## Лабораторная №2

## Ручное построение нисходящего синтаксического анализатора регулярных выражений (вариант №2)

1) Построим грамматику. Нам нужно иметь возможность делать замыкания Клини, операцию "или" (выбора), конкатенацию, скобки и получать буквенный символ.

р.ѕ. "//" разделяет варианты переходов из левой части правил

 $E \to TE'$  - конкатенация всего, кроме "|", с "|" и продолжением, если необходимо

E' -> eps // |TE' - рекурсия на случай, если в текущей строке несколько "|"

 $T ext{ -> } HT' ext{ -}$  нужен, потому что если был только T' строка могла бы начинаться с \*

T' -> eps // HT' // \*T' - рекурсия для конкатенации символов и скобочных выражений, а также для символа \* после них

H -> KH'

H' -> eps // ?

K -> a // (E) - буквы и скобки

2) Решение в коде - LexicalAnalyzer - лексический анализатор, Parser - синтаксический с использованием рекурсивного спуска

3) FIRST FOLLOW

E (, a ), \$

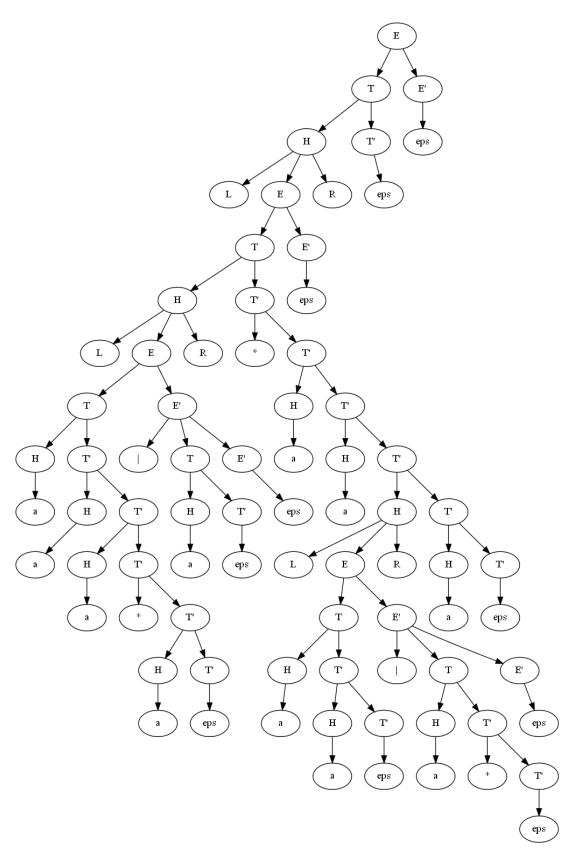
E' |, eps \$

T (, a ), |, \$

T' (, a, \*, eps ), \*, |, \$

H a, ( a, ), |, \*, (

4) Визуализация дерева разбора на примере строки ((abc\*b|a)\*ab(aa|b\*)b)\*



5) Тесты в коде