ФЕДЕРАЛЬНОЕ АГЕНТСТВО ПО ОБРАЗОВАНИЮ
ГОСУДАРСТВЕННОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ПРОФЕССИОНАЛЬНОГО ОБРАЗОВАНИЯ
УФИМСКИЙ ГОСУДАРСТВЕННЫЙ АВИАЦИОННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ

Кафедра ТК

курс лекций по дисциплине

Системное программное обеспечение

Тема: Способы задания регулярных языков

Преподаватель: к.т.н., доцент Карамзина А.Г.

Тема № 10

Способы задания регулярных языков

- Регулярные множества
- Построение КА на основе леволинейной грамматики
- Построение леволинейной грамматики на основе КА
- Свойства регулярных языков

Регулярные языки можно задать с помощью:

- регулярных (*праволинейных и леволинейных*) грамматик $G(VT,VN,P,S)$, $V=VN\cup VT$;
  $A\to\gamma B$ или $A\to\gamma$, где $A,B\in VN$, $\gamma\in VT^*$ $A\to B\gamma$ или $A\to\gamma$, где $A,B\in VN$,
- конечных автоматов (*КА*);
  $M(Q, V, \delta, q$… $F)$;
- регулярных множеств (*и обозначающих* …

Все три способа в равной степени могут быть использованы для определения регулярных языков.

Если над множествами цепочек символов из алфавита V определить операции конкатенации и итерации как:

PQ – конкатенация $P\in V^*$ и $Q\in V^*$: $PQ=\{pq\ \forall p\in P,\ \forall q\in Q\}$;
P^* – итерация $P\in V^*$: $P^*=\{p^n\ \forall p\in P,\ \forall n\in N\}$;

тогда для алфавита V *регулярные множества* определяются рекурсивно:

- $\varnothing$ – регулярное множество;
- $\{\lambda\}$ – регулярное множество;
- $\{a\}$ – регулярное множество $\forall a\in V$;
- если P и Q – произвольные регулярные множества, то множества $P\cup Q$, PQ и P^* также являются регулярными множествами;
- ничто другое не является регулярным множеством.

Регулярные множества – это множества цепочек символов над заданным алфавитом, построенные определенным образом (*с использованием операций объединения, конкатенации и итерации*).

Все три способа определения регулярных языков эквивалентны.

Существуют алгоритмы, которые позволяют для регулярного языка, заданного одним из указанных способов, построить другой способ, определяющий тот же самый язык.

Связь регулярных выражений и регулярных грамматик:

- для любого регулярного языка, заданного регулярным выражением, можно построить регулярную грамматику, определяющую тот же язык;
- для любого регулярного языка, заданного регулярной грамматикой, можно получить регулярное выражение, определяющее тот же язык.

Связь регулярных выражений и конечных автоматов:

- для любого построить КА
- для любого выражение, о

Связь регуля

- на осн эквивалентн
- для зада грамматику.

так как регулярные грамматики используются для определения лексических конструкций языков программирования, то, создав автомат на основе известной грамматики, можно получить распознаватель для лексических конструкций данного языка, то есть решить задачу разбора для лексических конструкций языка, заданных произвольной регулярной грамматикой

Построение КА на основе леволинейной грамматики.

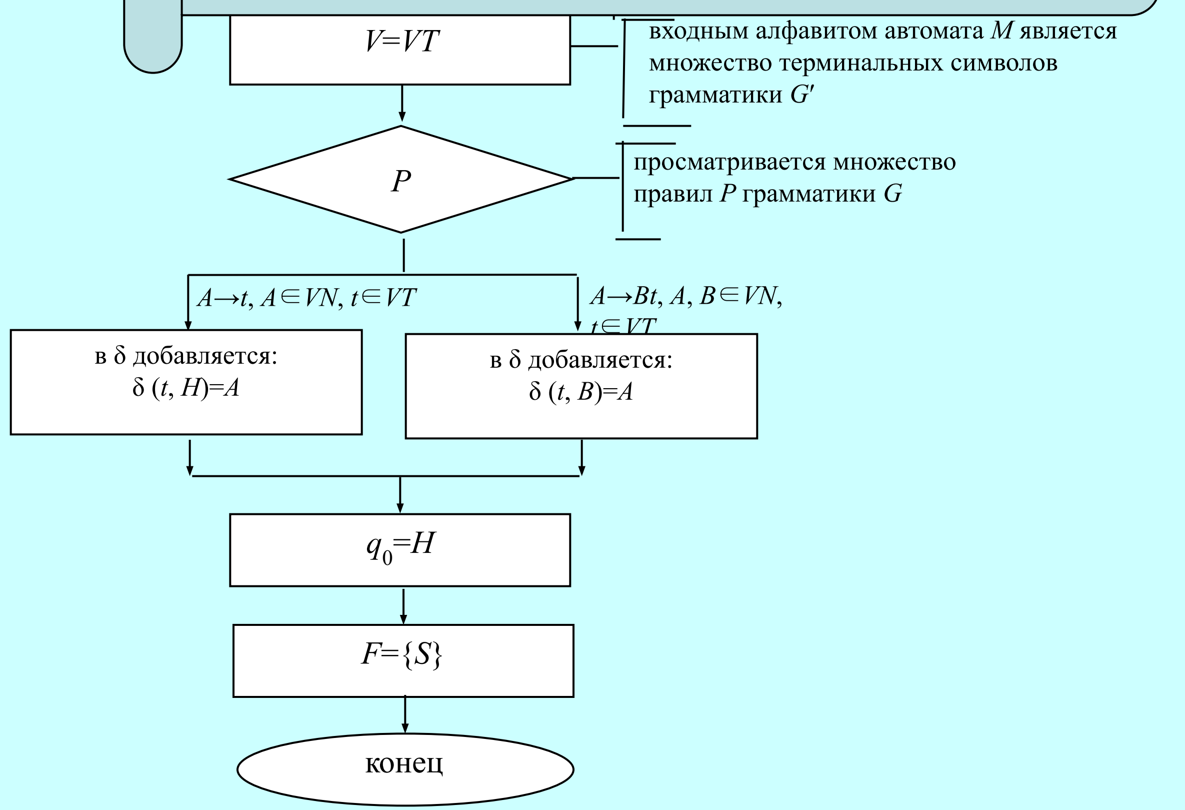
Имеется леволинейная грамматика

G(*VT*, *VN*, *P*, *S*),

необходимо построить эквивалентный ей КА

M(*Q*, *V*, δ, q_0, *F*).

Все языки программирования определяют нотацию записи «<u>слева направо</u>», в той же нотации работают и компиляторы, поэтому далее рассмотрим алгоритмы для <u>леволинейных</u> грамматик.



Пример:

дана регулярная леволинейная грамматика G, необходимо построить КА:

G({"a", "[", "]", "(", ")",";"}, {S,A,B}, P, S)

P:

$S \to S;|A()|Ba)$

$A \to [];|A(|A)|Aa|A;$

$B \to [a]|\ B]|B)|B(|Ba|B[|B);$

Сначала нужно преобразовать ее к автоматному виду:

G'({"a", "[", "]", "(", ")",";"}, {S, S_1, S_2, A, A_1, A_2, B, B_1, B_2, B_3}, P, S)

P':

$S \to S;|\ S_1)|\ S_2)$

$S_1 \to A($

$S_2 \to Ba$

$A \to A_2;|A(|A)|Aa|\ A;$

$A_2 \to A_1]$

$A_1 \to [$

$B \to B_2]\ |\ B]|B)|B(|Ba|B[|\ B_3;$

$B_2 \to B_1 a$

$B_1 \to [$

$B_3 \to B)$

$G'(\{"a", "[", "]", "(", ")", ";"\}, \{S, S_1, S_2, A, A_1, A_2, B, B_1, B_2, B_3\}, P, S)$

P':

$S \rightarrow S; | S_1) | S_2)$

$S_1 \rightarrow A($

$S_2 \rightarrow Ba$

$A \rightarrow A_2; | A(| A) | Aa | A;$

$A_2 \rightarrow A_1]$

$A_1 \rightarrow [$

$B \rightarrow B_2] | B] | B) | B(| Ba | B[| B_3;$

$B_2 \rightarrow B_1 a$

$B_1 \rightarrow [$

$B_3 \rightarrow B)$

Теперь на основе автоматной грамматики строится КА:

- строится множество состояний КА:

$$Q = VN \cup \{H\} = \{S, S_1, S_2, A, A_1, A_2, B, B_1, B_2, B_3, H\};$$

- входной алфавит:

$$V = \{"a", "[", "]", "(", ")", ";"\};$$

$G'(\{"a", "[", "]", "(", ")", ";"\}, \{S, S_1, S_2, A, A_1, A_2, B, B_1, B_2, B_3\}, P, S)$

P':

$S \to S; | S_1) | S_2)$

$S_1 \to A($

$S_2 \to Ba$

$A \to A_2; | A(| A) | Aa | A;$

$A_2 \to A_1]$

$A_1 \to [$

$B \to B_2] | B] | B) | B(| Ba | B[| B_3;$

$B_2 \to B_1 a$

$B_1 \to [$

$B_3 \to B)$

- просматривается множество правил грамматики и строятся функции переходов:

$\delta(';', S)=S$ $\delta(')', S_1)=S$ $\delta(')', S_2)=S$

$\delta('(', A)=S_1$ $\delta('a', B)=S_2$

$\delta(';', A_2)=A$ $\delta('(', A)=A$ $\delta(')', A)=A$ $\delta('a', A)=A$ $\delta(';', A)=A$

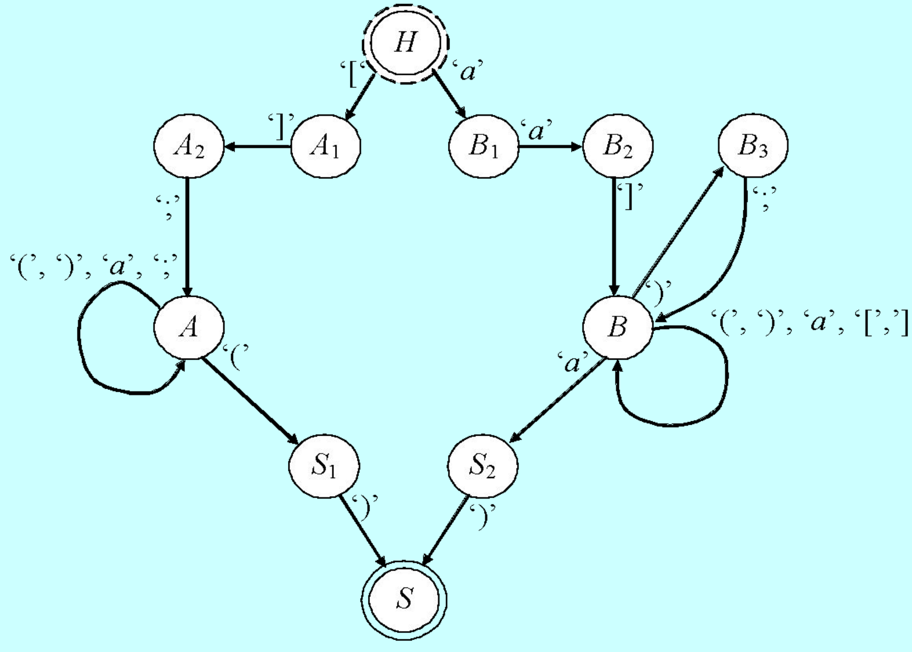
$\delta(']', A_1)=A_2$ $\delta('[', H)=A_1$

$\delta(']', B_2)=B$ $\delta(']', B)=B$ $\delta(')', B)=B$ $\delta('(', B)=B$

$\delta('a', B)=B$ $\delta('[', B)=B$ $\delta(';', B_3)=B$

$\delta('a', B_1)=B_2$ $\delta('[', H)=B_1$ $\delta(')', B)=B_3$

- начальное состояние КА $q_0=H$;
- множество конечных состояний $F=\{S\}$.



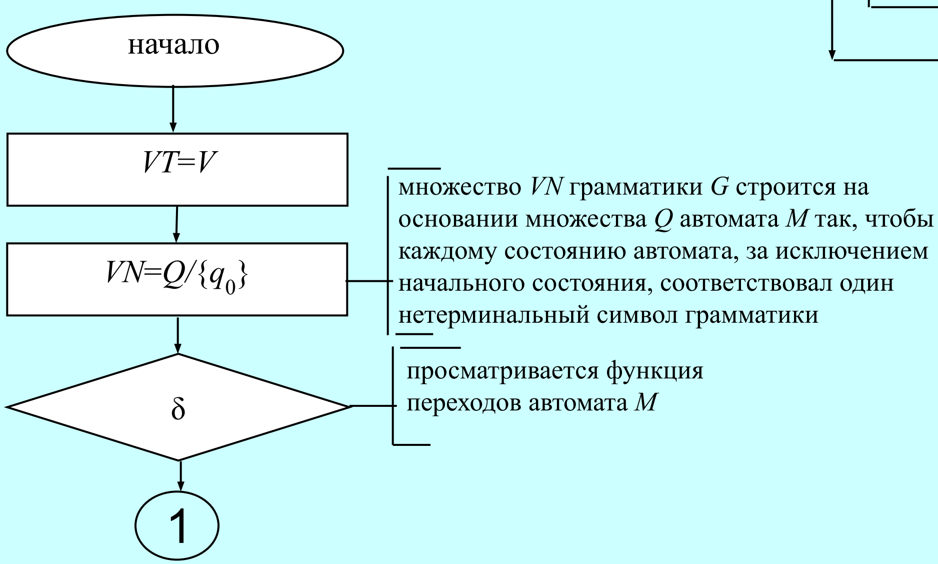
Построение леволинейной грамматики на основе КА.

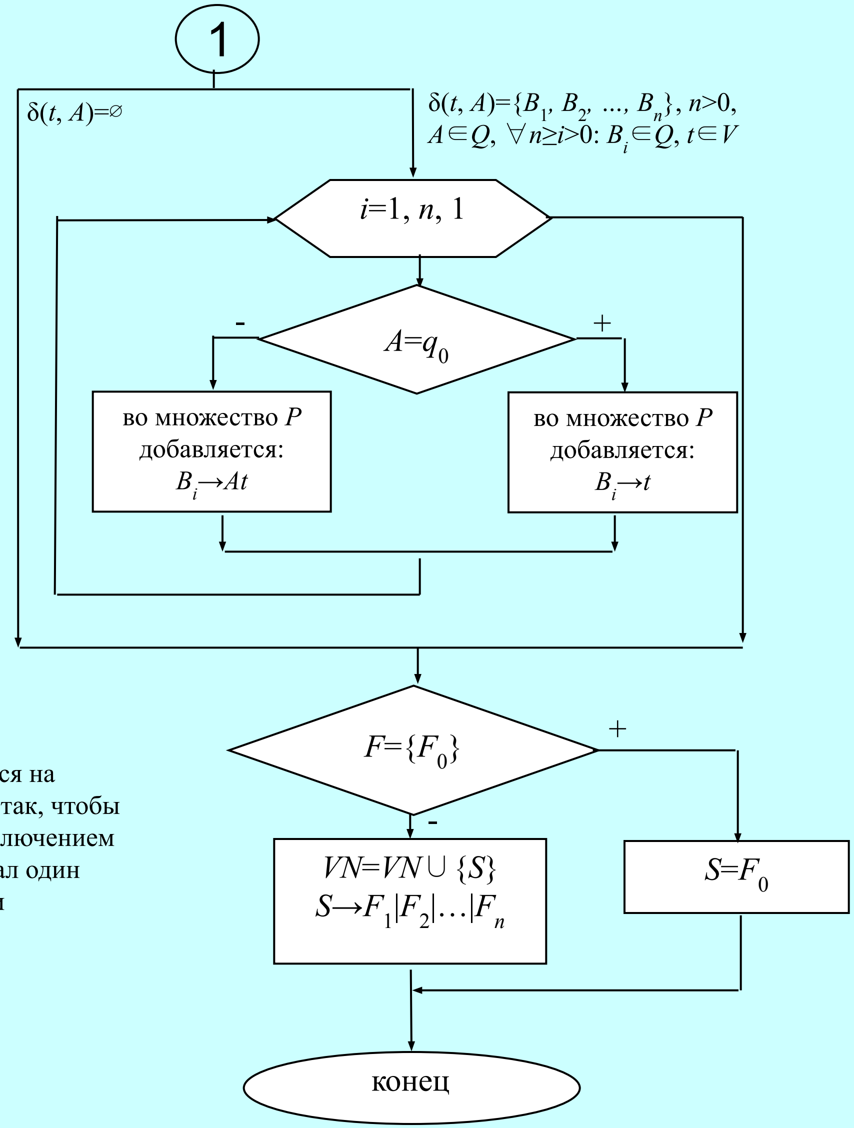
Имеется КА

$M(Q, V, \delta, q_0, F)$,

необходимо построить эквивалентную ему леволинейную грамматику

$G(VT, VN, P, S)$.





Свойства регулярных языков

множество целых чисел замкнуто
относительно операций сложения,
умножения и вычитания, но оно не
замкнуто
относительно операции деления –
при делении двух целых чисел
не всегда получается целое число

Множество называется *замкнутым* относ
результате выполнения этой опера
принадлежащими данному множеств
принадлежащий тому же множеству.

Регулярные языки замкнуты относительно следующих операций:

- пересечения;
- объединения;
- дополнения;
- итерации;
- конкатенации;
- гомоморфизма (*изменения имен символов и подстановки цепочек вместо символов*).

Контрольная работа № 4

Дана регулярная леволинейная грамматика *G*, необходимо построить полностью определенный КА

Вариант 1

G({“.”, +, -, 0, 1},{<знак>, <часть>, <число>}, P, <число>)

P:

<число> → <знак>0 | <знак>1 | <часть>. |<число>0 |<число>1

<часть> → <знак>0 | <знак>1 | <часть>0 | <часть>1

<знак> → + | -

Вариант 2

G({“.”, +, -, 0, 1},{<знак>, <часть>, <осн>, <число>}, P, <число>)

P:

<число> → <часть>. |<осн>0 |<осн>1 | <часть>0 | <часть>1

<осн> → <часть>. | <осн>0 | <осн>1

<часть> → 0 | 1 | <знак>0 | <знак>1 | <часть>0 | <часть>1

<знак> → + | -