

# Ручное построение нисходящих синтаксических анализаторов.

Шайхутдинов Тимур

24 марта 2016 г.

## 1 Разработка грамматики

### 1.1 Задание

Регулярные выражения с операциями конкатенации (простая последовательная запись строк), выбора (вертикальная черта), замыкания Клини. Приоритет операций стандартный. Скобки могут использоваться для изменения приоритета. Для обозначения базовых языков используются маленькие буквы латинского алфавита. Используйте один терминал для всех символов. Пример:  $((abc^*b|a)^*ab(aa|b^*)b)^*$

### 1.2 Грамматика

Построим грамматику по описанию задания

$$\begin{aligned} S &\rightarrow C \\ S &\rightarrow \varepsilon \\ C &\rightarrow (C) \\ C &\rightarrow n \\ C &\rightarrow C|C \\ C &\rightarrow C^* \\ C &\rightarrow CC \end{aligned}$$

Нетерминал	Описание
$S$	Правильное регулярное выражение
$C$	Непустое правильное регулярное выражение

В грамматике есть левая рекурсия, устраним её.

$$\begin{aligned} S &\rightarrow C \\ S &\rightarrow \varepsilon \\ C &\rightarrow (C)C' \\ C &\rightarrow nC' \\ C &\rightarrow (C) \\ C &\rightarrow n \\ C' &\rightarrow |CC' \\ C' &\rightarrow *C' \\ C' &\rightarrow CC' \\ C' &\rightarrow |C \\ C' &\rightarrow *C \\ C' &\rightarrow C \end{aligned}$$

Нетерминал	Описание
$S$	Правильное регулярное выражение
$C$	Непустое правильное регулярное выражение
$C'$	Непустое продолжение правильного регулярного выражения

Избавимся от правого ветвления.

$$\begin{aligned}
S &\rightarrow C \\
S &\rightarrow \varepsilon \\
C &\rightarrow (C)S' \\
C &\rightarrow nS' \\
C' &\rightarrow |CS' \\
C' &\rightarrow *S' \\
C' &\rightarrow S' \\
S' &\rightarrow C' \\
S' &\rightarrow \varepsilon
\end{aligned}$$

Нетерминал	Описание
$S$	Правильное регулярное выражение
$C$	Непустое правильное регулярное выражение
$S'$	Продолжение правильного регулярного выражения
$C'$	Непустое продолжение правильного регулярного выражения

## 2 Построение лексического анализатора

В грамматике пять терминалов: '|', '\*', '(', ')', 'n' (универсальный терминал для обозначение всех символов).

Заведём класс **Token** для хранения терминалов.

```
public enum Token {
    OPEN_BRACKET, CLOSE_BRACKET,
    KLEENE_CLOSURE, CHOOSE, CHARACTER, END
}
```

Терминал	Токен
(	OPEN_BRACKET
)	CLOSE_BRACKET
*	KLEENE_CLOSURE
	CHOOSE
n	CHARACTER
\$	END

## 3 Построение синтаксического анализатора

Множества **FIRST** и **FOLLOW** для нетерминалов грамматики.

Нетерминал	FIRST	FOLLOW
$S$	(, n, $\varepsilon$	(, n, $\varepsilon$ , *,  , ), \$
$C$	(, n	(, n, $\varepsilon$ , *,  , ), \$
$S'$	(, n,  , *, $\varepsilon$	(, n, $\varepsilon$ , *,  , ), \$
$C'$	(, n,  , *	(, n, $\varepsilon$ , *,  , ), \$

## 4 Тестирование

Программа вручную была протестирована на исчерпывающем наборе тестов.

### 4.1 Терминалы. Операции

Пустое выражение:

Одиночный символ: a

Конкатенация: aaasdf

Замыкание Клини:  $abs^{**}a^{*}a^{**}a^{**}a^{**}a^{*}$

Операция выбора:  $a|a|aa|a$

Правильная скобочная последовательность:  $(a)(aa)((aaa))(a))$

### 4.2 Простые конструкции

(exp): (a)

$exp^{*}|exp: a^{*}|a$

$(exp)(exp): (a)(a)$

$(exp)^{*}: (a)^{*}$

$exp|(exp): a|(a)$

### 4.3 Составные тесты

ПСП + конкатенация:  $((a)a)a(a), (a)(aa)(aaa(a))(a((a)a(a(a)))a)$

Приоритеты операций:  $a(a^{*}|a)^{*}a|(a(a)a^{*}a|a)a^{*}a|(a^{*}a)aa^{*}a, (a^{*}|a)|(a^{**}b|a)((a|b^{*})|c^{*}|a^{*})|d), (g|a)((de^{*})(ifmo|(r^{*}u^{*}))^{*})$

Тест из примера:  $((abc^{*}b|a)^{*}ab(aa|b^{*})b)^{*}$

Игнорирование пробелов и разделителей:  $a\ a^{*}\ | \ a\ b$

### 4.4 Некорректные выражения

Не ПСП с ненулевым балансом:  $a(a^{*}|a)^{*}a|(a(aa^{*}a|a)a^{*}a|(a^{*}a)aa^{*}a$

Не ПСП с нулевым балансом:  $(a^{*}|a)|(a^{**}b|a)((a|b^{*})|c^{*}|a^{*})|d)$

ПСП с отсутствием аргумента:  $(g|a)((de^{*})(ifmo|)(r^{*}u^{*}))^{*}$