

Лабораторная работа №2. Ручное построение нисходящих синтаксических анализаторов

1. Разработка грамматики

Построим грамматику.

- $E \rightarrow C \mid E \text{ or } C$
- $C \rightarrow X \mid C \text{ and } X$
- $X \rightarrow N \mid X \text{ xor } N$
- $N \rightarrow [a-z] \mid \text{not } N \mid (E)$

Нетерминал	Описание
E (Expression)	Логическое выражение
C (Conjunction)	Конъюнкция
X (Xor)	Ксор
N (Negate)	Отрицание

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

- $E \rightarrow CE' \mid C$
- $E' \rightarrow \text{or } CE' \mid \text{or } C$
- $C \rightarrow XC' \mid X$
- $C' \rightarrow \text{and } XC' \mid \text{and } X$
- $X \rightarrow NX' \mid N$
- $X' \rightarrow \text{xor } NX' \mid \text{xor } N$
- $N \rightarrow [a-z] \mid \text{not } N \mid (E)$

А потом упростим:

- $E \rightarrow CE'$
- $E' \rightarrow \text{or } CE' \mid \varepsilon$

- $C \rightarrow XC'$
- $C' \rightarrow \text{and } XC' \mid \varepsilon$
- $X \rightarrow NX'$
- $X' \rightarrow \text{xor } NX' \mid \varepsilon$
- $N \rightarrow [a-z] \mid \text{not } N \mid (E)$

	Нетерминал	Описание
	E	Логическое выражение
	E'	Продолжение логического выражения
	C	Конъюнкция
	C'	Продолжение конъюнкции
	X	Ксор
	X'	Продолжение ксора
	N	Отрицание

2. Построение лексического анализатора

В грамматике терминалы — $[a - z]$, *or*, *xor*, *and*, *not*.

Построим лексический анализатор. Заведем класс *Token* для хранения терминалов.

```
public enum Token {
    LPAREN, RPAREN, VARIABLE, AND, OR, XOR, NOT, END
}
```

	Терминал	Токен
	(LPAREN
)	RPAREN
	a-z	VARIABLE
	not	NOT
	and	AND
	or	OR
	xor	XOR
	\$	END

3. Построение синтаксического анализатора

Построим множества *FIRST* и *FOLLOW* для нетерминалов нашей грамматики.

	Нетерминал	FIRST	FOLLOW
	E	(, a-z, not), \$
	E'	or, ε), \$
	C	(, a-z, not	or,), \$
	C'	and, ε	or,), \$
	X	(, a-z, not	and, or,), \$
	X'	xor, ε	and, or,), \$
	N	(, a-z, not,	xor, and, or,), \$

Тесты:

- a, m, z — Тесты на правило $N \rightarrow [a - z]$
- a and b — Тест на примитивную операцию
- a xor b — Тест на примитивную операцию
- a or b — Тест на примитивную операцию
- *not* a — Тест на примитивную операцию
- (a) — Тест на правило (E)
- (((((a and b)))))) — Тест с большим количеством вложенных скобок
- (a and (b and (c or (not d xor not c))) or not c) — Большой тест
- not not not not not b — Множественное применение not
- (— Тест на некорректную скобочную последовательность
-) — Тест на некорректную скобочную последовательность
- (a an b) — Тест на некорректную операцию
- (a and (b xor)) — Тест на недостающую переменную