

Описания к транслятору-интерпретатору:

Список лексем с номерами:

- 1) 'int'
- 2) 'float'
- 3) 'Int1'
- 4) 'lf'
- 5) 'while'
- 6) ':'
- 7) '.'
- 8) ';' ;
- 9) ',' ;
- 10) '+'
- 11) '-'
- 12) '/'
- 13) '*'
- 14) '('
- 15) ')'
- 16) '['
- 17) ']'
- 18) '{'
- 19) '}'
- 20) 'print'
- 21) 'scan'
- 22) '>='
- 23) '<='
- 24) '=='
- 25) '<'
- 26) '>'
- 27) '='

28) '!='

1) *типы данных*: int, int1, float

2) операторы: $= + - / * () [] \{ \}$;

3) условные операторы и циклы с условиями, включая операции сравнения:

while усл:

If усл: else

<, >, <=, >=, !=, ==

4) операторы ввода и вывода: `scan()`, `print()`

Таблица переходов автомата:

 - буквы a, .., z, A, .., Z

<ц> - цифры 0..9

<с> - отдельные символы языка (+, -, (,), [,], {, }, ;)

<д> - другие символы (не символы языка)

ВХОДНО Й СИМВОЛ / СОСТОЯ НИЕ	<б>	<ц>	<<.»	<< »	!	>, <	=	/	*	\n	<с>	<д>
S 0	I	C		S	T	A	A	K	Z	S	Z	
I 1	I	I	Z*	Z	Z*	Z*	Z*	Z*	Z*	Z*	Z*	
C 2		C	D	Z	Z*	Z*	Z*	Z*	Z*	Z*	Z*	

<i>D 3</i>		<i>E</i>										
<i>T 4</i>							<i>Z</i>					
<i>E 5</i>		<i>E</i>	<i>Z*</i>	<i>Z</i>	<i>Z*</i>	<i>Z*</i>	<i>Z*</i>	<i>Z*</i>	<i>Z*</i>	<i>Z*</i>	<i>Z*</i>	
<i>A 6</i>	<i>Z*</i>	<i>Z*</i>	<i>Z*</i>	<i>Z</i>	<i>Z*</i>	<i>Z*</i>	<i>Z</i>	<i>Z*</i>	<i>Z*</i>	<i>Z*</i>	<i>Z*</i>	
<i>K 7</i>	<i>Z*</i>	<i>Z*</i>	<i>Z*</i>	<i>Z</i>	<i>Z*</i>	<i>Z*</i>	<i>Z*</i>	<i>Z*</i>	<i>L</i>	<i>Z*</i>	<i>Z*</i>	
<i>L 8</i>	<i>L</i>	<i>L</i>	<i>L</i>	<i>L</i>	<i>L</i>	<i>L</i>	<i>L</i>	<i>L</i>	<i>M</i>	<i>L</i>	<i>L</i>	<i>L</i>
<i>M 9</i>	<i>L</i>	<i>L</i>	<i>L</i>	<i>L</i>	<i>L</i>	<i>L</i>	<i>L</i>	<i>S</i>	<i>M</i>	<i>L</i>	<i>L</i>	<i>L</i>

Семантические программы:

цч - целое число
вч - вещественное число

ВХОДНОЙ СИМВОЛ / СОСТОЯНИЕ	<б>	<ц>	«.»	« »	!	>,<	=	/	*	\n	<с>	<д>
<i>S 0</i>	<i>I/I</i>	<i>C/2</i>		<i>S/8</i>	<i>T/1</i>	<i>A/1</i>	<i>A/1</i>	<i>K/1</i>	<i>знак */12</i>	<i>S/8</i>	<i>знак/12</i>	

<i>I 1</i>	<i>I/3</i>	<i>I/3</i>	<i>имя/9</i>	<i>имя/7</i>	<i>имя/9</i>	<i>имя/9</i>	<i>имя/9</i>	<i>имя/9</i>	<i>имя/9</i>	<i>имя/7</i>	<i>имя/9</i>	
<i>C 2</i>		<i>C/4</i>	<i>D/5</i>	<i>уч/8</i>	<i>уч/10</i>	<i>уч/10</i>	<i>уч/10</i>	<i>уч/10</i>	<i>уч/10</i>	<i>уч/10</i>	<i>уч/10</i>	
<i>D 3</i>		<i>E/6</i>										
<i>T 4</i>							<i>Знак !=/11</i>					
<i>E 5</i>		<i>E/6</i>	<i>вч/10</i>	<i>вч/8</i>	<i>вч/10</i>	<i>вч/10</i>	<i>вч/10</i>	<i>вч/10</i>	<i>вч/10</i>	<i>вч/10</i>	<i>вч/10</i>	
<i>A 6</i>	<i>знак/9</i>	<i>знак/9</i>	<i>знак/9</i>	<i>знак/9</i>	<i>знак/9</i>	<i>знак/9</i>	<i>знак/9</i>	<i>знак/9</i>	<i>знак/9</i>	<i>знак/9</i>	<i>знак/9</i>	
<i>K 7</i>	<i>знак /9</i>	<i>знак /9</i>	<i>знак /9</i>	<i>знак /7</i>	<i>знак /9</i>	<i>знак /9</i>	<i>знак /9</i>	<i>знак /9</i>	<i>L/8</i>	<i>знак /9</i>	<i>знак /9</i>	
<i>L 8</i>	<i>L/8</i>	<i>L/8</i>	<i>L/8</i>	<i>L/8</i>	<i>L/8</i>	<i>L/8</i>	<i>L/8</i>	<i>L/8</i>	<i>M/8</i>	<i>L/8</i>	<i>L/8</i>	<i>L/8</i>
<i>M 9</i>	<i>L/8</i>	<i>L/8</i>	<i>L/8</i>	<i>L/8</i>	<i>L/8</i>	<i>L/8</i>	<i>L/8</i>	<i>S/8</i>	<i>M/8</i>	<i>L/8</i>	<i>L/8</i>	<i>L/8</i>

C[i] – текущий входной символ;
 name – символьная строка;
 n – целочисленная переменная;
 x, d – вещественные переменные.
 1. name := C[i];
 2. n := ord(C[i]) – ord('0') ;

3. $\text{name} := \text{name} + \text{C}[\text{i}];$
4. $\text{n} := \text{n} * 10 + \text{ord}(\text{C}[\text{i}]) - \text{ord}('0');$
5. $\text{d} := 1; \text{x} := \text{n};$
6. $\text{d} := \text{d} * 0.1; \text{x} := \text{x} + (\text{ord}(\text{C}[\text{i}]) - \text{ord}('0')) * \text{d};$
7. Сравнение идентификатора с таблицей зарезервированных слов, для нахождения ключевых слов.
8. $\text{Name} := \text{null};$
9. $\text{j} := \text{j} - 1; 7();$
10. $\text{j}--;$
11. $\text{name} = \text{ord}(\text{C}[\text{i}]); 7();$

Грамматика:

$A \rightarrow \text{aH} = \text{SZ}$
 $C \rightarrow S < \text{SZ} \mid S > \text{SZ} \mid S \leq \text{SZ} \mid S \geq \text{SZ} \mid S \neq \text{SZ} \mid S == \text{SZ} \mid S \text{ and } \text{SZ} \mid S \text{ or } \text{SZ}$
 $S \rightarrow S + T \mid S - T \mid T$
 $T \rightarrow T * F \mid T / F \mid F$
 $F \rightarrow (S) \mid +G \mid -GZ \mid \text{aH} \mid k$
 $G \rightarrow (S) \mid \text{aH} \mid k$
 $H \rightarrow [S] \mid \lambda$
 $Z \rightarrow \lambda$

После устранения левой рекурсии, преобразования к нестрогой нормальной форме Грейбах и факторизации:

$P \rightarrow \text{int } R P \mid \text{int1 } B P \mid \text{float } R P \mid \{\text{AQ}\}$
 $R \rightarrow a M$
 $M \rightarrow , a M \mid ;$
 $B \rightarrow a[k] W$
 $W \rightarrow , a[k] W \mid ;$

[illegible]

T	$+GV$ $\square\square\square$	$-GV$ $\square\square-$			$(S)V$ $\square\square\square\square$		aHV $a\square\square$	kV $k\square$				
V	λ	λ	$*FV$ $\square\square*$	$/FV$ $\square\square/$	λ	λ	λ	λ	λ	λ	λ	λ
F	$+G$ $\square\square$	$-GZ$ $\square\square-$			(S) $\square\square\square$		aH $a\square$	k k				
G					(S) $\square\square\square$		aH $a\square$	k k				
H	λ	λ	λ	λ	λ	λ	λ	λ	$[S]$ $\square\square i$	λ	λ	
Z	λ	λ	λ	λ	λ	λ	λ	λ	λ	λ	λ	λ

Грамматика со сравнениями:

Порождающие правила для сравнения двух выражений:

$$C \rightarrow S < SZ \mid S > SZ \mid S \leq SZ \mid S \geq SZ \mid S \neq SZ \mid S == SZ \mid S \text{ and } SZ \mid S \text{ or } SZ$$

После преобразования к нормальной форме Грейбах и факторизации:

$$C \rightarrow (S)VUD \mid aHVUD \mid kVUD \mid +GVUD \mid -GVUD$$

$$D \rightarrow <SZ \mid >SZ \mid <=SZ \mid >=SZ \mid !=SZ \mid ==SZ \mid \text{and } SZ \mid \text{or } SZ$$

Семантические действия генератора ОПС для нетерминала C:

$$\square\square\square\square\square \mid a\square\square\square\square \mid k\square\square\square \mid \square\square\square\square\square \mid \square\square-\square\square$$

Для нетерминала D соответственно:

$$\square\square< \mid \square\square> \mid \square\square<= \mid \square\square>= \mid \square\square!= \mid \square\square== \mid \square\square\text{and} \mid \square\square\text{or}$$

Грамматика с условными операторами:

Порождающие правила, задающие условные операторы в полной и сокращенной форме (служебные слова if, then, :- терминалы):

$$A \rightarrow \text{if } C: AEZ$$

$$E \rightarrow \text{else } A \mid \lambda$$

Семантические действия генератора ОПС при порождении условных операторов нетерминалом A:

$$\square\square 1 \square\square 3$$

Семантические действия для нетерминала E:

$$2\square$$

Грамматика с циклами:

Порождающее правило для задания оператора цикла

(служебные слова while, :- терминалы):

$$A \rightarrow \text{while } C: AZ$$

Семантические действия генератора ОПС при порождении цикла:

$$4\square 1 \square 5$$

Числа 1, 2, 3, 4, 5 в семантических действиях обозначают выполнение семантических программ, генерирующих в ОПС операнды-метки (номера элементов ОПС) и операции условного и безусловного перехода на эти метки.

Семантические программы используют счетчик k – номер очередного генерируемого элемента ОПС , а также еще один магазин – магазин меток.

Программа 1.

1. В магазин меток записывается k .
2. В ОПС записывается пустой элемент – место для будущей метки.
3. В ОПС записывается операция jf – переход при условии $false$.

Программа 2.

1. Через верхний элемент магазина меток, как ссылку на ранее заготовленное место для метки, записывается $k + 2$.
2. В магазин меток записывается k .
3. В ОПС записывается пустой элемент – место для будущей метки.
4. В ОПС записывается операция j – безусловный переход.

Программа 3.

1. Через верхний элемент магазина меток, как ссылку на ранее заготовленное место для метки, записывается k .

Программа 4.

1. В магазин меток записывается k .

Программа 5.

1. Через верхний элемент магазина меток, как ссылку на ранее заготовленное место для метки, записывается $k + 2$.
2. В ОПС записывается метка, значение для которой читается из магазина меток.
3. В ОПС записывается операция j – безусловный переход.

Грамматика с составными операторами:

Порождающие правила для задания составного оператора, т.е. последовательности других операторов
(;, {, } - терминалы):

$$A \rightarrow \{AQ\}$$

$$Q \rightarrow ;AQ \mid \lambda$$

При этом семантические действия генератора ОПС для нетерминала A:

□□□□

Для нетерминала Q:

□□□

Грамматика с операторами ввода и вывода:

Порождающие правила для задания стандартных операторов ввода и вывода (служебные слова scan, print – терминалы):

$$A \rightarrow \text{scan} (aH) \mid \text{print} (S)$$

Семантические действия генератора ОПС соответственно:

□□a□s | □□□p

s – операция чтения со стандартного устройства ввода в переменную, операнд должен быть ссылкой на переменную

p - вывод значения арифметического выражения в стандартное устройство вывода, операнд – числовое значение.

Грамматика с описаниями переменных:

Начальный нетерминал P определяет программу в целом:

$$P \rightarrow \text{int } R \ P \mid \text{int1 } R \ P \mid \text{float } R \ P \mid \{A \ Q\}$$

$$R \rightarrow a \ M$$

$$M \rightarrow , \ a \ M \mid ;$$

Служебные слова int, int1, float задают описания:

int - целых переменных, int1 - одномерных массивов целых, float - вещественных переменных

Семантические действия генератора ОПС соответственно

для нетерминала P:

11□□ | 12□□ | 13□□ | 14□□15

для нетерминала R:

16□

для нетерминала R:

□16□ | □

Числа 11, 12, 13, 14, 15 задают семантические программы.

Семантические программы

Программа 11.

Переключение на заполнение таблицы переменных типа int (целочисленных), в таблице будут записываться имена переменных.

Программа 12.

Переключение на заполнение таблицы переменных типа одномерный массив int1 (целочисленных), в таблице будут записываться имена массивов.

Программа 13.

Переключение на заполнение таблицы переменных типа float (с плавающей точкой), в таблице будут записываться имена переменных.

Программа 14.

1. Завершение формирования таблиц переменных.
2. Генерация в ОПС операций выделения памяти блоками для каждого из типов переменных:
– для типа int и float – в виде массива целых, каждая переменная в нем занимает отдельный элемент и ей приписан номер;

– для типа int1 – в виде массива паспортов массивов, в них записывается размерность, а также ссылка на начало размещения массива в памяти.

3. Генерация в ОПС операций обнуления паспортов массивов для переменных типа int1.

Программа 15.

Генерация в ОПС операций освобождения всех выделенных в процессе выполнения программы блоков памяти.

Программа 16.

1. Проверка имени переменной, поступающей из входной цепочки, на совпадение с именами, ранее занесенными в таблицы. Если есть совпадение, то сигнализация об ошибке «повторное описание переменной».

2. Если ошибки нет, то добавление имени переменной в одну из таблиц переменных (на которую ранее было переключение).