Санкт-Петербургский политехнический университет Петра Великого

Институт машиностроения, материалов и транспорта

Высшая школа автоматизации и робототехники

Курсовая работа

Дисциплина: Объектно-ориентированное программирование

Тема: Обратная Польская запись

Студент гр. 3331506/00401 Савицкий И.В.

Преподаватель Ананьевский М.С.

Санкт-Петербург

2023

Оглавление

1. Введение

Алгоритм обратной польской записи – алгоритм представления математических и логических выражений, в котором операнды расположены перед знаками операций.

Обратная польская нотация была разработана австралийским философом и специалистом в области теории вычислительных машин Чарльзом Хэмблином в середине 1950-х на основе польской нотации, которая была предложена в 1920 году польским математиком Яном Лукасевичем. Работа Хэмблина была представлена на конференции в июне 1957, и издана в 1957 и 1962.

Отличительной особенностью обратной польской нотации является то, что все аргументы (или операнды) расположены перед знаком операции. В общем виде запись выглядит следующим образом:

- Запись набора операций состоит из последовательности операндов и знаков операций. Операнды в выражении при письменной записи разделяются пробелами.
- Выражение читается слева направо. Когда в выражении встречается знак операции, выполняется соответствующая операция над двумя последними встретившимися перед ним операндами в порядке их записи. Результат операции заменяет в выражении последовательность её операндов и её знак, после чего выражение вычисляется дальше по тому же правилу.
- Результатом вычисления выражения становится результат последней вычисленной операции.

В рамках данной курсовой работы будет рассмотрена реализация алгоритма на языке программирования C++ с использованием методов объектно-ориентированного программирования

2. Основная часть

2.1. Принцип работы

Вычисление выражений в обратной польской нотации основана на использовании стека. Алгоритм вычисления:

1. Обработка входного символа:
- Если на вход подан операнд, он помещается на вершину стека.
- Если на вход подан знак операции, то соответствующая операция выполняется над требуемым количеством значений, извлечённых из стека, взятых в порядке добавления. Результат выполненной операции кладётся на вершину стека.

2. Если входной набор символов обработан не полностью, перейти к шагу 1.

3. После полной обработки входного набора символов результат вычисления выражения лежит на вершине стека.

Рассмотрим алгоритм на примере выражения:

$$(8 + 2 * 5)/(1 + 3 * 2 - 4)$$

Соответствующая формула в обратной польской записи выглядит так:

$$8\ 2\ 5\ *\ +\ 1\ 3\ 2\ *\ +\ 4\ -\ /$$

Число на вершине стека – это правый операнд (а не левый). Это очень важно для операций деления, вычитания и возведения в степень, поскольку порядок следования операндов в данном случае имеет значение (в отличие от операций сложения и умножения). Другими словами, операция деления действует следующим образом: сначала в стек помещается числитель, потом знаменатель, и тогда операция даёт правильный результат. Отметим, что преобразовать обратную польскую запись в машинный код очень легко: нужно просто двигаться по формуле в обратной польской записи, записывая по одной команде для каждого символа. Если символ является константой или переменной, нужно вписывать команду помещения этой константы или переменной в стек, если символ является оператором, нужно вписывать команду выполнения это операции.

2.2. Реализация

Организация узла стека:

data – хранимые данные

ptr_next – указатель на следующий узел

Основные операции:

1. Добавление элемента в стек – передача в функцию push ссылки на вершину стека и добавляемого элемента, создание нового узла, присвоение data значения нового элемента, присвоение указателю на следующий узел указатель на текущую вершину стека, смена указателя на вершину стека на текущий узел.

```
void push(struct stack_over** top_ptr, const unsigned long long int symbol)
{
    struct stack_over* new_ptr = NULL;

    new_ptr = (struct stack_over*)malloc(sizeof(struct stack_over));

    if (new_ptr != NULL)
    {
        memset(new_ptr, 0x00, sizeof(struct stack_over));
        new_ptr->data = symbol;
        new_ptr->ptr_next = *top_ptr;
        *top_ptr = new_ptr;
    }
}
```

2. Удаление элемента из стека – передача в функцию pop ссылки на вершину стека, взятие значения элемента, хранимого в узле вершины стека, смена указателя на вершину стека на указатель на следующий узел, функция возвращает взятое значение элемента.

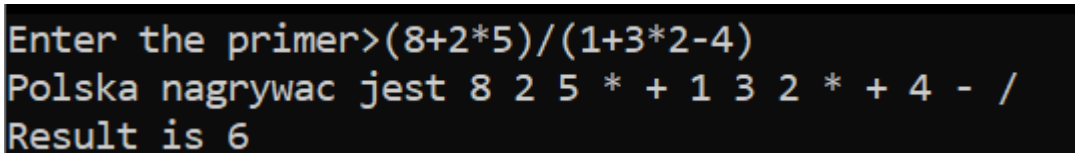
```
unsigned long long int pop(struct stack_over** top_ptr)
{
    unsigned long long int pop_symbol = '\0';

    if (*top_ptr == NULL)
    {
        return EXIT_SUCCESS;
    }

    pop_symbol = (*top_ptr)->data;
    *top_ptr = (*top_ptr)->ptr_next;
    return pop_symbol;
}
```

Примеры работы программы

В пункте 2.2 рассматривалось выражение, рассмотрим работу программы на его примере:



Результат работы программы сходится с ожидаемым.

3. Заключение

Выводы – в результате выполнения курсовой работы по изучению алгоритма обратной польской записи и его применения в расчёте выражений можно отметить:

1. Из-за отсутствия скобок обратная польская запись короче инфиксной. В программируемых устройствах сокращается объём тех частей программы, которые описывают вычисления, что важно для портативных и встроенных вычислительных устройств, имеющих жёсткие ограничения на объём памяти.

2. В отличие от инфиксной записи в обратной польской записи нельзя использовать одни и те же знаки для унарных и бинарных операций, поэтому при использовании унарного плюса или минуса нужно либо добавлять ноль в стек или использовать другой символ.

3. Для того чтобы обрабатывать некоторые случаи, нужно совершать предобработку входного выражения. Например, для правильного вычисления случая 2^3^2 необходимо сначала следующим способом обработать выражение: 2^(3^2).

Список литературы

1. Вирт, Н. Алгоритмы и структуры данных : с примерами на Паскале / Н. Вирт ; Никлаус Вирт ; [пер. с англ. Д. Б. Подшивалова]. – [2-е изд., испр.]. – Санкт-Петербург : Невский диалект, 2005. – 351 с. – (Библиотека программиста). – ISBN 5-7940-0065-1. – EDN QMOTFX.

2. Расулов, В. Е. Алгоритм преобразования арифметических выражений в обратную польскую запись / В. Е. Расулов, Ю. М. Рахимова // Новые технологии - нефтегазовому региону : материалы Международной научно-практической конференции, Тюмень, 16–20 мая 2016 года. Том III. – Тюмень: Тюменский индустриальный университет, 2016. – С. 23-25. – EDN VYCHHX.

3. Ахо, Альфред В. Структуры данных и алгоритмы / Альфред В. Ахо, Джон Э. Хопкрофт, Джеффри Д. Ульман ; [пер. с англ. и ред. А. А. Минько]. - Москва [и др.] : Вильямс, 2010 (Санкт-Петербург : Печатный двор им. А. М. Горького). - 391 с. : ил.; 23 см.; ISBN 978-5-8459-1610-5

Приложение

Код программы:

https://github.com/soomrack/MR2022/blob/main/Savitsky%20I.V/Coursework/PolskaNagrywac.c