rightarrow aP$

$P \rightarrow aN_2$   
 $P \rightarrow bP$   
 $P \rightarrow bN_2$   
 $P \rightarrow cP$   
 $P \rightarrow cN_2$   
 $B_1 \rightarrow bB_1$   
 $B_1 \rightarrow cB_1$   
 $B_1 \rightarrow bN_2$   
 $B_1 \rightarrow cN_2$   
 $N_1 \rightarrow bN_1$   
 $N_1 \rightarrow cN_1$   
 $N_1 \rightarrow bA$   
 $N_1 \rightarrow cA$   
 $N_2 \rightarrow \varepsilon$

Теперь можем построить распознаватель:

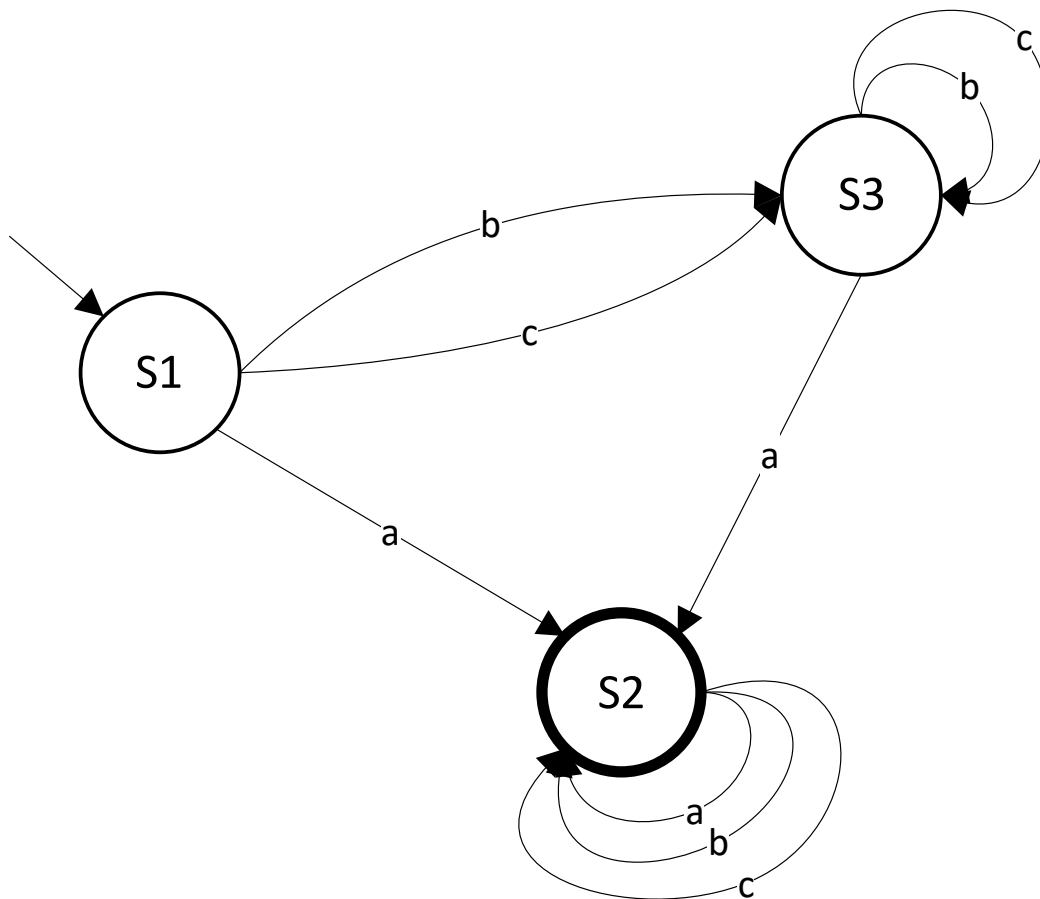
	↓					1
	S	A	P	B1	N1	N2
a	P, N2	P, N2	P, N2			
b	N1, A		P, N2	B1, N2	N1, A	
c	N1, A		P, N2	B1, N2	N1, A	

Приведём недетерминированный распознаватель к детерминированному:

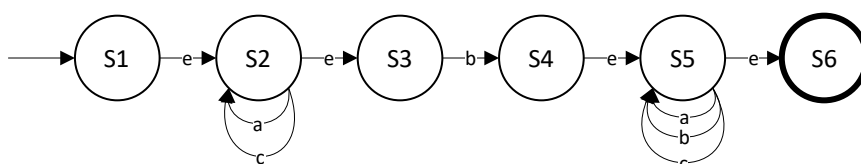
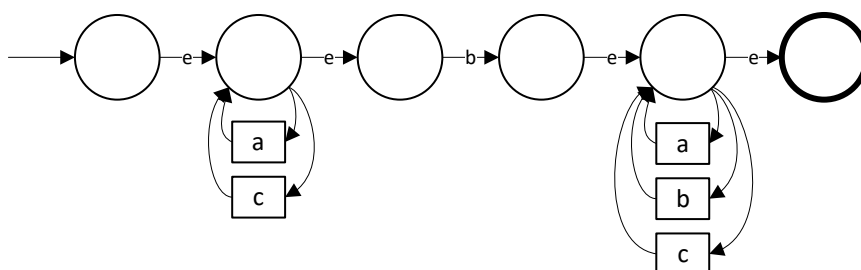
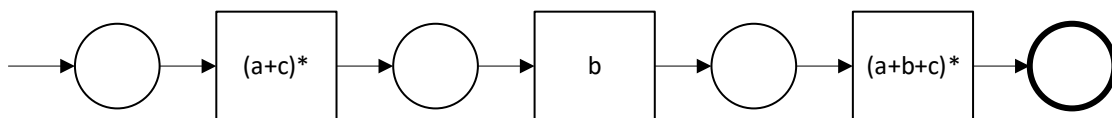
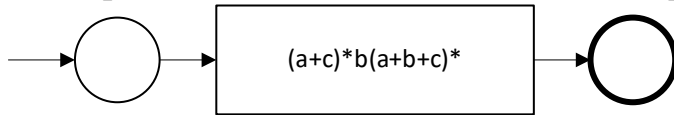
	↓	1	
	{S}	{P, N2}	{N1, A}
a	{P, N2}	{P, N2}	{P, N2}
b	{N1, A}	{P, N2}	{N1, A}
c	{N1, A}	{P, N2}	{N1, A}

	↓	1	
	S1	S2	S3
a	S2	S2	S2
b	S3	S2	S3
c	S3	S2	S3

Получили детерминированный распознаватель для языка L1:



2. Язык  $L_2$  в алфавите  $\{a,b,c\}$ , представляющий собой множество цепочек, которые содержат хотя бы один символ  $b$ , задан регулярным выражением:  $(a+c)^*b(a+b+c)^*$



Избавимся от е-переходов:

$e(S1) = \{S1, S2, S3\}$

$e(S2) = \{S2, S3\}$

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

$e(S4) = \{S4, S5, S6\}$

$e(S5) = \{S5, S6\}$

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

	↓	↓	↓	1	1	1
	$e(S1)$ $\{S1, S2, S3\}$	$e(S2)$ $\{S2, S3\}$	$e(S3)$ $\{S3\}$	$e(S4)$ $\{S4, S5, S6\}$	$e(S5)$ $\{S5, S6\}$	$e(S6)$ $\{S6\}$
a	$e(S2)$	$e(S2)$		$e(S5)$	$e(S5)$	
b	$e(S4)$	$e(S4)$	$e(S4)$	$e(S5)$	$e(S5)$	
c	$e(S2)$	$e(S2)$		$e(S5)$	$e(S5)$	

Удалим недостижимый символ S6

	↓	↓	↓	1	1
	S1	S2	S3	S4	S5
a	S2	S2		S5	S5
b	S4	S4	S4	S5	S5
c	S2	S2		S5	S5

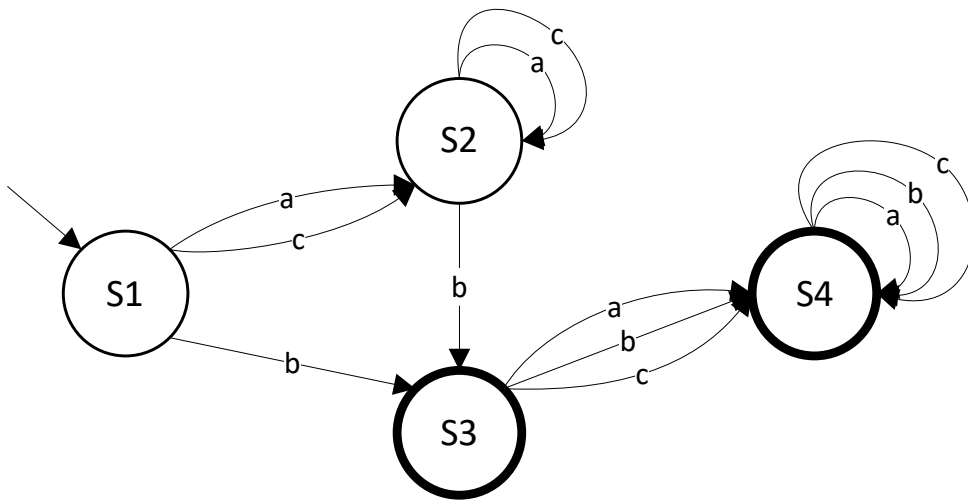
Преобразуем недетерминированный распознаватель в детерминированный

	↓		1	1
	$\{S1, S2, S3\}$	$\{S2\}$	$\{S4\}$	$\{S5\}$
a	$\{S2\}$	$\{S2\}$	$\{S5\}$	$\{S5\}$
b	$\{S4\}$	$\{S4\}$	$\{S5\}$	$\{S5\}$
c	$\{S2\}$	$\{S2\}$	$\{S5\}$	$\{S5\}$

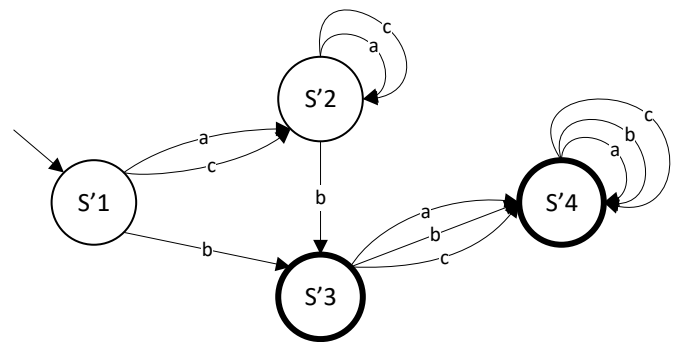
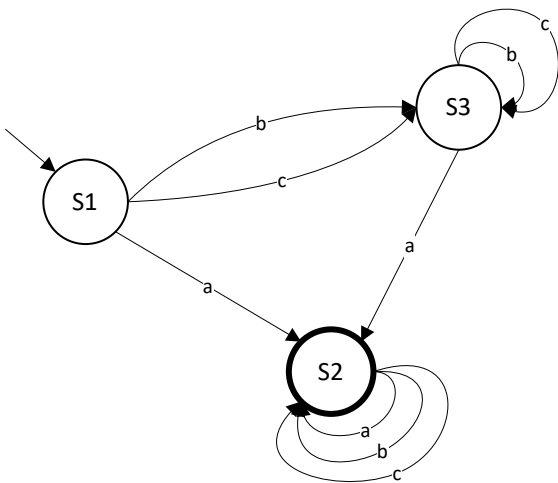
	↓		1	1
	S1	S2	S3	S4
a	S2	S2	S4	S4
b	S3	S3	S4	S4
c	S2	S2	S4	S4

Получили детерминированный распознаватель для языка L2





3. Построить минимальный детерминированный конечный распознаватель языка  $L_3$  в алфавите  $\{a, b, c\}$ , представляющий собой множество цепочек, которые содержат хотя бы один символ  $a$  или хотя бы один символ  $b$ .



Преобразуем недетерминированный распознаватель в детерминированный:

	↓	1	1		1	1	1
	{S1, S'1}	{S2, S'2}	{S3, S'3}	{S3, S'2}	{S2, S'3}	{S2, S'4}	{S3, S'4}
a	{S2, S'2}	{S2, S'2}	{S2, S'4}	{S2, S'2}	{S2, S'4}	{S2, S'4}	{S2, S'4}
b	{S3, S'3}	{S2, S'3}	{S3, S'4}	{S3, S'3}	{S2, S'4}	{S2, S'4}	{S3, S'4}
c	{S3, S'2}	{S2, S'2}	{S3, S'4}	{S3, S'2}	{S2, S'4}	{S2, S'4}	{S3, S'4}

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

Выполним минимизацию:

$K1 = \{S1, S4, e\}$

$K2 = \{S2, S3, S5, S6, S7\}$

	K1			K2				
	S1	S4	e	S2	S3	S5	S6	S7
a	K2	K2	K1	K2	K2	K2	K2	K2
b	K2	K2	K1	K2	K2	K2	K2	K2
c	K1	K1	K1	K2	K2	K2	K2	K2

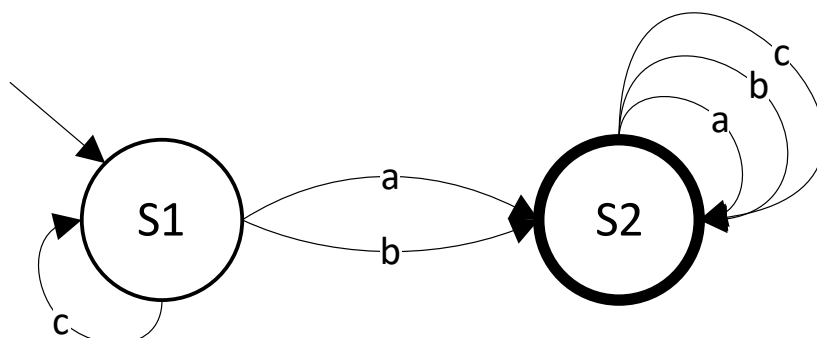
Получили таблицу переходов в классы 0-эквивалентных состояний. На основе этой таблицы можем построить таблицу переходов в классы 1-эквивалентных состояний.

	K1		K3	K2				
	S1	S4	e	S2	S3	S5	S6	S7
a	K2	K2	K3	K2	K2	K2	K2	K2
b	K2	K2	K3	K2	K2	K2	K2	K2
c	K1	K1	K3	K2	K2	K2	K2	K2

По таблице видно, что классы 2-эквивалентных состояний совпадают с классами 1-эквивалентных состояний, следовательно, классы 1-эквивалентных состояний представляют собой классы эквивалентных состояний. Переобозначим  $K1=S1$ ,  $K2=S2$ . Таблица переходов минимального распознавателя:

	↓	1
	S1	S2
a	S2	S2
b	S2	S2
c	S1	S2

Получили минимальный детерминированный распознаватель для языка  $L3$ .



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

```
MESSAGES = {
    -1: "Отвергнуть, невалидный входной символ",
    0: "Отвергнуть, цепочка не содержит а или b",
    1: "Допустить",
}

def validate(input):
    original_input = input
    s = 0
    while len(input) > 0 and s >= 0:
        current_symbol = input[0]
        if s == 0:
            if current_symbol == "c":
                s = 0
            elif current_symbol == "a" or current_symbol == "b":
                s = 1
            else:
                s = -1
                break
        elif s == 1:
            if current_symbol == "a" or current_symbol == "b" or current_symbol == "c":
                s = 1
            else:
                s = -1
                break

        input = input[1:]

    print(original_input, MESSAGES[s])
    return s
```

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

```
MESSAGES = {
    -1: "Отвергнуть, невалидный входной символ",
    0: "Отвергнуть, цепочка не содержит а или b",
    1: "Допустить",
}

PERMITTING = [1]

MATRIX = {
    "a": [1, 1],
    "b": [1, 1],
    "c": [0, 1]
}

def validate(input):
    original_input = input
    s = 0
    while len(input) > 0 and s >= 0:
        current_symbol = input[0]
        if current_symbol in MATRIX:
            s = MATRIX[input[0]][s]
        else:
            s = -1
            break
```

```
input = input[1:]  
  
if s in PERMITTING:  
    s = 1  
  
print(original_input, MESSAGES[s])  
return s
```

6. Подобрать наборы тестовых данных так, чтобы в процессе тестирования сработал каждый переход конечного распознавателя.
  - a. саabc
  - b. cbabc
7. Подобрать наборы тестовых данных так, чтобы в процессе тестирования распознаватель закончил обработку цепочек в каждом состоянии конечного распознавателя.
  - a. ccc
  - b. a
8. Выполнить тестирование программ для реализации минимального детерминированного конечного распознавателя языка L3.

Результат выполнения программ:

саabc Допустить  
cbabc Допустить  
ccc Отвергнуть, цепочка не содержит a или b  
a Допустить

саabc Допустить  
cbabc Допустить  
ccc Отвергнуть, цепочка не содержит a или b  
a Допустить

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