

МИНОБРНАУКИ РОССИИ
САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ
ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ
«ЛЭТИ» ИМ. В.И. УЛЬЯНОВА (ЛЕНИНА)
Кафедра МОЭВМ

ОТЧЕТ

по лабораторной работе №1
по дисциплине «Алгоритмы и структуры данных»
Тема: «Рекурсия»

Студент гр. 8381

Сосновский Д.Н.

Преподаватель

Жангиров Т.Р.

Санкт-Петербург

2019

Цель работы

Ознакомиться с основными понятиями и приёмами рекурсивного программирования, получить навыки программирования рекурсивных процедур и функций на языке программирования C++. Написать программу, использующую рекурсию для решения поставленной задачи.

Задание

Построить синтаксический анализатор для понятия *текст_со_скобками*.

текст_со_скобками ::= элемент | элемент текст_со_скобками
элемент = A | B | (текст_со_скобками) | [текст_со_скобками] |
{текст_со_скобками}

Ход работы

Разработка программы производилась на базе ОС Windows 10 в среде разработки QtCreator с использованием фреймворка Qt.

Для удобства использования программы был разработан GUI с помощью редактора UI – форм, встроенного в используемую среду разработки. GUI представляет собой два текстовых поля – одно для ввода текста, другое для вывода результат проверки, две кнопки – первая кнопка запускает алгоритм проверки, вторая кнопка открывает диалог выбора файла для загрузки из него текста для проверки. В дополнение к этому есть ещё кнопка – метка о том, что вывод необходимо произвести в файл. Для каждой из кнопок был описан слот clicked().

При выборе файла в открывающемся диалоговом окне в текстовом поле для ввода исходного текста появится содержимое выбранного файла.

По нажатию кнопки начала алгоритма проверки в функцию для проверки передаётся указатель на текст, размер текста и номер символа текущего понятия элемента для анализа (в начале он равен 0).

Алгоритм проверки

Изначально проверяется, является ли текущий символ (расположенный по указателю на текст) одним из видов левой скобки. При выполнении этого условия программа вызывает функцию, которая находит этой скобке парную правую скобку, пробегая по имеющемуся тексту. Если парная скобка была найдена, то функция рекурсивно вызывает сама себя, при этом смещая указатель на текст на один элемент вправо и уменьшая длину текста на 2 соответственно. По завершении рекурсивного обхода функция продолжает работу с текстом, при этом изменяя номер символа текущего понятия элемента для анализа (иными словами – левую границу) на номер элемента найденной правой скобки + 1.

Если же символ не является левой скобкой, то вызывается функция, проверяющая конкретно один элемент на корректность. В случае, если элемент является буквой «А» или «В», программа продолжает работу.

Если парная правая скобка не была найдена, программа сгенерирует, обработает и выведет ошибку REDUNDANT_BRACKET.

Если между парными левой и правой скобкой нет других элементов, программа сгенерирует, обработает и выведет ошибку NO_SYMBOL.

Если один элемент при проверке является правой скобкой, программа сгенерирует, обработает и выведет ошибку REDUNDANT_BRACKET, так как согласно с алгоритмом ситуации рассмотрения правой скобки возникать не может.

Если один элемент при проверке не является ни символом «А», ни символом «В», ни правой скобкой – программа сгенерирует, обработает и выведет ошибку INCORRECT_SYMBOL.

В ещё случае пустого поля входного текста программа выведет сообщение, что исходного текста не было найдено. Алгоритм проверки при этом запущен не будет.

Логирование

В течение выполнения программы программа ведет запись каждого шага рекурсии в файл Logs.txt. В данном файле сообщается о глубине рекурсии и о переданном в неё тексте при каждом вызове её из самой себя.

Оценка эффективности

Реализованный алгоритм имеет сложность $O(n^2)$. Это обусловлено тем, что при выделении каждого подтекста для рекурсивного вызова функции в нём будет совершаться в худшем случае $\left(\frac{n}{2}\right)^2$ операций.

Тестирование программы

Программа была успешно протестирована на различных тестах, все данные о тестировании собраны в приложении в разделе «Тестирование»

Вывод

В ходе выполнения данной лабораторной работы был изучен метод рекурсии с применением его в практической задаче – написании программы синтаксического анализатора. Была разработана программа, которая с помощью рекурсивных вызовов самой себя анализирует входную строку на синтаксическое соответствие необходимому понятию.

ПРИЛОЖЕНИЕ

Тестирование

В данном приложении приведена информация о тестировании программы.

The screenshot shows a window titled "Синтаксический анализатор" (Syntax Analyzer). It has a text input field containing "{A} {A(A)}". To the right of the input field are two buttons: "Load from..." and "Write to file" (which is unchecked). Below the input field is a "CHECK SYNTAX" button. Underneath this button, the word "Verdict:" is followed by a text box containing "Correct syntax" in green text.

The screenshot shows the same "Синтаксический анализатор" window. The text input field now contains "{A} {A(ABB))}". The "CHECK SYNTAX" button has been clicked. The "Verdict:" text box now displays an error message in red: "! - Содержится лишняя скобка {A} {A(ABB)".

