Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Московский государственный технический университет имени Н.Э. Баумана
(национальный исследовательский университет)»
(МГТУ им. Н.Э. Баумана)

ФАКУЛЬТЕТ _____ИНФОРМАТИКА И СИСТЕМЫ УПРАВЛЕНИЯ_____

КАФЕДРА _____КОМПЬЮТЕРНЫЕ СИСТЕМЫ И СЕТИ_____

НАПРАВЛЕНИЕ ПОДГОТОВКИ 09.03.01 Компьютерные системы и сети.

О Т Ч Е Т

по домашней работе № __2__

Дисциплина: ___Машинно-зависимые языки и основы компиляции_____

Название домашней работы: _____Лексические и синтаксические__

_анализаторы_____

Студент гр. ИУ6-42Б _____ ____**Медведев_А.Е._**
(Подпись, дата) (И.О. Фамилия)

Преподаватель _____ _____
(Подпись, дата) (И.О. Фамилия)

Москва, 2021

Задание:

Домашнее задание №2. Лексические и синтаксические анализаторы.
Разработать грамматику и распознаватель выражений языка программирования C++, оперирующих вещественными числами в формате с плавающей точкой. Например:
-34.3456E-6 + 0.56E+3* 0.7989E2

Цель работы:

Закрепление знаний теоретических основ и основных методов приемов разработки лексических и синтаксических анализаторов регулярных и контекстносвободных формальных языков

Описание грамматики в форме Бэкуса–Наура, с указанием типа грамматики:

<дробное знаковое число> ::= <знак><целое без знака><точка><целое без знака>|
<знак><целое без знака><точка><целое без знака><Символ степени> <знак>
<целое без знака> | <целое без знака><точка><целое без знака> | <целое без знака><точка><целое без знака><Символ степени><знак><целое без знака>,
<целое без знака> ::= <цифра><целое без знака>|<цифра>,
<цифра > ::= 0|1|2|3|4|5|6|7|8|9,
<знак> ::= +| -,
<Символ степени> ::= E,
<точка> ::= .

Так как у нас есть конструкции, которые вызывают сами себя, то решать поставленную задачу лучше всего методом рекурсивного спуска.

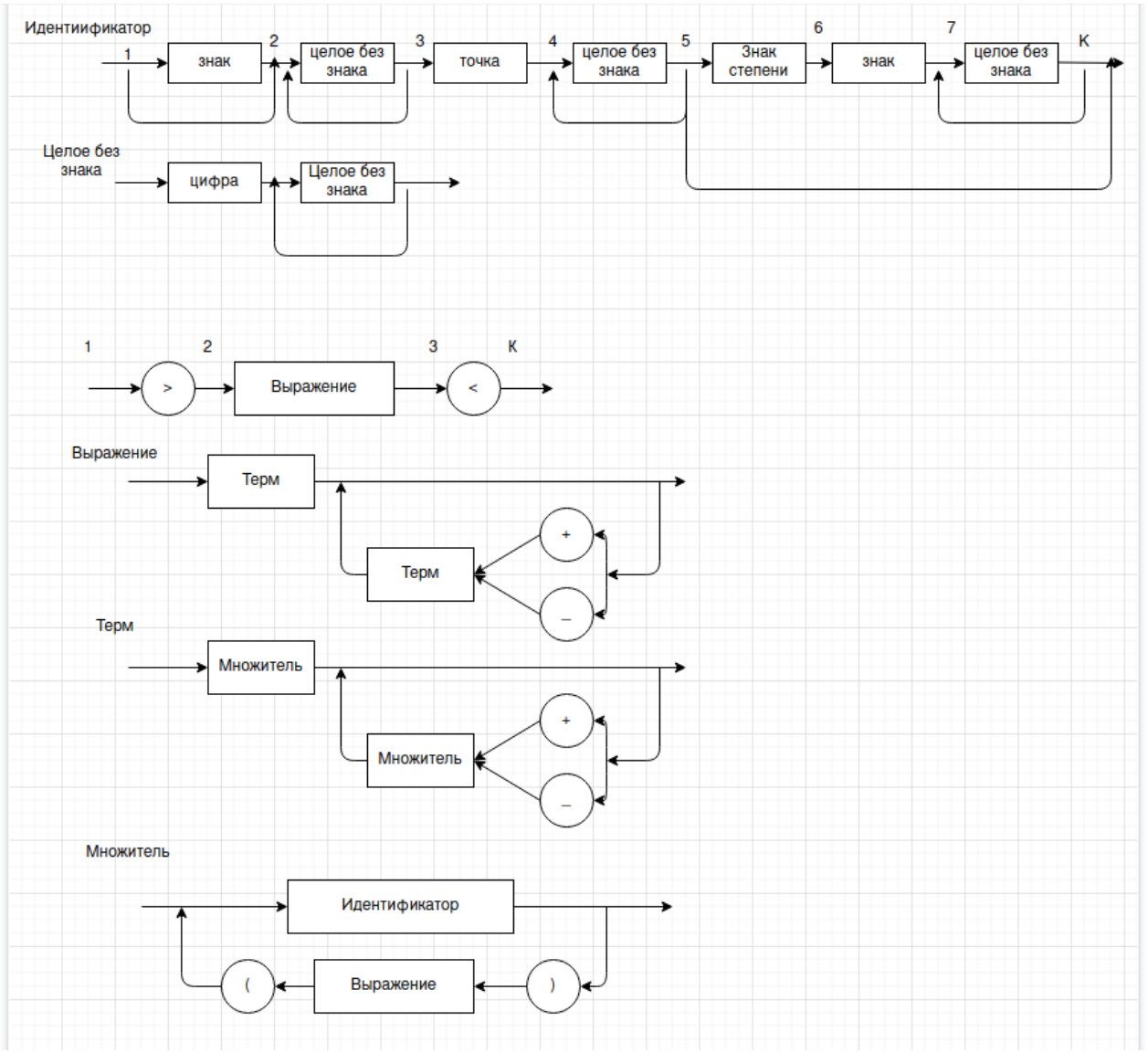

Рисунок 1 - Схема алгоритма рекурсивного спуска

Таблица - конечный автомат

	Знак	Цифра	Точка	Символ степени	Другое
0	1	1	-1	-1	-2
1	9	1	2	-1	9
2	-1	3	-1	-1	-2
3	9	3	-1	4	9
4	5	-1	-1	-1	-2
5	-1	6	-1	-1	-2
6	-1	6	-1	-1	9

Код программы:

```
#include <iostream>
#include <algorithm>
#include <string>

bool id(std::string &str);
bool mult(std::string &str);
bool term(std::string &str);
bool culc(std::string &str);

int main()
{
   std::string str;
   getline(std::cin, str, '\n');
   bool answer = true;
   str.erase(remove_if(str.begin(), str.end(), isspace), str.end());
   while (str.length() > 0 && answer)
   {
      answer = culc(str);
      if (answer && str.length() == 0)
         std::cout << "Yes";
      else
         std::cout << "No";
      getline(std::cin, str, '\n');
      str.erase(remove_if(str.begin(), str.end(), isspace), str.end());
      answer = true;
   }

   return 0;
}

bool id(std::string &str)
{
   int state[7][5] = {
   //-  1  .  e  an
/*0*/ 1, 1,-1,-1,-2,
/*1*/ 9, 1, 2,-1, 9,
/*2*/-1, 3,-1,-1,-2,
/*3*/ 9, 3,-1, 4, 9,
/*4*/ 5,-1,-1,-1,-2,
/*5*/-1, 6,-1,-1,-2,
/*6*/-1, 6,-1,-1, 9
   };
```

```
int state_now = 0;

std::string s;
int col = 0;
while ((state_now >= 0) && (state_now != 9) && (str.length() != 0))
{
   if (str[0] == '-' || str[0] == '+') col = 0;
   else if (str[0] >= '0' && str[0] <= '9') col = 1;
   else if (str[0] == '.') col = 2;
   else if (str[0] == 'E') col = 3;
   else col = 4;

   state_now = state[state_now][col];
   if (state_now != -1 && state_now != 9)
   {
      s += str[0];
      str.erase(0, 1);
   }
}
if (str.length() == 0 && (state_now == 6 || state_now == 3))
   state_now = 9;
if (state_now == 9 && s.length() != 0)
{
   std::cout << "Identify = " << s << "\n";
   return true;
}
else
{
   if (state_now == -2)
   {
      if (str[0] == '+' || str[0] == '-' ||
         str[0] == '(' || str[0] == ')' ||
         str[0] == '*' || str[0] == '/')
      {
         std::cout << "Identify = " << s << "\n";
         return true;
      }
   }
   if (state_now == -1)
   {
      std::cout << "Wrong symbol *" << str[0] << "*\n";
   }
   else
   {
      std::cout << "Identifier waits..." << str << "\n";
```

```
        }
        return false;
    }

    return true;
}


bool mult(std::string &str)
{
    bool answer = false;
    if (str[0] == '(')
    {
        str.erase(0, 1);
        answer = culc(str);
        if (answer && str[0] == ')')
            str.erase(0, 1);
        else
            std::cout << "error: " << str << "\n";
    }
    else answer = id(str);
    return answer;
}

bool term(std::string &str)
{
    bool answer = false;
    answer = mult(str);
    if (answer)
    {
        while ((str[0] == '*' || str[0] == '/') && answer)
        {
            str.erase(0, 1);
            answer = mult(str);
        }
    }
    return answer;
}
bool culc(std::string &str)
{
    bool answer = false;
    answer = term(str);
    if (answer)
    {
        while ((str[0] == '+' || str[0] == '-') && answer)
```

```
        {
            str.erase(0, 1);
            answer = term(str);
        }
    }
    return answer;
}
```

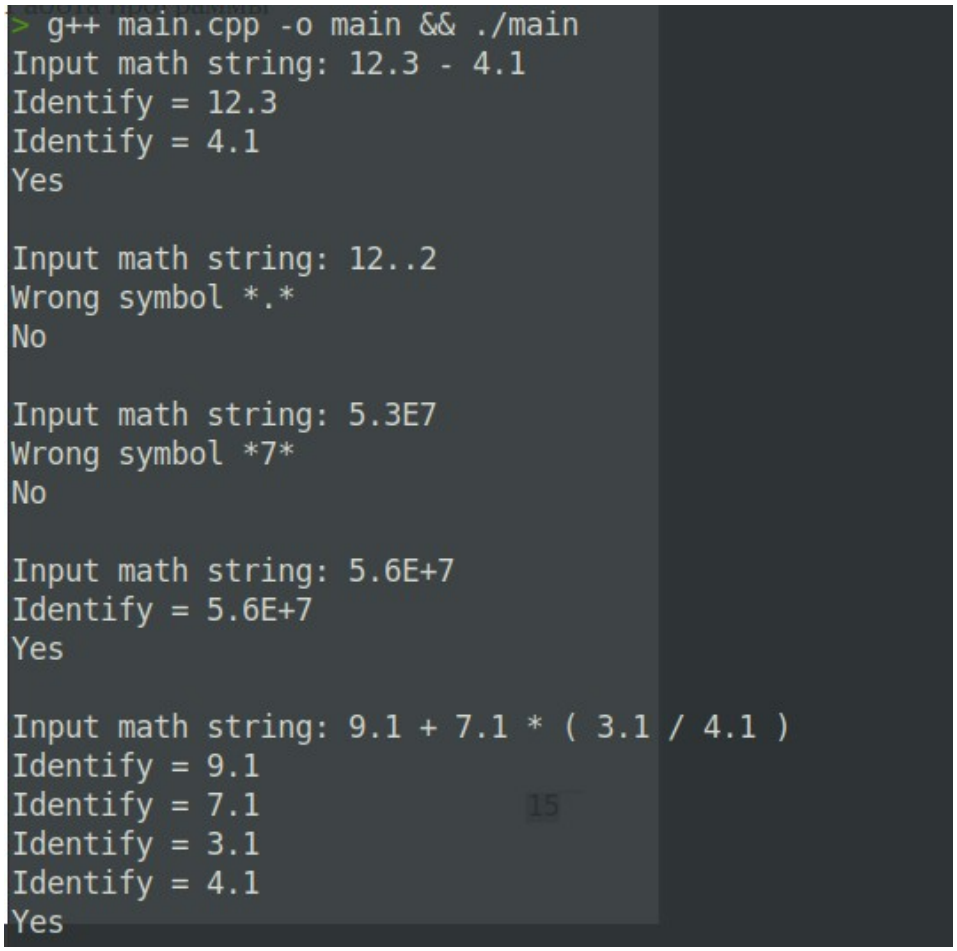

Рисунок 1 - тесты

Таблица тестов

Ввод	Ожидалось	Вывод
Input math string: 12.3 - 4.1	Identify = 12.3 Identify = 4.1 Yes	Identify = 12.3 Identify = 4.1 Yes
Input math string: 12..2	Wrong symbol *.* No	Wrong symbol *.* No
Input math string: 5.3E7	Wrong symbol *7* No	Wrong symbol *7* No
Input math string: 5.6E+7	Identify = 5.6E+7 Yes	Identify = 5.6E+7 Yes

Input math string: 9.1 + 7.1 * (3.1 / 4.1)	Identify = 9.1 Identify = 7.1 Identify = 3.1 Identify = 4.1 Yes	Identify = 9.1 Identify = 7.1 Identify = 3.1 Identify = 4.1 Yes

Контрольные вопросы:

1. Дайте определение формального языка и формальной грамматики.

Формальным языком L в алфавите A называют произвольное подмножество множества A*. Таким образом, язык определяется как множество допустимых предложений.

Формальная грамматика – это математическая система, определяющая язык посредством порождающих правил – правил продукции. Она определяется как четверка: G = (VT, VN, P, S),

2. Как определяется тип грамматики по Хомскому?

– Тип 0, грамматики фразовой структуры или грамматики «без ограничений»: в таких грамматиках допустимо наличие любых правил вывода, что свойственно грамматикам естественных языков;

– Тип 1, контекстно-зависимые (неукорачивающие) грамматики:в этих грамматиках для правил вида α X β → α x β возможность подстановки строки x вместо символа X определяется присутствием контекста, что также свойственно грамматикам естественных языков;

– Тип 2, контекстно-свободные грамматики: подстановки не зависят от контекста;

– Тип 3, регулярная грамматика: расширение допускает не более одного правила вида S → e, но в этом случае аксиома не должна появляться в правых частях правил

3. Поясните физический смысл и обозначения формы Бэкуса–Наура.

Форма Бэкуса-Наура (БНФ). БНФ связывает терминальные и нетерминальные символы, используя две операции: «::=» – «можно

заменить на»; «|» – «или». Основное достоинство – группировка правил, определяющих каждый нетерминал. Нетерминалы при этом записываются в угловых скобках

4. Что такое лексический анализ? Какие методы выполнения лексического анализа вы
знаете?

Лексический анализ – процесс преобразования исходного текста в строку однородных символов. Каждый символ результирующей строки – токен соответствует слову языка – лексеме и характеризуется набором атрибутов, таких как тип, адрес и т. п., поэтому строку токенов часто представляют таблицей, строка которой соответствует одному токену.

Методы:

а) При помощи конечных автоматов

б) Восходящий анализ

в) Низходящий анализ

5. Что такое синтаксический анализ? Какие методы синтаксического анализа вы знаете?
К каким грамматикам применяются перечисленные вами методы?

Синтаксический анализ – процесс распознавания конструкций языка в строке токенов. Главным результатом является информация об ошибках в выражениях, операторах и описаниях программы. Построение конечного автомата, получающего на вход строку лексем

Метод рекурсивного спуска. Метод рекурсивного спуска основывается на синтаксических диаграммах языка. Согласно этому методу для каждого нетерминала разрабатывают рекурсивную процедуру. Основная программа вызывает процедуру аксиомы, которая вызывает процедуры нетерминалов, упомянутые в правой части аксиомы и т. д. В эти же процедуры встраивают семантическую обработку распознанных конструкций.

6. Что является результатом лексического анализа?

При лексическом разборе будет получена таблица токенов и, возможно, расширены таблицы переменных и литералов

7. Что является результатом синтаксического анализа?

Результатом является информация об ошибках в выражениях, операторах и описаниях программы.

8. В чем заключается метод рекурсивного спуска?

Метод рекурсивного спуска основывается на синтаксических диаграммах языка. Согласно этому методу для каждого нетерминала разрабатывают рекурсивную процедуру. Основная программа вызывает процедуру аксиомы, которая вызывает процедуры нетерминалов, упомянутые в правой части аксиомы и т. д. В эти же процедуры встраивают семантическую обработку распознанных конструкций.

9. Что такое таблица предшествования и для чего она строится?

В таблицу сводятся отношения предшествования терминалов (знаков операций), полученных с учетом приоритетов операций. После чего возможно будет четко определить начало или конец основы

10. Как с использованием таблицы предшествования осуществляют синтаксический
анализ?

Из входной строки выделяются символы операций, после чего в соответствии с таблицей для пар символов определяются отношения, чтобы затем свертывать выражение в соответствии с отношениями и заменить на результат. Эта операция повторяется в цикле пока не будет найдена ошибка или строка не будет преобразована.

Вывод: В ходе выполнения домашнего задания была создана программа распознающая строку математического выражения дробных чисел языка c++, используя метот рекурсивного спуска.