

Санкт-Петербургский национальный исследовательский университет
информационных технологий, механики и оптики
Факультет информационных технологий и программирования
Кафедра «Компьютерные Технологии»

Волков И. Р.

Отчет по лабораторной работе
«Ручное построение нисходящих синтаксических анализаторов»

Вариант 3

Санкт-Петербург
2012

Язык

Арифметические выражения в постфиксной записи. Используются операции сложения, вычитания, умножения.

В качестве операндов выступают целые числа. Используйте один терминал для всех чисел.

Пример: 1 2 10 + *

Разработка грамматики

Построим грамматику.

$$E \rightarrow E E o$$

$$E \rightarrow n$$

Нетерминал	Описание
E	Арифметическое выражение в постфиксной записи

Терминал	Описание
n	Целое число, операнд
o	Операция: сложение, вычитание или умножение

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

$$E \rightarrow n E'$$

$$E' \rightarrow E o E'$$

$$E' \rightarrow \epsilon$$

Нетерминал	Описание
E'	Продолжение арифметического выражения в постфиксной записи

Построение лексического анализатора

В грамматике 2 терминала.

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

```
enum Token {  
    NUMBER,  
    OPERATOR,  
    END  
};
```

Терминал	Токен
n	NUMBER
o	OPERATOR
\$	END

Интерфейс и реализация лексического анализатора находятся в файлах LexicalAnalyzer.h и LexicalAnalyzer.cpp.

Построение синтаксического анализатора

Построим множества FIRST и FOLLOW для нетерминалов грамматики.

Нетерминал	FIRST	FOLLOW
E	n	\$, o
E'	n, ε	\$, o

Структура данных для хранения дерева находится в файле Tree.h.

Интерфейс и реализация синтаксического анализатора с использованием рекурсивного спуска находятся в файлах Parser.h и Parser.cpp.

Визуализация дерева разбора

Программа умеет рисовать деревья разбора по очереди для тестов из входного файла tests.txt.

Разработка набора тестов

Ниже приведён набор тестов.

Тест	Описание
3	Тест на правила: $E \rightarrow n E'$, $E' \rightarrow \epsilon$
5 6 +	Тест на все правила, то есть: $E \rightarrow n E'$, $E' \rightarrow E o E'$, $E' \rightarrow \epsilon$
65 -99 *	Тест с отрицательным операндом
432 123 - 567 +	Небольшой случайный тест
4 + +	Неверный тест для проверки ParseException из синтаксического анализатора
567 876 9 * *	Небольшой случайный тест
56 abcd	Неверный тест для проверки ParseException из лексического анализатора

Кроме этих тестов, еще 100 случайных тестов созданы генератором тестов testsGenerator.py. Размер максимального теста — 61 число (и 60 операторов, соответственно).

Ссылки

- Репозиторий с исходными кодами: <https://github.com/Tsar/parsingLab2>