

### **Синтаксический анализатор: алгоритмы синтаксического разбора**

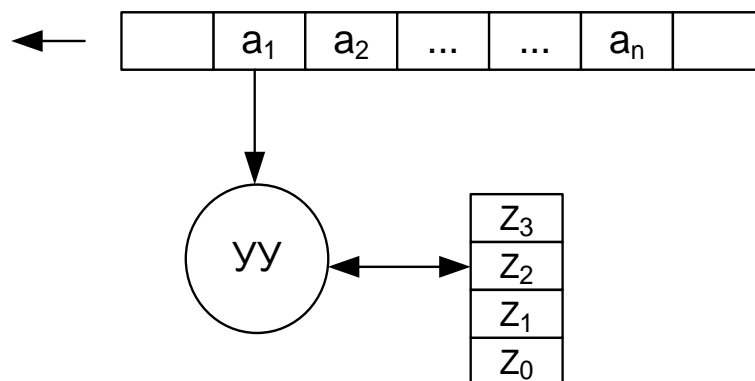
**Цель:** ознакомление с назначением и принципами работы синтаксического анализатора, получение практических навыков построения синтаксического анализатора для заданной грамматики.

**1. Синтаксический анализатор:** часть компилятора, выполняющая синтаксический анализ.

**Вход:** таблица лексем (ТЛ) и таблица идентификаторов (ТИ).

**Выход:** дерево разбора.

**2. Схема работы автомата с магазинной памятью:**



**3. Формальное описание МП-автомата:**

$M = \langle Q, V, Z, \delta, q_0, z_0, F \rangle$   
 $Q$  – множество состояний;  
 $V$  – алфавит входных символов;  
 $Z$  – специальный алфавит магазинных символов;  
 $\delta$  – функция переходов автомата  $Q \times (V \cup \{\lambda\}) \times Z \rightarrow P(Q \times Z^*)$ ,  
где  $P(Q \times Z^*)$  – множество подмножеств  $Q \times Z^*$ ;  
 $q_0 \in Q$  – начальное состояние автомата;  
 $z_0 \in Z$  – начальное состояние магазина (маркер дна);  
 $F \subseteq Q$  – множество конечных состояний.

#### 4. Работа МП-автомата $M = \langle Q, V, Z, \delta, q_0, z_0, F \rangle$

- 1) текущее состояние автомата –  $(q, a\alpha, z\beta)$
- 2) возможны два случая:
  - a. читает символ  $a$ , находящийся под головкой (сдвигает ленту);
  - b. не читает ничего (читает  $\lambda$ , не сдвигает ленту);
- 3) по функции переходов  $\delta$  определяет новое состояние  $q'$ , если  $(q', \gamma) \in \delta(q, a, z)$  или  $(q', \gamma) \in \delta(q, \lambda, z)$ .
- 4) читает верхний символ  $z$  (в магазине) и записывает цепочку  $\gamma$  т.к.  $(q', \gamma) \in \delta(q, a, z)$ , при этом, если  $\gamma = \lambda$ , то верхний символ магазина просто удаляется.
- 5) работа автомата заканчивается  $(q, \lambda, \lambda)$

#### Напоминание:

на каждом шаге автомата возможны три случая:

- 1) функция  $\delta(q, a, z)$  определена – осуществляется переход в новое состояние;
- 2) функция  $\delta(q, a, z)$  не определена, но определена  $\delta(q, \lambda, z)$  – осуществляется переход в новое состояние (лента не продвигается);
- 3) функции  $\delta(q, a, z)$  и  $\delta(q, \lambda, z)$  не определены – дальнейшая работа автомата не возможна (цепочка не разобрана).

По произвольной КС-грамматике  $G_{II} = \langle T, N, P, S \rangle$  всегда можно построить недетерминированный МП-автомат, который допускает цепочки языка, заданного данной грамматикой.

Работа распознавателя:

- 1) если верхний символ магазина (вершина стека) МП-автомата является нетерминальным символом  $A$ , то его можно заменить на цепочку символов  $\alpha$  при условии, что в грамматике языка есть правило  $A \rightarrow \alpha$ , где  $A \in N$ ,  $\alpha \in V^*$ . Считывающая головка автомата при этом не сдвигается (этот шаг работы называется «подбор альтернативы» или выбор правила);
- 2) если верхний символ магазина (вершина стека) является терминальным символом  $a$ , который совпадает с текущим символом входной цепочки, то этот символ выталкивается из стека и считывающая головка передвигается на одну позицию вправо.

5. Дано описание языка:

<i>Компонента</i>	<i>Описание</i>
Символы	Windows-1251
Символы-сепараторы	пробел – допускается везде кроме идентификаторов и ключевых слов; ; (точка с запятой) – разделитель инструкций; { } – программный блок; ( ) – параметры; ( ) – приоритетность операций.
Идентификаторы	только малые латинские буквы, от 1 до 5 букв; идентификатор не может совпадать с ключевыми словами; максимальное количество идентификаторов $2^{16}$
Типы данных	<b>integer</b> – целочисленные данные (четыре байта, значения от $-2^{31}$ до $2^{31}-1$ ), автоматическая инициализация 0, LE; <b>string</b> – строка (любые символы, макс. 255 символов, первый байт – длина строки), автоматическая инициализация строкой длины 0
Операции с данными	+ – бинарный, суммирование, (integer, integer); + – бинарный, конкатенация, (string, string); – – бинарный, вычитание, (integer, integer); * – бинарный, умножение, (integer, integer); / – бинарный, деление, (integer, integer)
Программные конструкции	главная функция (точка входа): main { return <integer-идентификатор> <integer-литерал>; } Функции: <тип данных> function<идентификатор> ( <тип данных> <идентификатор>,...) { return <идентификатор>; } Параметры функции: передаются по значению.
Литералы	<b>числа</b> $2^{31}$ до $2^{31}-1$ , интерпретируются как integer, могут быть только rvalue; <b>строки</b> , символы, заключенные в “(кавычки), могут быть только rvalue
Выражения	арифметические с применением +, -, /, *, (); строковые с применением +,()
...	...

## 6. Пример правильной программы:

```
integer function fi(integer x, integer y)
{
  declare integer z;
  z= x*(x+y);
  return z;
}
string function fs (string a, string b)
{
  declare string c;
  declare string function substr(string a, integer p,
                                integer n);
  c = substr(a, 1,3)+ b;
  return c;
};
main
{
  declare integer x;
  declare integer y;
  declare integer z;
  declare string sa;
  declare string sb;
  declare string sc;
  declare integer function strlen(string p);
  x = 1;
  y = 5;
  sa = '1234567890';
  sb = '1234567890';
  z = fi(x,y);
  sc = fs(sa,sb);
  print 'контрольный пример';
  print z;
  print sc;
  print strlen(sc);
  return 0;
};
```

## 7. Лексемы:

<i>конструкция</i>	<i>лексема</i>	<i>примечание</i>
integer string	t	ТИ: integer или string, значение по умолчанию: для integer – нуль, для string – пустая строка
идентификатор	i	ТИ: строка идентификатора, усеченная до 5 символов. Префикс: имя конструкции
литералы	l	integer или string, значение.
function	f	
declare	d	
return	r	
print	p	
main	m	
;	;	
,	,	
{	{	
}	}	
(	(	
)	)	
=	=	
+	v	
-		
*		
/		

## 8. Лексический анализатор:

- **убрать все лишние пробелы:**
  - подстроки, состоящие из более, чем из одного пробела заменить на один пробел;
  - пробельные префиксы и суффиксы для символов `;,}{()=+–/*;`;
  - ввести специальный символ для подсчета номера строки `|`.
- **построить регулярные выражения для лексем:**
  - например, для ключевого слова `main` регулярное выражение `→ main`
- **выполнить распознавание лексем:**
  - распознавателем регулярного языка является конечный автомат
- **построить таблицу лексем и таблицу идентификаторов:**
  - на выходе лексический анализатор формирует ТЛ и ТИ
- **при неуспешном распознавании или обнаружении некоторых ошибок во входном тексте выдать сообщение об ошибке.**

Результат лексического разбора (таблица лексем):

<i>Вход лексического анализатора</i>	<i>Выход (таблица лексем)</i>	<i>Дополнительная информация (таблица идентификаторов)</i>
integer	t	
function	f	
fi	i	fi – идентификатор функции, integer
(	(	
integer	t	
x	i	fix – имя, параметр, integer
,	,	
integer	t	
y	i	fiy– имя, параметр integer
)	)	
{	{	
declare	d	
integer	t	
z	i	fiz – имя, integer, значение: 0
;	;	
z	i	указатель на fiz
=	=	
x	i	указатель на fix
*	v	*
(	(	

x	i	указатель на fix
+	v	+
y	i	указатель на fiy
)	)	
;	;	
return	r	
z	i	указатель на fiz
;	;	
}	}	
string	t	
function	f	
fs	i	fs –идентификатор функции, string
(	(	
string	t	
a	i	fsa – имя,параметр string
,	,	
string	t	
b	i	fsb – имя, параметр string
)		
{		
declare	d	
string	t	
c	i	fsc – имя, string, значение: пустая строка
;		
declare	d	
string	t	
function	f	
substr	i	substr – идентификатор функции, string
(		
string	t	
a	i	substra – имя, параметр string
,	,	
integer	t	
p	i	substrp – имя, параметр integer
,	,	
integer	t	
n	i	substrn – имя, параметр integer
)	)	
;		
c	i	указатель на fsc

=	=	
substr	i	указатель на substr
(	(	
a	i	указатель на fsa
,		
1	l	L01 – литерал, integer, значение:1
,	,	
3	l	L02 – литерал, integer, значение:3
)		
+	v	+
b	i	указатель на fsb
;		
return	r	указатель на fsc
c	i	
;	;	
}	}	
main	m	
{	{	
declare	d	
integer	t	
x	i	mainx – имя, integer, значение: 0
;	;	
declare	d	
integer	t	
y	i	mainy – имя, integer, значение: 0
;		
declare	d	
integer	t	
z	i	mainz – имя, integer, значение: 0
declare	d	
string	t	
sa	i	mainsa – имя, string, значение: пустая строка
;	;	
declare	d	
string	t	
sb	i	mainsb – имя, string, значение: пустая строка
;	;	
declare	d	
string	t	
sc	i	mainsc – имя, string, значение: пустая строка



;	;	
declare	d	
integer	t	
function	f	
strlen	i	strlen – идентификатор функции, integer
(		
string	t	
p	i	strlenp – имя, параметр string
)		
;	;	
x	i	указатель на mainx
=	=	
l	l	указатель на L01
;		
y	i	указатель на mainy
=	=	
5	l	L03 – литерал, integer, значение:5
;		
sa	i	указатель на mainsa
=	=	
'1234567890'		L03 – литерал, string, значение: [10]1234567890
;		
sb	i	указатель на mainsb
=	=	
'1234567890'	l	указатель на L03
;	;	
z	i	
=	=	
fi	i	указатель на fi
(	(	
x	i	указатель на mainx
,	,	
y	i	указатель на mainy
)	)	
;	;	
sc	i	указатель на mainsc
=	=	
fc	i	указатель на fc
(	(	
sa	i	указатель на mainsa
,	,	

sb	i	указатель на mainsb
)	)	
;	;	
print	p	
'контрольный пример'	l	L04 – литерал, string, значение: [17]контрольный пример
;	;	
print	p	
z	i	указатель на mainz
;	;	
print	p	
sc	i	указатель на mainc
;	;	
print	p	
strlen	i	указатель на strlen
(	(	
sc	i	указатель на mainsc
)	)	
;	;	
return	r	
0	l	L05 – литерал, integer, значение:0
;	;	
}	}	
;	;	

## 9. Синтаксический анализатор

Грамматики типа 2 иерархии Хомского:

$G_{II} = \langle T, N, P, S \rangle$  – контекстно-свободные грамматики.

Правила имеют вид:  $A \rightarrow \alpha$ , где  $A \in N$ ,  $\alpha \in V^*$ .

**Грамматика:**

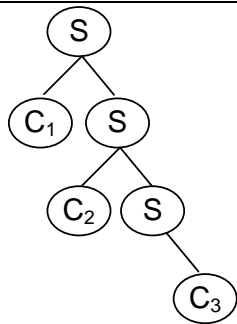
<i>Терминалы</i>	<i>Назначение</i>
t	тип данных
i	идентификатор
l	литерал
f	function
d	declare
r	return
p	print
m	main
;	;
,	,
{	{
}	}
(	(
)	)
v	+ – * /

**Правила грамматики:**

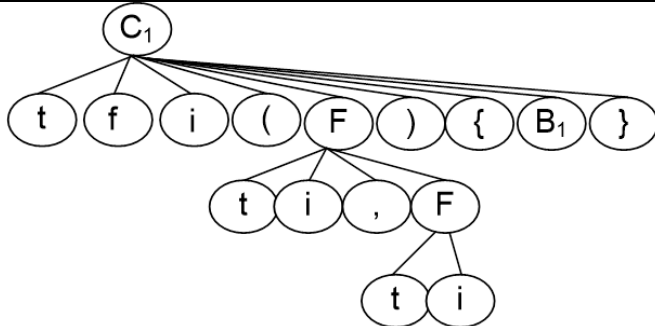
$S \rightarrow C; | CS;$   
 $C \rightarrow t f i(F) \{ B \}; | m \{ B \};$   
 $B \rightarrow N r E;$   
 $N \rightarrow O | ON$   
 $O \rightarrow d t i; | r E; | i = E; | d t f i(F);$   
 $E \rightarrow i | l | (E) | E v E | i(W)$   
 $F \rightarrow t i | t i, F$   
 $W \rightarrow i | l | i, W | l, W$

**Назначение нетерминалов:**

Стартовый символ (2 альтернативы)  
 Программная конструкция  
 Тело программной конструкции  
 Последовательность операторов  
 Оператор программы  
 Выражение  
 Параметры функции  
 Подвыражение



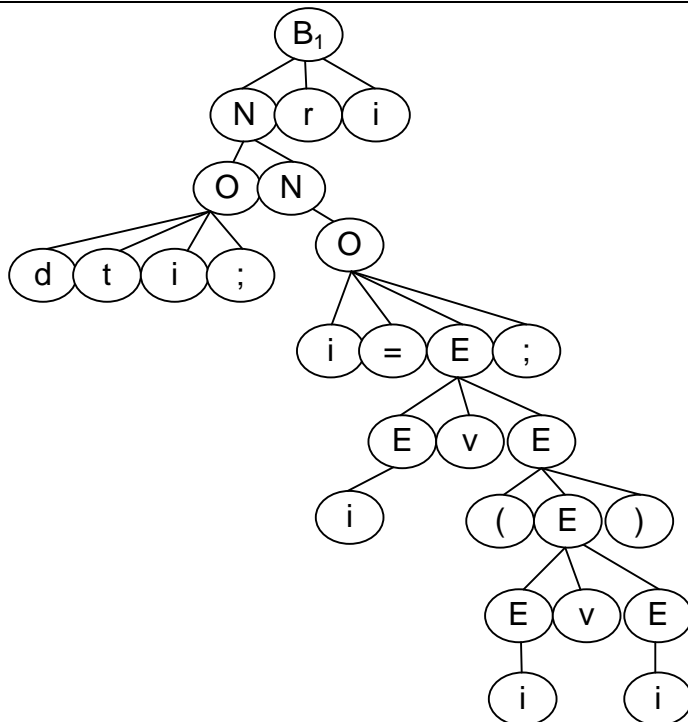
$S \rightarrow CS$ ; (второе правило для  $S$ )



```

integer function fi(integer x, integer y)
{
  declare integer z;
  z= x*(x+y);
  return z;
}
  
```

(первое правило для  $C \rightarrow tfi(F)\{B\};$ )  
 (для двух параметров функции применяется второе правило  $F \rightarrow ti, F$ , затем первое  $F \rightarrow ti$ )



```

{
  declare integer z;
  z= x*(x+y);
  return z;
}
  
```

(единственное правило для  $B \rightarrow NrE$ ;)   
 (второе правило для  $N \rightarrow ON$ )   
 (первое правило для  $O \rightarrow dti$ ;)   
 (первое правило для  $N \rightarrow O$ )   
 (третье правило для  $O \rightarrow i=E$ ;)   
 (первое правило для  $E \rightarrow i$ )   
 (третье правило для  $E \rightarrow EvE$ )   
 (первое правило для  $E \rightarrow i$ )   
 (первое правило для  $E \rightarrow i$ )

## 10. Семантика компилятора:

№	Правило
1	Наличие функции main
2	Усечение слишком длинных идентификаторов до 5 символов
3	Сначала осуществляется проверка на ключевые слова, а затем на идентификатор. Не допускаются идентификаторы, совпадающие с ключевыми словами
4	Нет повторяющихся наименований функций
5	Нет повторяющихся объявлений идентификаторов
6	Предварительное объявление, применяемых функций
7	Предварительное объявление, применяемых идентификаторов.
8	Соответствие типов формальных и фактических параметров при вызове функций
9	Усечение слишком длинного значения string-литерала
10	Округление слишком большого значения integer-литерала
11	Если ошибка возникает на этапе лексического анализа, синтаксический анализ не выполняется
12	При возникновении ошибки в процессе лексического анализа, ошибочная фраза игнорируется (предполагается, что ее нет) и осуществляется попытка разбора следующей фразы. Граница фразы, любой сепаратор (пробел, скобка, запятая, точка с запятой и пр.)
13	Если 3 подряд фразы не разобраны, то работа транслятора останавливается
14	При возникновении ошибки в процессе синтаксического анализа, ошибочная фраза игнорируется (предполагается, что ее нет) и осуществляется попытка разбора следующей фразы. Граница фразы – точка с запятой.

## 11. Построение МП-автомата $M = \langle Q, V, Z, \delta, q_0, z_0, F \rangle$

Пусть  $G = \langle T, N, P, S \rangle$  – контекстно-свободная грамматика.

Магазинный автомат  $M = \langle Q, V, Z, \delta, q_0, z_0, F \rangle$ :

$$Q = \{q_0\}, V = T, F = \{q_0\}, Z = T \cup N \cup \{z_0\}$$

$$\forall A: (A \rightarrow \alpha) \in P \Rightarrow \delta(q_0, \lambda, A) = (q_0, \alpha^R)$$

$$\forall a \in T \Rightarrow \delta(q_0, a, a) = (q_0, \lambda)$$

$$\delta(q_0, \lambda, z_0) = (q_0, \lambda)$$

Стартовая конфигурация МП-автомата:  $(q_0, w, z_0 S)$

12. Пример:  $G = \langle T, N, P, S \rangle$

$S \rightarrow C | CS$

$C \rightarrow \text{tfi}(F)\{B\}; | m\{B\};$

$B \rightarrow \text{NrE};$

$N \rightarrow O | ON$

$O \rightarrow \text{dti}; | rE; | i=E; | \text{dtfi}(F);$

$E \rightarrow i | l | (E) | EvE | i(W)$

$F \rightarrow ti | ti, F$

$W \rightarrow i | l | i, W | l, W$

а) Определим компоненты МП-автомата:

$$Q = \{q_0\}, V = \{d, f, i, l, r, t, v, \backslash\}, \backslash\{, \backslash\}, \backslash(, :, =\}$$

$$Z = V \cup \{S, C, B, N, O, E, F, W\} \cup \{q_0\}$$

б) Функция переходов  $\forall A : (A \rightarrow \alpha) \in P \Rightarrow \delta(q_0, \lambda, A) = (q_0, \alpha^R)$

аргументы	$\alpha^R$	$\alpha^R$	$\alpha^R$	$\alpha^R$	$\alpha^R$
$q_0, \lambda, S$	C	SC			
$q_0, \lambda, C$	; } B { ) F ( i f t	; } B { m			
$q_0, \lambda, B$	; E r N				
$q_0, \lambda, N$	O	NO			
$q_0, \lambda, O$	; i t d	; E r	; E = i	; ) F ( i f t d	
$q_0, \lambda, E$	i	l	) E (	E v E	) W ( i
$q_0, \lambda, F$	i t	F, i t			
$q_0, \lambda, W$	i	l	W, i	W, l	

Стек – список элементов, организованных по принципу LIFO.

Обозначения:

$S$  – стартовый символ.

$z_0$  – символ дна стека (в программе можно использовать знак \$).

Конец входной ленты обозначим символом, совпадающим с символом дна стека \$ (можно использовать как признак завершения разбора).

с) Функция переходов  $\forall a \in T \Rightarrow \delta(q_0, a, a) = (q_0, \lambda)$

(на вершине стека находится терминальный символ, который совпадает с текущим символом входной цепочки. Этот символ выталкивается из стека, считывающая головка передвигается на одну позицию вправо)

аргументы	Значение
$q_0, d, d$	$q_0, \lambda$
$q_0, f, f$	
$q_0, i, i$	
$q_0, r, r$	
$q_0, t, t$	
$q_0, \}, \}$	
$q_0, \{, \{$	
$q_0, (, ($	
$q_0, ), )$	
$q_0, :, :$	
$q_0, ;, ;$	

д) Функция переходов  $\delta(q_0, \lambda, z_0) = (q_0, \lambda)$

$z_0$  - начальное состояние (маркер дна)

е) Работа магазинного автомата.

1	начальное состояние МП-автомата
Лента	$t\bar{f}i(t_i, t_i) \{ dt_i; i = iv(iv_i); ri; \} t\bar{f}i(t_i, t_i) \{ dt_i; dt\bar{f}i(t_i, t_i, t_i); i = i(i, l, l)vi; ri; \}; m \{ dt_i; dt_i; dt_i; dt_i; dt_i; dt_i; dt\bar{f}i(t_i); i = i; i = l; i = l; i = l; i = i(i, i); i = i(i, i); pl; pi; pi; pi(i); rl; \};$
Магазин	$Z_0S$

1.1	подбор подходящего правила грамматики по нетерминалу
Лента (не продвигается)	$t\bar{f}i(t_i, t_i) \{ dt_i; i = iv(iv_i); ri; \} t\bar{f}i(t_i, t_i) \{ dt_i; dt\bar{f}i(t_i, t_i, t_i); i = i(i, l, l)vi; ri; \}; m \{ dt_i; dt_i; dt_i; dt_i; dt_i; dt_i; dt\bar{f}i(t_i); i = i; i = l; i = l; i = l; i = i(i, i); i = i(i, i); pl; pi; pi; pi(i); rl; \};$
Магазин	$Z_0C$ (1-е правило для $S \rightarrow C$ )

На каждом шаге работы МП-автомата сохраняется его состояние (позиция на ленте, состояние магазина, номер правила)

1.1.1	
Лента	$tfi(ti,ti)\{dti;i=iv(ivi);ri;\}tfi(ti,ti)\{dti;dtfi(ti,ti,ti);i=i(i,l,l)vi;ri;\};m\{dti;dti;dti;dti;dti;dti;dtfi(ti);i=i;i=l;i=l;i=l;i=i(i,i);i=i(i,i);pl;pi;pi;pi(i);rl;\};$
Магазин	$Z_0;\}B\{)F(ift \text{ (1-е правило для } C \rightarrow tfi(F)\{B\};)$
Лента	$ti,ti)\{dti;i=iv(ivi);ri;\}tfi(ti,ti)\{dti;dtfi(ti,ti,ti);i=i(i,l,l)vi;ri;\};m\{dti;dti;dti;dti;dti;dti;dtfi(ti);i=i;i=l;i=l;i=l;i=i(i,i);i=i(i,i);pl;pi;pi;pi(i);rl;\};$
Магазин	$Z_0;\}B\{)F$

1.1.1.1	
Лента	$ti,ti)\{dti;i=iv(ivi);ri;\}tfi(ti,ti)\{dti;dtfi(ti,ti,ti);i=i(i,l,l)vi;ri;\};m\{dti;dti;dti;dti;dti;dti;dtfi(ti);i=i;i=l;i=l;i=l;i=i(i,i);i=i(i,i);pl;pi;pi;pi(i);rl;\};$
Магазин	$Z_0;\}B\{)it \text{ (1-е правило для } F \rightarrow ti)$
Лента	$ti,ti)\{dti;i=iv(ivi);ri;\}tfi(ti,ti)\{dti;dtfi(ti,ti,ti);i=i(i,l,l)vi;ri;\};m\{dti;dti;dti;dti;dti;dti;dtfi(ti);i=i;i=l;i=l;i=l;i=i(i,i);i=i(i,i);pl;pi;pi;pi(i);rl;\};$
Магазин	$Z_0;\}B\{)it$
Лента	$,ti)\{dti;i=iv(ivi);ri;\}tfi(ti,ti)\{dti;dtfi(ti,ti,ti);i=i(i,l,l)vi;ri;\};m\{dti;dti;dti;dti;dti;dti;dtfi(ti);i=i;i=l;i=l;i=l;i=i(i,i);i=i(i,i);pl;pi;pi;pi(i);rl;\};$
Магазин	$Z_0;\}B\{)$
<b>Автомат остановлен</b>	

1.1.1.2 возврат к сохраненному сост. 1.1.1.1, выбор другого правила для F	
Лента	$ti,ti)\{dti;i=iv(ivi);ri;\}tfi(ti,ti)\{dti;dtfi(ti,ti,ti);i=i(i,l,l)vi;ri;\};m\{dti;dti;dti;dti;dti;dti;dtfi(ti);i=i;i=l;i=l;i=l;i=i(i,i);i=i(i,i);pl;pi;pi;pi(i);rl;\};$
Магазин	$Z_0;\}B\{)F,it \text{ (2-е правило для } F \rightarrow ti,F)$
Лента	$ti)\{dti;i=iv(ivi);ri;\}tfi(ti,ti)\{dti;dtfi(ti,ti,ti);i=i(i,l,l)vi;ri;\};m\{dti;dti;dti;dti;dti;dti;dtfi(ti);i=i;i=l;i=l;i=l;i=i(i,i);i=i(i,i);pl;pi;pi;pi(i);rl;\};$
Магазин	$Z_0;\}B\{)F$

1.1.1.2.1	
Лента	$ti)\{dti;i=iv(ivi);ri;\}tfi(ti,ti)\{dti;dtfi(ti,ti,ti);i=i(i,l,l)vi;ri;\};m\{dti;dti;dti;dti;dti;dti;dtfi(ti);i=i;i=l;i=l;i=l;i=i(i,i);i=i(i,i);pl;pi;pi;pi(i);rl;\};$
Магазин	$Z_0;\}B\{)it \text{ (1-е правило для } F \rightarrow ti)$
Лента	$dti;i=iv(ivi);ri;\}tfi(ti,ti)\{dti;dtfi(ti,ti,ti);i=i(i,l,l)vi;ri;\};m\{dti;dti;dti;dti;dti;dti;dtfi(ti);i=i;i=l;i=l;i=l;i=i(i,i);i=i(i,i);pl;pi;pi;pi(i);rl;\};$
Магазин	$Z_0;\}B$



1.1.1.2.2	
Лента	$dti; i=iv(ivi); ri; \} tfi(ti, ti) \{ dti; dtfi(ti, ti, ti); i=i(i, l, l)vi; ri; \}; m \{ dti; dti; dti; dti; dti; dti; dtfi(ti); i=i; i=l; i=l; i=l; i=i(i, i); i=i(i, i); pl; pi; pi; pi(i); rl; \};$
Магазин	$Z_0; \}; ErN$ (правило для $B \rightarrow NrE;$ )
Лента	$dti; i=iv(ivi); ri; \} tfi(ti, ti) \{ dti; dtfi(ti, ti, ti); i=i(i, l, l)vi; ri; \}; m \{ dti; dti; dti; dti; dti; dti; dtfi(ti); i=i; i=l; i=l; i=l; i=i(i, i); i=i(i, i); pl; pi; pi; pi(i); rl; \};$
Магазин	$Z_0; \}; ErO$ (1-е правило для $N \rightarrow O$ )
Лента	$dti; i=iv(ivi); ri; \} tfi(ti, ti) \{ dti; dtfi(ti, ti, ti); i=i(i, l, l)vi; ri; \}; m \{ dti; dti; dti; dti; dti; dti; dtfi(ti); i=i; i=l; i=l; i=l; i=i(i, i); i=i(i, i); pl; pi; pi; pi(i); rl; \};$
Магазин	$Z_0; \}; ErNO$ (2-е правило для $N \rightarrow ON$ )
Лента	$dti; i=iv(ivi); ri; \} tfi(ti, ti) \{ dti; dtfi(ti, ti, ti); i=i(i, l, l)vi; ri; \}; m \{ dti; dti; dti; dti; dti; dti; dtfi(ti); i=i; i=l; i=l; i=l; i=i(i, i); i=i(i, i); pl; pi; pi; pi(i); rl; \};$
Магазин	$Z_0; \}; ErN; itd$ (1-е правило для $O \rightarrow dti;$ )
Лента	$i=iv(ivi); ri; \} tfi(ti, ti) \{ dti; dtfi(ti, ti, ti); i=i(i, l, l)vi; ri; \}; m \{ dti; dti; dti; dti; dti; dti; dtfi(ti); i=i; i=l; i=l; i=l; i=i(i, i); i=i(i, i); pl; pi; pi; pi(i); rl; \};$
Магазин	$Z_0; \}; ErN$
Лента	$i=iv(ivi); ri; \} tfi(ti, ti) \{ dti; dtfi(ti, ti, ti); i=i(i, l, l)vi; ri; \}; m \{ dti; dti; dti; dti; dti; dti; dtfi(ti); i=i; i=l; i=l; i=l; i=i(i, i); i=i(i, i); pl; pi; pi; pi(i); rl; \};$
Магазин	$Z_0; \}; ErO$ (1-е правило для $N \rightarrow O$ )
Лента	$i=iv(ivi); ri; \} tfi(ti, ti) \{ dti; dtfi(ti, ti, ti); i=i(i, l, l)vi; ri; \}; m \{ dti; dti; dti; dti; dti; dti; dtfi(ti); i=i; i=l; i=l; i=l; i=i(i, i); i=i(i, i); pl; pi; pi; pi(i); rl; \};$
Магазин	$Z_0; \}; Er; itd$ (1-е правило для $O \rightarrow dti;$ )
<b>Автомат остановлен</b>	

1.1.2	
Лента	$tfi(ti, ti) \{ dti; i=iv(ivi); ri; \} tfi(ti, ti) \{ dti; dtfi(ti, ti, ti); i=i(i, l, l)vi; ri; \}; m \{ dti; dti; dti; dti; dti; dti; dtfi(ti); i=i; i=l; i=l; i=l; i=i(i, i); i=i(i, i); pl; pi; pi; pi(i); rl; \};$
Магазин	$Z_0; \} B \{ m$ (2-е правило для $C \rightarrow m \{ B \};$ )
<b>Автомат остановлен</b>	

1.2	
Лента	$tfi(ti, ti) \{ dti; i=iv(ivi); ri; \} tfi(ti, ti) \{ dti; dtfi(ti, ti, ti); i=i(i, l, l)vi; ri; \}; m \{ dti; dti; dti; dti; dti; dti; dtfi(ti); i=i; i=l; i=l; i=l; i=i(i, i); i=i(i, i); pl; pi; pi; pi(i); rl; \};$
Магазин	$Z_0SC$

1.2.1	
Лента	$t_{fi}(t_i, t_i) \{ dt_i; i=iv(iv_i); ri; \} t_{fi}(t_i, t_i) \{ dt_i; dt_{fi}(t_i, t_i, t_i); i=i(i, l, l)vi; ri; \}; m \{ dt_i; dt_i; dt_i; dt_i; dt_i; dt_i; dt_{fi}(t_i); i=i; i=l; i=l; i=l; i=i(i, i); i=i(i, i); pl; pi; pi; pi(i); rl; \};$
Магазин	$Z_0S; \} B \{ ) F(ift \text{ (1-е правило для } C \rightarrow t_{fi}(F) \{ B \}; )$
Лента	$t_i, t_i) \{ dt_i; i=iv(iv_i); ri; \} t_{fi}(t_i, t_i) \{ dt_i; dt_{fi}(t_i, t_i, t_i); i=i(i, l, l)vi; ri; \}; m \{ dt_i; dt_i; dt_i; dt_i; dt_i; dt_i; dt_{fi}(t_i); i=i; i=l; i=l; i=l; i=i(i, i); i=i(i, i); pl; pi; pi; pi(i); rl; \};$
Магазин	и т.д.

### Правильная траектория:

	$t_{fi}(t_i, t_i) \{ dt_i; i=iv(iv_i); ri; \} t_{fi}(t_i, t_i) \{ dt_i; dt_{fi}(t_i, t_i, t_i); i=i(i, l, l)vi; ri; \}; m \{ dt_i; dt_i; dt_i; dt_i; dt_i; dt_i; dt_{fi}(t_i); i=i; i=l; i=l; i=l; i=i(i, i); i=i(i, i); pl; pi; pi; pi(i); rl; \};$
Магазин	$Z_0S$
Магазин	$Z_0SC$
Магазин	$Z_0S \} B \{ ) F(ift$
Лента	$t_i, t_i) \{ dt_i; i=iv(iv_i); ri; \} t_{fi}(t_i, t_i) \{ dt_i; dt_{fi}(t_i, t_i, t_i); i=i(i, l, l)vi; ri; \}; m \{ dt_i; dt_i; dt_i; dt_i; dt_i; dt_i; dt_{fi}(t_i); i=i; i=l; i=l; i=l; i=i(i, i); i=i(i, i); pl; pi; pi; pi(i); rl; \};$
Магазин	$Z_0S \} B \{ ) F$
Магазин	$Z_0S \} B \{ ) F, t_i$
Лента	$t_i) \{ dt_i; i=iv(iv_i); ri; \} t_{fi}(t_i, t_i) \{ dt_i; dt_{fi}(t_i, t_i, t_i); i=i(i, l, l)vi; ri; \}; m \{ dt_i; dt_i; dt_i; dt_i; dt_i; dt_i; dt_{fi}(t_i); i=i; i=l; i=l; i=l; i=i(i, i); i=i(i, i); pl; pi; pi; pi(i); rl; \};$
Магазин	$Z_0S \} B \{ ) F$
Магазин	$Z_0S \} B \{ ) it$
Лента	$dt_i; i=iv(iv_i); ri; \} t_{fi}(t_i, t_i) \{ dt_i; dt_{fi}(t_i, t_i, t_i); i=i(i, l, l)vi; ri; \}; m \{ dt_i; dt_i; dt_i; dt_i; dt_i; dt_i; dt_{fi}(t_i); i=i; i=l; i=l; i=l; i=i(i, i); i=i(i, i); pl; pi; pi; pi(i); rl; \};$
Магазин	$Z_0S \} B$
Магазин	$Z_0S \}; ErN$
Магазин	$Z_0S \}; ErNO$
Магазин	$Z_0S \}; ErN; itd \text{ (1-е правило для } O \rightarrow dt_i; )$
Лента	$i=iv(iv_i); ri; \} t_{fi}(t_i, t_i) \{ dt_i; dt_{fi}(t_i, t_i, t_i); i=i(i, l, l)vi; ri; \}; m \{ dt_i; dt_i; dt_i; dt_i; dt_i; dt_i; dt_{fi}(t_i); i=i; i=l; i=l; i=l; i=i(i, i); i=i(i, i); pl; pi; pi; pi(i); rl; \};$
Магазин	$Z_0S \}; ErN$
Магазин	$Z_0S \}; ErO \text{ (1-е правило для } N \rightarrow O)$
Магазин	$Z_0S \}; Er; E=i \text{ (1-е правило для } O \rightarrow i=E; )$
Лента	$iv(iv_i); ri; \} t_{fi}(t_i, t_i) \{ dt_i; dt_{fi}(t_i, t_i, t_i); i=i(i, l, l)vi; ri; \}; m \{ dt_i; dt_i; dt_i; dt_i; dt_i; dt_i; dt_{fi}(t_i); i=i; i=l; i=l; i=l; i=i(i, i); i=i(i, i); pl; pi; pi; pi(i);$

	rl; };
Магазин	$Z_0S\};Er;E$
Магазин	$Z_0S\};Er;EvE$
Магазин	$Z_0S\};Er;Evi$
Лента	$(ivi);ri; \} tfi(ti,ti)\{ dti;dtfi(ti,ti,ti);i=i(i,l,l)vi;ri; \};m\{ dti;dti;dti;dti;dti;dti;dtfi(ti);i=i;i=l;i=l;i=l;i=i(i,i);i=i(i,i);pl;pi;pi;pi(i);rl; \};$
Магазин	$Z_0S\};Er;E$
Магазин	$Z_0S\};Er;)E($
Лента	$ivi);ri; \} tfi(ti,ti)\{ dti;dtfi(ti,ti,ti);i=i(i,l,l)vi;ri; \};m\{ dti;dti;dti;dti;dti;dti;dtfi(ti);i=i;i=l;i=l;i=l;i=i(i,i);i=i(i,i);pl;pi;pi;pi(i);rl; \};$
Магазин	$Z_0S\};Er;)E$
Магазин	$Z_0S\};Er;)EvE$
Магазин	$Z_0S\};Er;)Evi$
Лента	$i);ri; \} tfi(ti,ti)\{ dti;dtfi(ti,ti,ti);i=i(i,l,l)vi;ri; \};m\{ dti;dti;dti;dti;dti;dti;dtfi(ti);i=i;i=l;i=l;i=l;i=i(i,i);i=i(i,i);pl;pi;pi;pi(i);rl; \};$
Магазин	$Z_0S\};Er;)E$
Магазин	$Z_0S\};Er;)i$
Лента	$i; \} tfi(ti,ti)\{ dti;dtfi(ti,ti,ti);i=i(i,l,l)vi;ri; \};m\{ dti;dti;dti;dti;dti;dti;dtfi(ti);i=i;i=l;i=l;i=l;i=i(i,i);i=i(i,i);pl;pi;pi;pi(i);rl; \};$
Магазин	$Z_0S\};E$
Магазин	$Z_0S\};i$
Лента	$tfi(ti,ti)\{ dti;dtfi(ti,ti,ti);i=i(i,l,l)vi;ri; \};m\{ dti;dti;dti;dti;dti;dti;dtfi(ti);i=i;i=l;i=l;i=l;i=i(i,i);i=i(i,i);pl;pi;pi;pi(i);rl; \};$
Магазин	$Z_0S$
Магазин	$Z_0SC$
Магазин	$Z_0S; \} B \} )F(ift$
Лента	$ti,ti)\{ dti;dtfi(ti,ti,ti);i=i(i,l,l)vi;ri; \};m\{ dti;dti;dti;dti;dti;dti;dtfi(ti);i=i;i=l;i=l;i=l;i=i(i,i);i=i(i,i);pl;pi;pi;pi(i);rl; \};$
Магазин	$Z_0S; \} B \} )F$
Магазин	$Z_0S; \} B \} ) F,it$
Лента	$ti)\{ dti;dtfi(ti,ti,ti);i=i(i,l,l)vi;ri; \};m\{ dti;dti;dti;dti;dti;dti;dtfi(ti);i=i;i=l;i=l;i=l;i=i(i,i);i=i(i,i);pl;pi;pi;pi(i);rl; \};$
Магазин	$Z_0S; \} B \} )F$
Магазин	$Z_0S; \} B \} )it$
Лента	$dti;dtfi(ti,ti,ti);i=i(i,l,l)vi;ri; \};m\{ dti;dti;dti;dti;dti;dti;dtfi(ti);i=i;i=l;i=l;i=l;i=i(i,i);i=i(i,i);pl;pi;pi;pi(i);rl; \};$
Магазин	$Z_0S; \} B$
Магазин	$Z_0S; \};ErN$
Магазин	$Z_0S; \};ErNO$
Магазин	$Z_0S; \};ErN;itd$
Лента	$dtfi(ti,ti,ti);i=i(i,l,l)vi;ri; \};m\{ dti;dti;dti;dti;dti;dti;dtfi(ti);i=i;i$

	$=l;i=l;i=l;i=i(i,i);i=i(i,i);pl;pi;pi;pi(i);rl; \};$
Магазин	$Z_0S; \};ErN$
Магазин	$Z_0S; \};ErNO$
Магазин	$Z_0S; \};ErN; )F(ift d$
Лента	$ti,ti,ti);i=i(i,l,l)vi;ri; \};m\{ dti;dti;dti;dti;dti;dti;dtfi(ti);i=i;i=l;i=l;i=l;i=i(i,i);i=i(i,i);pl;pi;pi;pi(i);rl; \};$
Магазин	$Z_0S; \};ErN; )F$
Магазин	$Z_0S; \};ErN; )F,it$
Лента	$ti,ti);i=i(i,l,l)vi;ri; \};m\{ dti;dti;dti;dti;dti;dti;dtfi(ti);i=i;i=l;i=l;i=l;i=i(i,i);i=i(i,i);pl;pi;pi;pi(i);rl; \};$
Магазин	$Z_0S; \};ErN; )F$
Магазин	$Z_0S; \};ErN; )F,it$
Лента	$ti);i=i(i,l,l)vi;ri; \};m\{ dti;dti;dti;dti;dti;dti;dtfi(ti);i=i;i=l;i=l;i=l;i=i(i,i);i=i(i,i);pl;pi;pi;pi(i);rl; \};$
Магазин	$Z_0S; \};ErN; )F$
Магазин	$Z_0S; \};ErN; )it$
Лента	$i=i(i,l,l)vi;ri; \};m\{ dti;dti;dti;dti;dti;dti;dtfi(ti);i=i;i=l;i=l;i=l;i=i(i,i);i=i(i,i);pl;pi;pi;pi(i);rl; \};$
Магазин	$Z_0S; \};ErN$
Магазин	$Z_0S; \};ErO$
Магазин	$Z_0S; \};Er;E=i$
Лента	$i(i,l,l)vi;ri; \};m\{ dti;dti;dti;dti;dti;dti;dtfi(ti);i=i;i=l;i=l;i=l;i=i(i,i);i=i(i,i);pl;pi;pi;pi(i);rl; \};$
Магазин	$Z_0S; \};Er;E$
Магазин	$Z_0S; \};Er;EvE$
Магазин	$Z_0S; \};Er;Ev)W(i$
Лента	$i,l,l)vi;ri; \};m\{ dti;dti;dti;dti;dti;dti;dtfi(ti);i=i;i=l;i=l;i=l;i=i(i,i);i=i(i,i);pl;pi;pi;pi(i);rl; \};$
Магазин	$Z_0S; \};Er;Ev)W$
Магазин	$Z_0S; \};Er;Ev)W,i$
Лента	$l,l)vi;ri; \};m\{ dti;dti;dti;dti;dti;dti;dtfi(ti);i=i;i=l;i=l;i=l;i=i(i,i);i=i(i,i);pl;pi;pi;pi(i);rl; \};$
Магазин	$Z_0S; \};Er;Ev)W$
Магазин	$Z_0S; \};Er;Ev)W,l$
Лента	$l)vi;ri; \};m\{ dti;dti;dti;dti;dti;dti;dtfi(ti);i=i;i=l;i=l;i=l;i=i(i,i);i=i(i,i);pl;pi;pi;pi(i);rl; \};$
Магазин	$Z_0S; \};Er;Ev)W$
Магазин	$Z_0S; \};Er;Ev)l$
Лента	$l;ri; \};m\{ dti;dti;dti;dti;dti;dti;dtfi(ti);i=i;i=l;i=l;i=l;i=i(i,i);i=i(i,i);pl;pi;pi;pi(i);rl; \};$
Магазин	$Z_0S; \};Er;E$
Магазин	$Z_0S; \};Er;l$
Лента	$ri; \};m\{ dti;dti;dti;dti;dti;dti;dtfi(ti);i=i;i=l;i=l;i=l;i=i(i,i);i=i(i$

	,i);pl;pi;pi;pi(i);rl;};
Магазин	$Z_0S;};Er$
Лента	$i;};m\{dti;dti;dti;dti;dti;dti;dtfi(ti);i=i;i=l;i=l;i=l;i=i(i,i);i=i(i,i);pl;pi;pi;pi(i);rl;};$
Магазин	$Z_0S;};E$
Магазин	$Z_0S;};i$
Лента	$m\{dti;dti;dti;dti;dti;dti;dtfi(ti);i=i;i=l;i=l;i=l;i=i(i,i);i=i(i,i);pl;pi;pi;pi(i);rl;};$
Магазин	$Z_0S$
Магазин	$Z_0C$
Магазин	$Z_0;}\}B\{m$
Лента	$dti;dti;dti;dti;dti;dti;dtfi(ti);i=i;i=l;i=l;i=l;i=i(i,i);i=i(i,i);pl;pi;pi;pi(i);rl;};$
Магазин	$Z_0;}\}B$
Магазин	$Z_0;};ErN$
Магазин	$Z_0;};ErNO$
Магазин	$Z_0;};ErN;itd$
Лента	$dti;dti;dti;dti;dti;dtfi(ti);i=i;i=l;i=l;i=l;i=i(i,i);i=i(i,i);pl;pi;pi;pi(i);rl;};$
Магазин	$Z_0;};ErN$
Магазин	$Z_0;};ErNO$
Магазин	$Z_0;};ErN;itd$
Лента	$dti;dti;dti;dti;dtfi(ti);i=i;i=l;i=l;i=l;i=i(i,i);i=i(i,i);pl;pi;pi;pi(i);rl;};$
Магазин	$Z_0;};ErN$
Магазин	$Z_0;};ErNO$
Магазин	$Z_0;};ErN;itd$
Лента	$dti;dti;dti;dtfi(ti);i=i;i=l;i=l;i=l;i=i(i,i);i=i(i,i);pl;pi;pi;pi(i);rl;};$
Магазин	$Z_0;};ErN$
Магазин	$Z_0;};ErNO$
Магазин	$Z_0;};ErN;itd$
Лента	$dti;dti;dtfi(ti);i=i;i=l;i=l;i=l;i=i(i,i);i=i(i,i);pl;pi;pi;pi(i);rl;};$
Магазин	$Z_0;};ErN$
Магазин	$Z_0;};ErNO$
Магазин	$Z_0;};ErN;itd$
Лента	$dti;dtfi(ti);i=i;i=l;i=l;i=l;i=i(i,i);i=i(i,i);pl;pi;pi;pi(i);rl;};$
Магазин	$Z_0;};ErN$
Магазин	$Z_0;};ErNO$
Магазин	$Z_0;};ErN;itd$
Лента	$dtfi(ti);i=i;i=l;i=l;i=l;i=i(i,i);i=i(i,i);pl;pi;pi;pi(i);rl;};$
Магазин	$Z_0;};ErN$
Магазин	$Z_0;};ErNO$

Магазин	$Z_0; \}; \text{ErNO}; \text{F(ift}$
Лента	$\text{ti}); i=i; i=l; i=l; i=l; i=i(i, i); i=i(i, i); \text{pl}; \text{pi}; \text{pi}; \text{pi}(i); \text{rl}; \};$
Магазин	$Z_0; \}; \text{ErNO}; \text{F}$
Магазин	$Z_0; \}; \text{ErNO}; \text{it}$
Лента	$i=i; i=l; i=l; i=l; i=i(i, i); i=i(i, i); \text{pl}; \text{pi}; \text{pi}; \text{pi}(i); \text{rl}; \};$
Магазин	$Z_0; \}; \text{ErNO}$
Магазин	$Z_0; \}; \text{ErN}; \text{E}=i$
Лента	$i; i=l; i=l; i=l; i=i(i, i); i=i(i, i); \text{pl}; \text{pi}; \text{pi}; \text{pi}(i); \text{rl}; \};$
Магазин	$Z_0; \}; \text{ErN}; \text{E}$
Магазин	$Z_0; \}; \text{ErN}; i$
Лента	$i=l; i=l; i=l; i=i(i, i); i=i(i, i); \text{pl}; \text{pi}; \text{pi}; \text{pi}(i); \text{rl}; \};$
Магазин	$Z_0; \}; \text{ErN}$
Магазин	$Z_0; \}; \text{ErNO}$
Магазин	$Z_0; \}; \text{ErN}; \text{E}=i$
Лента	$i=l; i=l; i=i(i, i); i=i(i, i); \text{pl}; \text{pi}; \text{pi}; \text{pi}(i); \text{rl}; \};$
Магазин	$Z_0; \}; \text{ErN}$
Магазин	$Z_0; \}; \text{ErNO}$
Магазин	$Z_0; \}; \text{ErN}; \text{E}=i$
Лента	$l; i=l; i=i(i, i); i=i(i, i); \text{pl}; \text{pi}; \text{pi}; \text{pi}(i); \text{rl}; \};$
Магазин	$Z_0; \}; \text{ErN}; \text{E}$
Магазин	$Z_0; \}; \text{ErN}; l$
Лента	$i=l; i=i(i, i); i=i(i, i); \text{pl}; \text{pi}; \text{pi}; \text{pi}(i); \text{rl}; \};$
...	...
Лента	
Магазин	$Z_0$
Магазин	
<b>Цепочка разобрана</b>	

аргументы	$\alpha^R$	$\alpha^R$	$\alpha^R$	$\alpha^R$	$\alpha^R$
$q_0, \lambda, S$	C	SC			
$q_0, \lambda, C$	$; \} B \{ \text{F(ift}$	$; \} B \{ m$			
$q_0, \lambda, B$	$; \text{ErN}$				
$q_0, \lambda, N$	O	NO			
$q_0, \lambda, O$	$; \text{itd}$	$; \text{Er}$	$; \text{E}=i$	$; \text{F(ift}$	
$q_0, \lambda, E$	i	l	)E(	EvE	)W(i
$q_0, \lambda, F$	it	F, it			
$q_0, \lambda, W$	i	l	W, i	W, l	