

Лабораторная работа №2.  
Ручное построение нисходящих синтаксических  
анализаторов.  
Вариант №8

Нечаев Михаил

20.04.2014

## 1 Разработка грамматики

### 1.1 Задание

Описания переменных в Си.

### 1.2 Грамматика

$$\begin{aligned} S &\rightarrow WS \\ S &\rightarrow \varepsilon \\ W &\rightarrow < ident > V \\ V &\rightarrow < pointers > < ident >; \\ V &\rightarrow < pointers > < ident >, V \\ V &\rightarrow < ident >; \\ V &\rightarrow < ident >, V \end{aligned}$$

Нетерминал	Описание
$S$	Начальный терминал.
$W$	Корректное описание переменных одного типа.
$V$	Список переменных одного типа.

Левая рекурсия и правое ветвление отсутствует.

## 2 Построение лексического анализатора

В нашей грамматике четыре терминала: ‘,’, ‘;’, ‘<pointers>’ (указатели), ‘<ident>’ (идентификатор: тип или имя переменной).

Заведём класс **Token** для хранения терминалов.

```
public enum Token {  
    COMMA, SEMICOLON, POINTERS, END, IDENT  
}
```

Терминал	Токен
.	COMMA
;	SEMICOLON
<pointers>	POINTERS
\$	END
<ident>	IDENT

### 3 Построение синтаксического анализатора

Построим множество **FIRST** и **FOLLOW** для нетерминалов грамматики.  
???

Нетерминал	FIRST	FOLLOW
<i>S</i>	;, $\varepsilon$	<ident>, \$
<i>W</i>	;, $\varepsilon$	<ident>, \$
<i>V</i>	<ident>	<ident>, \$

### 4 Тесты с визуализацией

#### 4.1 Пример из задания

```
"int a, *b, ***c, d;"
```

```
S
# W
# # <type>::int
# # V
# # # <variable>::a
# # # ,
# # # V
# # # # <pointer's>::*
# # # # <name>::b
# # # # ,
# # # # V
# # # # # <pointer's>::***
# # # # # <name>::c
# # # # # ,
# # # # # V
# # # # # # <variable>::d
# # # # # # ;
# S
# # <end>
```

#### 4.2 $S \rightarrow WS$

```
0| "int x; int y; int z,\nz;"
1| " int x;\n\n\tint y;\t"
2| "int\nx\n,\ny\n;double y;"

0|
```

```

S
# W
# # <type>::int
# # V
# # # <variable>::x
# # # ;
# S
# # W
# # # <type>::int
# # # V
# # # # <variable>::y
# # # # ;
# # S
# # # W
# # # # <type>::int
# # # # V
# # # # # <variable>::z
# # # # # ,
# # # # # V
# # # # # # <variable>::z
# # # # # # ;
# # # S
# # # # <end>

```

1|

```

S
# W
# # <type>::int
# # V
# # # <variable>::x
# # # ;
# S
# # W
# # # <type>::int
# # # V
# # # # <variable>::y
# # # # ;
# # S
# # # <end>

```

2|

```

S
# W
# # <type>::int
# # V
# # # <variable>::x
# # # ,
# # # V
# # # # <variable>::y
# # # # ;

```

```

# S
# # W
# # # <type>::double
# # # V
# # # # <variable>::y
# # # # ;
# # S
# # # <end>

```

### 4.3 $S \rightarrow \varepsilon$

```

0| ""
1| " "
2| " \t "
3| " \r \n \t \t \n"

```

```

0|
S
# <end>

```

```

1|
S
# <end>

```

```

2|
S
# <end>

```

```

3|
S
# <end>

```

### 4.4 Несколько корректных тестов

```

0| "int x;"
1| "int x,Y;"
2| "int x,y; double z;"
3| "int x;double z;"
4| "int x,*y,\n**z;\tx*y;"

```

```

0|
S
# W
# # <type>::int
# # V
# # # <variable>::x
# # # ;
# S
# # <end>

```

```

1|

```

```

S
# W
# # <type>::int
# # V
# # # <variable>::x
# # # ,
# # # V
# # # # <variable>::Y
# # # # ;
# S
# # <end>

```

```

2|
S
# W
# # <type>::int
# # V
# # # <variable>::x
# # # ,
# # # V
# # # # <variable>::y
# # # # ;
# S
# # W
# # # <type>::double
# # # V
# # # # <variable>::z
# # # # ;
# # S
# # # <end>

```

```

3|
S
# W
# # <type>::int
# # V
# # # <variable>::x
# # # ;
# S
# # W
# # # <type>::double
# # # V
# # # # <variable>::z
# # # # ;
# # S
# # # <end>

```

```

4|
S
# W

```

```

# # <type >::int
# # V
# # # <variable >::x
# # # ,
# # # V
# # # # <pointer 's >::*
# # # # <name>::y
# # # # ,
# # # # V
# # # # # <pointer 's >::**
# # # # # <name>::z
# # # # # ;
# S
# # W
# # # <type >::x
# # # V
# # # # <pointer 's >::*
# # # # <name>::y
# # # # ;
# # S
# # # <end>

```

#### 4.5 Тесты на ParseException. Некорректный символ

```

0| "1"
1| "int 3"
2| "_"
3| "int x,)"
4| "int x,y&"
5| "int x,y;&"

```

```

0| Illegal character 1 at pos 1
1| Illegal character 3 at pos 5
2| Illegal character _ at pos 1
3| Illegal character ) at pos 7
4| Illegal character & at pos 8
5| Illegal character & at pos 9

```

#### 4.6 Тесты на ParseException. Неожиданный нетерминал

```

0| "*,",
1| ",",
2| ";",
3| "  *",
4| "int ,",
5| "int ;",
6| "int ,",
7| "int;",
8| "int x,;",
9| "int x,,",
10| "int x,*,",

```

0| 'Ident ' or 'End' expected at pos 1  
1| 'Ident ' or 'End' expected at pos 1  
2| 'Ident ' or 'End' expected at pos 1  
3| 'Ident ' or 'End' expected at pos 3  
4| 'Pointers ' or 'Ident ' expected at pos 5  
5| 'Pointers ' or 'Ident ' expected at pos 5  
6| 'Pointers ' or 'Ident ' expected at pos 4  
7| 'Pointers ' or 'Ident ' expected at pos 4  
8| 'Pointers ' or 'Ident ' expected at pos 7  
9| 'Pointers ' or 'Ident ' expected at pos 7  
10| 'Ident ' expected at pos 8