ЛЕКЦИЯ 3
“КОНЕЧНЫЕ И МАГАЗИННЫЕ АВТОМАТЫ. ДЕТЕРМИНИРОВАННЫЕ И НЕДЕТЕРМИНИРОВАННЫЕ КОНЕЧНЫЕ АВТОМАТЫ”

ПЛАН

1. Интерпретирующие (конечные и магазинные) автоматы.
2. Связь конечных автоматов и формальных грамматик.
3. Детерминированные и недетерминированные конечные автоматы.
4. Построение ДКА по НКА.

ТЕРМИНОЛОГИЯ

Интерпретирующий автомат ≡ абстрактный автомат [abstract automaton].
Конечный автомат [finite-state-machine]
Магазинный автомат [push-down automaton].

ПОЧЕМУ ИНТЕРПРЕТИРУЮЩИЕ АВТОМАТЫ?

1. Могут решать ряд частных задач трансляции.
2. Моделируются эффективными быстродействующими алгоритмами.
3. Функционируют в ограниченном (фиксированном) кванте памяти.
4. Разработаны солидные теоретические обоснования в виде теорем и алгоритмов, которые позволяют приспособить интерпретирующий автомат для решения тех или иных задач, связанных с синтаксическим анализом и трансляцией.

Конечным автоматом (КА) формально называют совокупность следующих объектов (компонентов):

$$A = (Q, \Sigma, \delta, q_0, F),$$

где Q – конечное множество неструктурируемых состояний автомата;

Σ – непустое множество (алфавит) входных символов (букв или литер);

δ – функция переходов КА, определяемая на декартовом произведении $\boldsymbol{Q \otimes \Sigma}$;

$q_0 \in Q$ – начальное состояние КА;

$F \subset Q$ – множество заключительных состояний КА.

ВИДЫ ОПИСАНИЯ КОНЕЧНОГО АВТОМАТА:

– явное задание $Q, \Sigma, \delta, q_0,$ и F;

– неявное описание автомата в виде графа;

– неявное описание в виде переходной таблицы, неявное описание в виде таблицы (функции) переходов;

– описание в виде регулярного выражения.

Иногда, если это необходимо, описание дополняется таблицей (функцией) выходов

ПРИМЕРЫ ОПИСАНИЯ

Автомат, осуществляющий преобразование из римской системы счисления в десятичную [Алкок Д. Язык Паскаль в иллюстрациях. – М.: Мир, 1991. – С. 60.].

- Явное описание объектов конечного автомата.

$Q = \{A, B, C, D, E, F, S\}$; $\Sigma = \{I, V, X\}$; $q_0 = S$; $F = \{A, B, C, D, E, F\}$;

$\delta = \{\delta(S, I) = A;\ \delta(S, V) = C;\ \delta(S, X) = B;\ \delta(A, I) = E;\ \delta(A, V) = F;\ \delta(A, X) = F;\ \delta(B, V) = C;$
$\delta(B, I) = A;\ \delta(B, X) = B;\ \delta(C, I) = D;\ \delta(D, I) = E;\ \delta(E, I) = \mathrm{F}\}$.

- Описание в виде графа.

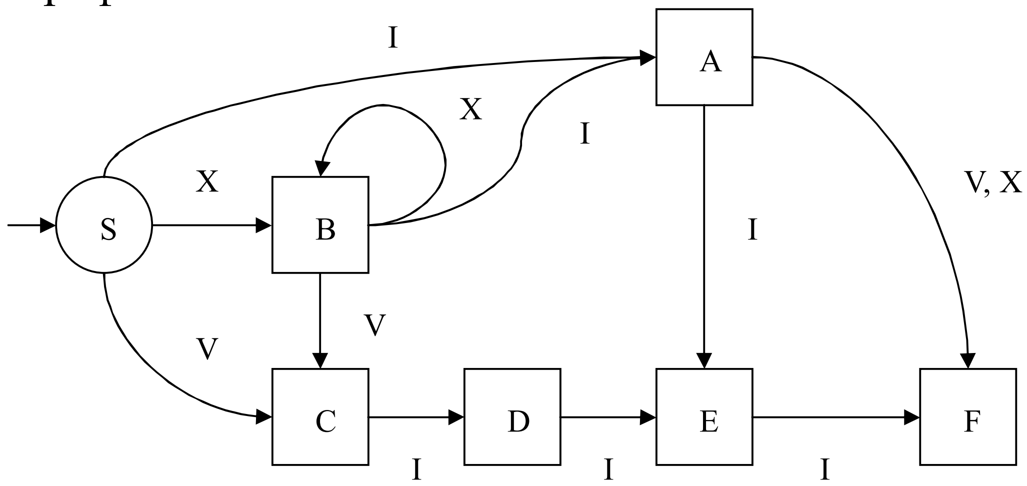


Рисунок 1 – Граф конечного автомата

- Описание в виде переходной таблицы

Состояние	S			A			B			C			D			E			F		
Вход	I	V	X	I	V	X	I	V	X	I	V	X	I	V	X	I	V	X	I	V	X
Выход	A	C	B	E	F	F	A	C	B	D	–	–	E	–	–	F	–	–	–	–	–

- Описание в виде функций переходов и выходов.

Таблица (функция) переходов

Сигнал		н	е	д	п	ш	с	в
Состояние		S	A	B	C	D	E	F
Вход	I	A	E	A	D	E	F	-
	V	C	F	C	–	–	–	–
	X	B	F	B	–	–	–	–

Таблица (функция) выходов

Сигнал		н	е	д	п	ш	с	в
Состояние		S	A	B	C	D	E	F
Вход	I	е	с	е	ш	с	в	о
	V	п	в	п	о	о	о	о
	X	д	в	д	о	о	о	о

Сигналы, помещённые в таблицу: н – начало; е – единица; д –десятка; п – пятёрка; д – шестёрка, с – семёрка; в – восьмерка; о - ошибка.

- Описание в виде регулярного выражения.

$$\mathrm{I} \vee \mathrm{I}(\mathrm{I} \vee \mathrm{II} \vee \mathrm{V} \vee \mathrm{X}) \vee \mathrm{V} \vee \mathrm{V}(\mathrm{I} \vee \mathrm{II} \vee \mathrm{III}) \vee \mathrm{X}\{\mathrm{X}\} \vee \mathrm{X}\{\mathrm{X}\}(\mathrm{I} \vee \mathrm{II} \vee \mathrm{III} \vee \mathrm{IV} \vee \mathrm{V} \vee \mathrm{VI} \vee \mathrm{VII} \vee \mathrm{VIII} \vee \mathrm{IX})$$

Множество цепочек, ***допустимых*** конечным автоматом A, называется ***регулярным*** множеством

$$T(A) = \{\, x |\ x \in \Sigma^{+} \wedge \tilde{\delta}(q_0, x) \cap F \neq \varnothing \},$$

где $\tilde{\delta}(q_0, x) = \tilde{\delta}(q_0, a_1 a_2 ... a_n) = \bigcup\limits_{P} \delta(p, a_n), \qquad n \geq 2$.

Теорема 1. Если $A = (Q, \Sigma, \delta, q_0, F)$ – конечный автомат, то существует праволинейная (регулярная) грамматика $G[S] = (V_T, V_N, S, R)$ такая, что $T(A) = L\{G[S]\}$.

Теорема 2. Если $G[S] = (V_T, V_N, S, R)$ – праволинейная грамматика, то существует конечный автомат $A = (Q, \Sigma, \delta, q_0, F)$ такой, что $L\{G[S]\} = T(A)$.

Алгоритм соответствия праволинейной грамматики и конечного автомата

1. Каждый нетерминал представляется узлом или состоянием на диаграмме.

2. Каждому правилу вида $\boldsymbol{Q}$ ::= t, $t \in V_T$, $\boldsymbol{Q} \in V_N$ соответствует дуга, направленная от начального состояния к состоянию $\boldsymbol{Q}$, помеченная терминальным символом t.

3. Каждому правилу вида $\boldsymbol{Q}$ ::= $\boldsymbol{R}t$, $t \in V_T$, $\boldsymbol{Q}, \boldsymbol{R} \in V_N$ соответствует дуга, направленная от состояния $\boldsymbol{R}$ к состоянию $\boldsymbol{Q}$, помеченная терминальным символом t.

Пример. Праволинейная грамматика $G[F]$, соответствующая КА рисунка 1

<F> ::= <A>V|<A>X|<E>V

<A> ::= <B>I|I

<B> ::= <B>X|X

<C> ::= <B>V|V

<D> ::= <C>I

<E> ::= <A>I|<D>I

Для $G[F]$ аксиома F, $V_T = \{I, V, X\}$, $V_N = \{\mathrm{A, B, C, D, E, F}\}$.

Алгоритм соответствия леволинейной грамматики и конечного автомата

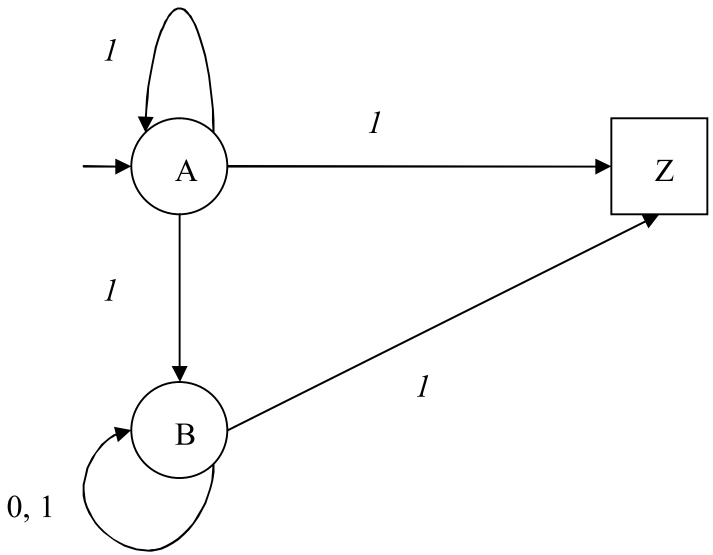
2°. Каждому правилу вида $\boldsymbol{Q}$ ::= t, $t \in V_T$, $\boldsymbol{Q} \in V_N$ соответствует дуга, направленная от состояния $\boldsymbol{Q}$ к конечному состоянию, помеченная терминальным символом t.

3°. Каждому правилу вида $\boldsymbol{Q}$ ::= $t\boldsymbol{R}$, $t \in V_T$, $\boldsymbol{Q}, \boldsymbol{R} \in V_N$ соответствует дуга, направленная от состояния $\boldsymbol{Q}$ к состоянию $\boldsymbol{R}$, помеченная терминальным символом t.

Пример. Леволинейная грамматика.

$$\mathbf{A} \ ::= \ 1\mathbf{A}|1\mathbf{B}|1$$
$$\mathbf{B} \ ::= \ 0\mathbf{B}|1\mathbf{B}|1$$

Для грамматики $G[A]$ аксиома **A**, V_T **=** {0, 1}, V_N = {**A, B**}. Соответствующий граф:



$\delta = \{\delta(A,\ 1) = A;\ \delta(A,\ 1) = B;\ \delta(A,\ 1) = Z;\ \delta(B,\ 0) = B;\ \delta(B,\ 1) = B;\ \delta(B,\ 1) = Z.$

Регулярное выражение: $\{1\}1 \vee \{1\}1\{0 \vee 1\}1$.

Правонилейная грамматика $G[Z]$, эквивалентная исходной $G[A]$:

$$\mathbf{Z} \ ::= \ \mathbf{A}1|\mathbf{B}1|1$$
$$\mathbf{A} \ ::= \ \mathbf{A}1|1$$
$$\mathbf{B} \ ::= \ \mathbf{B}0|\mathbf{B}1|\ \mathbf{A}1|1$$

НЕДЕТЕРМИНИРОВАННЫЕ КОНЕЧНЫЕ АВТОМАТЫ (НКА)

Недетерминированный конечный автомат (НКА) определяется набором объектов

$$\boldsymbol{A = (Q, \Sigma, \delta, S, F)},$$

где $\boldsymbol{Q}$ – конечное множество неструктурируемых состояния автомата;
$\boldsymbol{\Sigma}$ – непустое множество (алфавит) входных символов (букв или литер);
$\boldsymbol{\delta}$ – функция переходов НКА, определяемая на декартовом произведении $\boldsymbol{Q \otimes \Sigma}$;
$\boldsymbol{S \subset Q}$ – множество начальных состояний НКА;
$\boldsymbol{F \subset Q}$ – множество заключительных состояний НКА.

Теорема. Если существует недетерминированный автомат A, то существует детерминированный автомат $A°$, такой что регулярные цепочки, допускаемые недетерминированным автоматом, допускаются детерминированным $T(A) = T(A°)$.

Алгоритм построения ДКА по НКА

1. Всё множество состояний НКА Q разбивается на подмножества Ω_i, каждое из которых определяется ситуацией, когда функция перехода δ определена неоднозначно

$$\Omega_i = \left\{ q_i \in Q : \delta(q_{i-1}, a_i) = \overset{0}{q}_i, \overset{1}{q}_i, \overset{2}{q}_i, \ldots, \overset{r}{q}_i \right\}, \qquad r \geq 1.$$

Подмножество Ω_i образовано состояниями, в которые возможен переход из текущего состояния по символу $\boldsymbol{a_i}$, $\boldsymbol{r}$ – число возможных переходов. Само подмножество $\boldsymbol{\Omega_i}$ будем называть обобщённым (укрупнённым) состоянием КА.

2. Каждое обобщённое состояние $\boldsymbol{\Omega_i}$ анализируется для каждого принадлежащего ему элемента на однозначность функции перехода. При этом возможно возникновение новых обобщённых состояний.

3. Анализ продолжается до тех пор, пока не будет достигнута однозначность перехода между обобщёнными состояниями (классами состояний), отдельными состояниями и классами и отдельными состояниями. В результате получаем множество состояний и функцию перехода ДКА.

4. Ликвидируется (иногда) частичная определённость КА путём введения дополнительных фиктивных состояний и соответствующих им образов функции перехода.

Недостижимое состояние – состояние конечного автомата, которое не может быть достигнуто из начального состояния ни для какой входной цепочки.

Алгоритм поиска множества достижимых состояний

1. Инициализировать множество (список) достижимых состояний, поместив в него начальное состояние конечного автомата.

2. Пополнять множество достигнутых состояний новыми состояниями, которые могут быть достигнуты из состояний, составляющих множество, под воздействием одной входной литеры.

Элементы множества состояний Q конечного автомата, не попавшие во множество достижимых состояний, признаются недостижимыми. Подмножество недостижимых состояний исключается из множества Q.

Пример 1. Рассмотрим КА, приведённый на рисунке выше. Этот автомат, по определению, является недетерминированным.

Шаг 1.

$$\Omega_1 = \{\boldsymbol{\delta(A,\ 1) = A,\ B,\ Z}\};\ \Omega_2 = \{\boldsymbol{\delta(B,\ 1) = B,\ Z}\}.$$

Исследование на однозначность шаг 2 множеств Ω_1 и Ω_2 изобразим с помощью дерева

$\Omega_1 = \{A, B, Z\}$ $\qquad\qquad$ $\Omega_2 = \{B, Z\}$

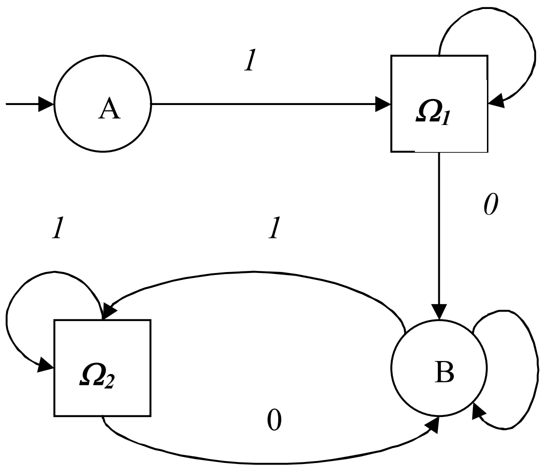
1 $\qquad$ 0 $\qquad\qquad$ 1 $\qquad$ 0

$\Omega_1 = \{A, B, Z\}$ $\qquad$ $\{-, B, -\}$ $\qquad$ $\Omega_2 = \{B, Z\}$ $\qquad$ $\{-, B, -\}$

Получаем:

Вход	**A**	**B**	**Z**	$\boldsymbol{\Omega_1}$	$\boldsymbol{\Omega_2}$
1	$\boldsymbol{\Omega_1}$	$\boldsymbol{\Omega_2}$	–	$\boldsymbol{\Omega_1}$	$\boldsymbol{\Omega_2}$
0	–	**B**	–	**B**	**B**

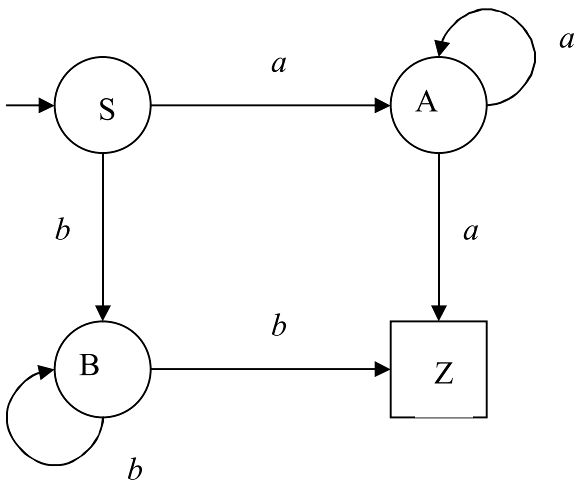
Состояние **Z** является недостижимым



Пример 2. Построить ДКА для регулярной грамматики

$$\begin{aligned} \mathbf{Z} &::= \mathbf{A}a|b\mathbf{B} \\ \mathbf{A} &::= \mathbf{A}a|a \\ \mathbf{B} &::= b\mathbf{B}|b \end{aligned}$$

соответствующий граф.



Таблица, функции переходов:

	S	**A**	**B**	**Z**
a	**A**	**{Z, A}**	–	–
b	**B**	–	**{Z, B}**	–

Введём фиктивное состояние ошибки **E**, на это состояние замкнём все неопределённые в исходной таблице переходы.

	S	A	B	Z	{Z, A}	{Z, B}	E
a	**A**	**{Z, A}**	**E**	**E**	**{Z, A}**	**E**	**E**
b	**B**	**E**	**{Z, B}**	**E**	**E**	**{Z, B}**	**E**

Состояние ***Z*** является недостижимым.

Обозначим {**Z, A**} $\rightarrow$ **C**, {**Z, B**} $\rightarrow$ **D**.

Получим таблицу переходов полностью определённого ДКА, эквивалентного исходному НКА.

	S	A	B	C	D	E
a	**A**	**C**	**E**	**C**	**E**	**E**
b	**B**	**E**	**D**	**E**	**D**	**E**