ЛЕКЦИЯ 2

Регулярные языки и лексический анализ

Пример: Текст программы, который идет на вход лексического анализатора:

```
Program prim;
Var x, y, z: real;
Begin
    X : = 5;
    Y : = 5;
    Z : = X + Y;
End.
```

Известны коды типов лексем:

10 – ключевое слово.

20 – разделитель.

30 – идентификатор.

40 – константа.

Внутренние таблицы лексического анализатора

Таблица ключевых слов:

№	Ключевое слово
1	PROGRAM
2	BEGIN
3	END
4	FOR
5	REAL
6	VAR

Таблица разделителей:

№	Разделитель
1	;
2	,
3	+
4	-
5	/
6	*
7	:
8	=
9	.

Выход лексического анализатора

Таблица идентификаторов:

№	Идентификатор
1	Prim
2	X
3	Y
4	Z

Таблица констант:

№	Значение
1	5

Дескрипторный текст:

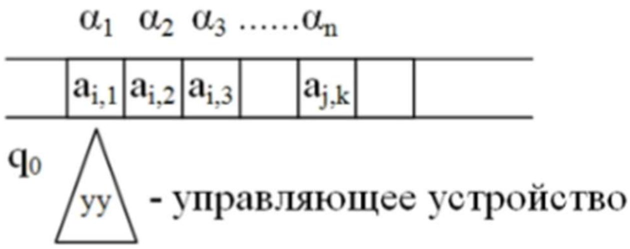
(10,1) (30,1) (20,1)
(10,6) (30,2) (20,2) (30,3) (20,2) (30,4) (20,7) (10,5) (20,1)
(10,2)
(30,2) (20,7) (20,8) (40,1) (20,1)
(30,3) (20,7) (20,8) (40,1) (20,1)
(30,4) (20,7) (20,8) (30,2) (20,3) (30,3) (20,1)
(10,3) (20,9)

Конечные автоматы

Конечным автоматом называется $A = \{V, Q, \Delta, q_0, F\}$, где

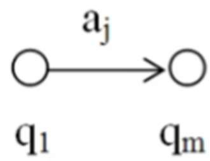
- $V = \{ a_1, a_2, a_3, \dots ,a_m \}$ – конечное множество символов входного алфавита.
- $Q = \{ q_0, q_1, q_2, q_3, \dots ,q_{n-1} \}$ – конечное множество алфавита состояний.
- $q_0 \in Q$ – начальное состояние автомата.
- Δ: $Q \times V \rightarrow Q$ – функция переходов.
- $F \subseteq Q$ – множество допустимых состояний или заключительных.

КА можно представить следующим образом:



Функцию переходов можно представить в виде:

- команды (q_i , a_j) → q_m , где q_i и $q_m \in Q$, $a_j \in V$
- дуг графа (графическое представление)
- матрицы переходов

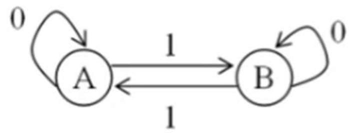


Пример: Пусть дан алфавит V = { 0, 1}, множество состояний Q = { A, B }, где начальное состояний q_0 = A, а F = { A }.

Переходы

- δ(A, 0) = A
- δ(A, 1) = B
- δ(B, 0) = B
- δ(B, 1) = A

	0	1	
A	A	B	1
B	B	A	0



При чтении цепочки 01001011 управление последовательно передается в следующем порядке:

$A^0 \to A^1 \to B^0 \to B^0 \to B^1 \to A^0 \to A^1 \to B^1 \to A$ – данная цепочка допускается, т.к. A – допускающее состояние.

Построение конечного автомата (КА) по регулярной грамматике

Пусть дана регулярная грамматика G={N, T, P, S}.

Правила G имеют вид: $A_i \to a_j A_k$ или $A_i \to a_j$, где $A_i, A_k \in N$ и $a_j \in T$.

Тогда КА A={V, Q, δ, q_0, F}, допускающий тот же самый язык, что порождает регулярная грамматика G, строится следующим образом:

1. V=T;
2. Q=N U{Z}, Z∉T, Z∉N, Z – заключительное состояние КА;
3. q_0={S};
4. F={Z};
5. δ строится в след. виде:

a) каждому правилу подстановки в грамматике G вида $A_i \to a_j A_k$ ставится в соответствие команда $(A_i, a_j) \to A_k$.

b) каждому правилу подстановки в грамматике G вида $A_i \to a_j$ ставится в соответствие команда $(A_i, a_j) \to Z$.

Пример: G={N, T, P,S}, б, ц – обозначения букв и цифр соответственно.
N={I,K}, T={б,ц}, S={I},
P={ 1. I→б
2. I→бK
3. K→бK
4. K→цK
5. K→б
6. K→ц }.

Построить для регулярной грамматики G КА A={V, Q, δ, q_0,F}, где:
1. V=T={б,ц};
2. Q=N ∪ {Z}={I, K, Z}, Z – заключительное состояние КА;
3. q_0={S}={I};
4. F={Z};
5. функция δ: a) каждому правилу подстановки в G вида Ai→ajAk ставится в соответствие команда (Ai, aj)→Ak.

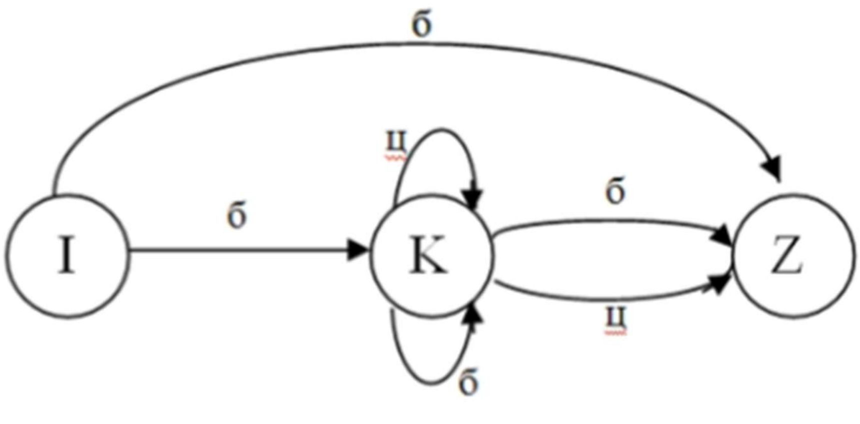
b) каждому правилу подстановки в G вида Ai→aj ставится в соответствие команда (Ai, aj)→Z.

Решение:

в виде совокупности команд:

1. (I,б)→Z
2. (I,б)→K
3. (K,б)→K
4. (K,ц)→K
5. (K,б)→Z
6. (K,ц)→Z.

в виде диаграммы состояний:

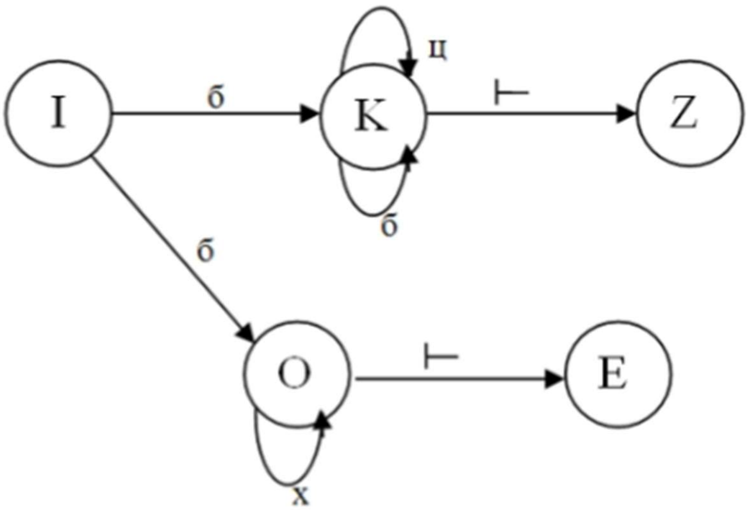


концевой маркер |—

Z – допустить входную цепочку;

O – запомнена ошибка во входной цепочке;

E – отвергнуть входную цепочку.



Эквивалентные состояния

	0	**1**	
x	**x**	**y**	**0**
y	**z**	**x**	**1**
z	**x**	**z**	**0**

	0	**1**	
a	**a**	**c**	**0**
b	**b**	**c**	**0**
c	**b**	**a**	**1**

Состояния **a** и **x** не эквивалентны, т.к. имеется различающая цепочка 101:

$a^1 \rightarrow c^0 \rightarrow b^1 \rightarrow c \rightarrow$допускающее
$x^1 \rightarrow y^0 \rightarrow z^1 \rightarrow z \rightarrow$отвергающее

Два состояния конечного автомата начинаются эквивалентными, когда, начав работу из этих состояний, конечный автомат будет допускать одни и те же цепочки.

Если для двух состояний автомата нет различающихся цепочек – эквивалентные состояния.

Проверка эквивалентности двух состояний

	y	z	
0	0	3	0
1	2	5	0
2	2	7	0
3	6	7	0
4	1	6	1
5	6	5	0
6	6	3	1
7	6	3	0

1)Состояния 0, 1, 2 эквивалентны. 2)Состояния 3, 5, 7 эквивалентны. 3)Состояния 4 и 6 не эквивалентны ни друг другу, ни другим состояниям.

Проверка эквивалентности двух состояний

Таблица эквивалентности состояний (0,1):

	y	z	
0	0	3	0
1	2	5	0
2	2	7	0
3	6	7	0
4	1	6	1
5	6	5	0
6	6	3	1
7	6	3	0

	y	z
(0,1)	(0,2)	(3,5)
(0,2)	(0,2)	(3,7)
(3,5)	(6)	(5,7)
(3,7)	(6)	(3,7)
(5,7)	(6)	(3,5)

	y	z
(0,7)	(0,6)	(3)
(0,6)		

Нарушено усл.1 для (0,6)

y – различающая цепочка для пары (0,7)

Если из состояния 0 у нас выходной символ 0 – мы остаемся в y, если из состояния

То есть (0,1); (0,2); (3,5); (3,7); (5,7) эквиваленты. Т.к. эквивалентность транзитивна, то и (1,2) эквивалентно.

Введем следующие обозначения:

0,1,2 объединяем в A,

3,5,7 объединяем в B.

Упрощаем автомат, используя новые обозначения:

	Y	Z	
A	A	B	0
B	6	B	0
4	A	6	1
6	6	B	1

Недостижимые состояния

Пример:

	0	1	
s0	s1	s5	0
s1	s2	s7	1
s2	s2	s5	1
s3	s5	s7	0
s4	s5	s6	0
s5	s3	s1	0
s6	s8	s0	1
s7	s0	s1	1
s8	s3	s6	0

Недостижимые состояния — это состояния, которые не достижимы из начального состояния ни для какой входной цепочки.

(для примера выше - недостижимо c4 – так как его нигде нет. c6 только из c4, c8 только из c6, поэтому c6 и c8 тоже недостижимы)

Получение минимального автомата

Метод разбиения:

P0=({1,2,3,4},{5,6,7}) по допускающим и отвергающим

P1=({1,2},{3,4},{5,6,7}) по входу a

P2=({1,2},{3},{4},{5,6,7}) относительно a

P3=({1,2},{3},{4},{5},{6,7}) относительно a или b.

Состояния внутри каждого подмножества эквивалентны.

	a	**b**	
1	**6**	**3**	**0**
2	**7**	**3**	**0**
3	**1**	**5**	**0**
4	**4**	**6**	**0**
5	**7**	**3**	**1**
6	**4**	**1**	**1**
7	**4**	**2**	**1**

Метод разбиения:

	a	b	
1	6	3	0
2	7	3	0
3	1	5	0
4	4	6	0
5	7	3	1
6	4	1	1
7	4	2	1

Минимальный эквивалентный автомат

	a	b	
{1,2}	{6,7}	{3}	0
{3}	{1,2}	{5}	0
{4}	{4}	{6,7}	0
{5}	{6,7}	{3}	1
{6,7}	{4}	{1,2}	1