

**Отчет по лабораторной работе**  
**«Ручное построение нисходящих синтаксических анализаторов»**  
**студента группы 3538**  
**Андрея Козлова**

## Вариант 6. Логические формулы в стиле Си

Логические формулы. Используются операции  $\&$ ,  $|$ ,  $\wedge$ ,  $!$ . Приоритет операций стандартный. Скобки могут использоваться для изменения приоритета.

В качестве операндов выступают переменные с именем из одной буквы. Используйте один терминал для всех переменных. Для каждой логической операции должен быть заведен один терминал.

Пример:  $(!a | b) \& a \& (a | !(b \wedge c))$

# 1 Разработка грамматики

Построим грамматику.

$E \rightarrow A$   
 $E \rightarrow A \mid E$   
 $A \rightarrow B$   
 $A \rightarrow B \wedge A$   
 $B \rightarrow C$   
 $B \rightarrow C \& B$   
 $C \rightarrow D$   
 $C \rightarrow !C$   
 $D \rightarrow v$   
 $D \rightarrow (E)$

Нетерминал	Описание
E	Логическое выражение
A	Дизъюнкт (слабая дизъюнкция)
B	Дизъюнкт (строгая дизъюнкция)
C	Конъюнкт или его отрицание
D	Переменная или логическое выражение

В грамматике есть левая рекурсия. Устраним её. Получится грамматика:

$E \rightarrow A E'$   
 $E' \rightarrow \mid A E'$   
 $E' \rightarrow \varepsilon$   
 $A \rightarrow B A'$   
 $A' \rightarrow \wedge B A'$   
 $A' \rightarrow \varepsilon$   
 $B \rightarrow C B'$   
 $B' \rightarrow \& C B'$   
 $B' \rightarrow \varepsilon$   
 $C \rightarrow D$   
 $C \rightarrow !C$   
 $D \rightarrow v$   
 $D \rightarrow (E)$

Нетерминал	Описание
E	Логическое выражение
E'	Продолжение слабой дизъюнкции
A	Дизъюнкт (слабая дизъюнкция)
A'	Продолжение строгой дизъюнкции
B	Дизъюнкт (строгая дизъюнкция)
B'	Продолжение конъюнкции
C	Конъюнкт или его отрицание
D	Переменная или логическое выражение

## 2 Построение лексического анализатора

В нашей грамматике 7 терминалов.

Построим лексический анализатор. Заведем класс Token для хранения терминалов. Не забудем также про конец строки.

Терминал	Токен
v	VARIABLE
	OR_OPERATOR
^	XOR_OPERATOR
&	AND_OPERATOR
!	NOT_OPERATOR
(	LEFT_PARENTHESIS
)	RIGHT_PARENTHESIS
\$	END

## 3 Построение синтаксического анализатора

Построим множества FIRST и FOLLOW для нетерминалов грамматики.

Нетерминал	FIRST	FOLLOW
E	v, (, !	), \$
E'	, ε	), \$
A	v, (, !	, ), \$
A'	^, ε	, ), \$
B	v, (, !	^,  , ), \$
B'	&, ε	^,  , ), \$
C	v, (, !	&, ^,  , ), \$
D	v, (	&, ^,  , ), \$

## 4 Визуализация дерева разбора

Пакет tools содержит лексический (LexicalAnalyzer), синтаксический (Parser) анализаторы и вспомогательные классы (Tree, Token).

Пакет visualizer содержит визуализатор (Visualizer) и вспомогательные классы (Vertex, Edge). Визуализатор рисует дерево разбора для теста из файла test.txt.

## 5 Разработка набора тестов

Ниже приведён набор тестов.

Тест	Описание
a bc a * b	Неверные тесты, порождающие ParseException на уровне лексического анализатора
b ( b ) a && b	Неверные тесты, порождающие ParseException на уровне синтаксического анализатора
a	Тест на правила: $E \rightarrow A E'$ , $E' \rightarrow \epsilon$ , $A \rightarrow B A'$ , $A' \rightarrow \epsilon$ , $B \rightarrow C B'$ , $B' \rightarrow \epsilon$ , $C \rightarrow D$ , $D \rightarrow v$
(a   b ^ c & !d)	Тест на все правила
A   b & C	Тест на названия переменных
((a)) ^ (!b   (a & c)   d ^ d)   a	Небольшой случайный тест
(a ^ !(((a   a) & D & c ^ c)   b) & d)   V	Небольшой случайный тест

## 6 Ссылки

- Исходный код на [github.com](https://github.com).