

Министерство науки и высшего образования Российской Федерации
Федеральное государственное автономное образовательное учреждение
высшего образования
САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
АЭРОКОСМИЧЕСКОГО ПРИБОРОСТРОЕНИЯ

КАФЕДРА КОМПЬЮТЕРНЫХ ТЕХНОЛОГИЙ И ПРОГРАММНОЙ ИНЖЕНЕРИИ
(КАФЕДРА № 43)

ОТЧЁТ
ЗАЩИЩЁН С ОЦЕНКОЙ
ПРЕПОДАВАТЕЛЬ

профессор

должность, уч. степень, звание

подпись, дата

Гвоздик М.И.

инициалы, фамилия

ОТЧЁТ О ЛАБОРАТОРНОЙ РАБОТЕ №4

Нисходящая трансляция на основе вычисления синтезируемых атрибутов

по дисциплине: Методы трансляции

РАБОТУ ВЫПОЛНИЛ

СТУДЕНТ ГР. №

4731

подпись, дата

Е. Д. Богачев

инициалы, фамилия

Санкт-Петербург 2019

Выполните программную реализацию решения задачи, выбранной по номеру варианта, в виде синтаксически управляемой нисходящей трансляции с использованием наследуемых и синтезируемых атрибутов.

4	$S ::= E$ $E ::= E + T$ $E ::= T$ $T ::= T * F$ $T ::= F$ $F ::= x$ $F ::= 5$ $F ::= 6$ $F ::= (E)$ S – аксиома	Синтаксически управляемое дифференцирование алгебраического выражения
---	--	---

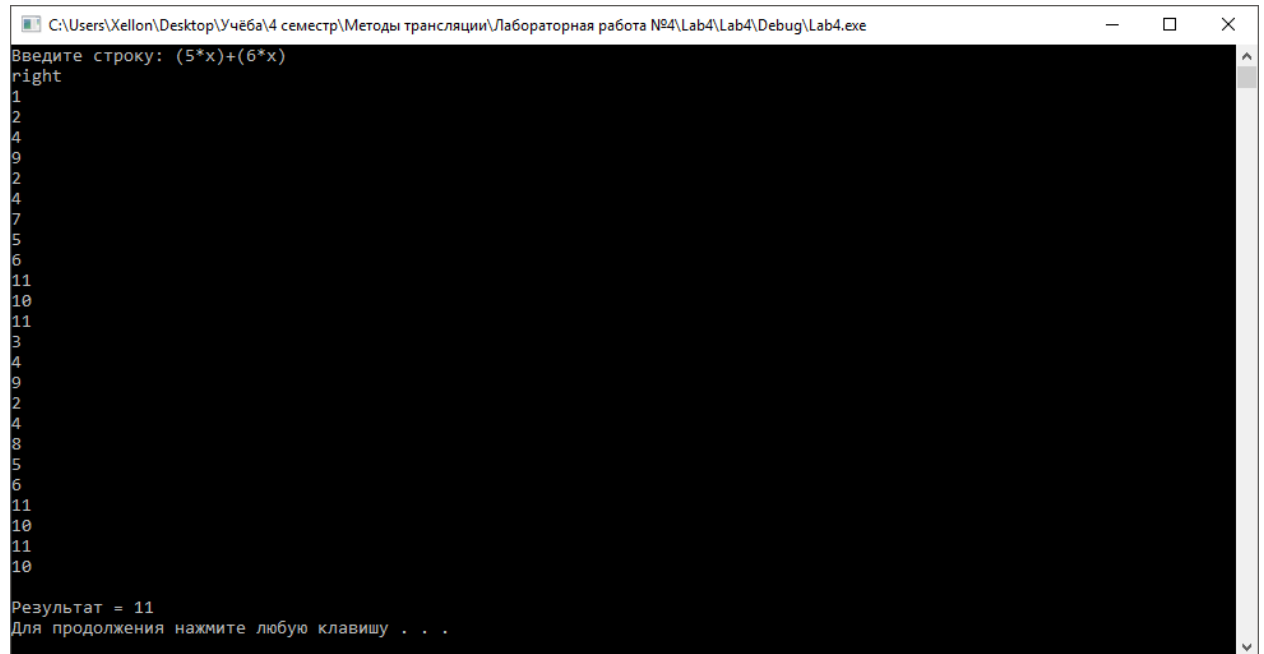
1. $S \rightarrow E$
2. $E \rightarrow TZ$ $\{T_i = T_s\}$
3. $Z \rightarrow +TZ$ $\{T_i = " + "\}$
4. $T \rightarrow FH$ $\{F_i = F_s; T_s = F_s\}$
5. $H \rightarrow *FH$ $\{F_i = " * "\}$
6. $F \rightarrow x$
7. $F \rightarrow 5$
8. $F \rightarrow 6$ $\{E_i = F_s\}$
9. $F \rightarrow (E)$
10. $Z \rightarrow ^\wedge$
11. $H \rightarrow ^\wedge$

[illegible]

Вывод:

Реализовал программное решение задачи, выбранной по номеру варианта, в виде синтаксически управляемой нисходящей трансляции с использованием наследуемых и синтезируемых атрибутов.

Листинг программы



```
C:\Users\Xellon\Desktop\Учеба\4 семестр\Методы трансляции\Лабораторная работа №4\Lab4\Debug\Lab4.exe
Введите строку: (5*x)+(6*x)
right
1
2
4
9
2
4
7
5
6
11
10
11
3
4
9
2
4
8
5
6
11
10
11
10
Результат = 11
Для продолжения нажмите любую клавишу . . .
```

Код:

```
#include <regex>
#include <iostream>
#include <sstream>
#include <map>
#include <string>
#include <windows.h>
using namespace std;

struct Node                                     //Структура,
являющаяся звеном стека
{
    char symb;
    Node *Next;
};

class Stack                                     //Создаем тип данных
Стек
{
    Node *Head;                                //Указатели на адреса
    начала стека
public:
    Stack() :Head(NULL) {};                    //Инициализируем адреса
    как пустые
    void Show();                               //Прототип функции
    отображения стека на экране
    void Creat(char symb);                     //Прототип функции
    добавления элементов в список
    void Del();                                //Прототип функции
    удаления элемента в списке
    char Ret();                                //Прототип функции
    вывода символа из стека
};

void Stack::Creat(char symb)                    //Функция заполнения стека
{
    Node *temp = new Node;                     //Выделение памяти под
    новый элемент структуры
    temp->symb = symb;                          //Записываем значение в
    структуру
    temp->Next = Head;                          //Сдвигаем указатель головы
    Head = temp;
}

char Stack::Ret()
{
    char ret;
    Node *temp = Head;
    ret = Head->symb;
    return ret;
}
```

```

void Stack::Show()                                     //Функция вывода стека
{
    Node *temp = Head;
    while (temp != NULL)                               //Пока не встретим пустое значение
    {
        cout << temp->symb << " ";                   //Выводим каждое считанное значение на
        экран
        temp = temp->Next;                             //Смена адреса на адрес
        следующего элемента
    }
    cout << "\n";
}

void Stack::Del()                                     //Функция удаления элемента стека
{
    Node *temp = Head;
    Head = temp->Next;
    delete temp;
}

void LL1(string str3, int array[14][8], Stack st)
{
    int el = 0, x = 0, y = 0;
    st.Creat('$');
    st.Creat('D');
    while (el < str3.size())
    {
        if (str3[el] == 'x')
        {
            y = 0;
            if (st.Ret() == 'S')
            {
                x = 0;
            }
            else if (st.Ret() == 'E')
            {
                x = 1;
            }
            else if (st.Ret() == 'Z')
            {
                x = 2;
            }
            else if (st.Ret() == 'T')
            {
                x = 3;
            }
            else if (st.Ret() == 'H')
            {
                x = 4;
            }
            else if (st.Ret() == 'F')

```

```

        {
            x = 5;
        }
        else if (st.Ret() == 'x')
        {
            x = 6;
        }
        else if (st.Ret() == '5')
        {
            x = 7;
        }
        else if (st.Ret() == '6')
        {
            x = 8;
        }
        else if (st.Ret() == '+')
        {
            x = 9;
        }
        else if (st.Ret() == '*')
        {
            x = 10;
        }
        else if (st.Ret() == '(')
        {
            x = 11;
        }
        else if (st.Ret() == ')')
        {
            x = 12;
        }
        else if (st.Ret() == '$')
        {
            x = 13;
        }
    }
    if (str3[el] == '5')
    {
        y = 1;
        if (st.Ret() == 'S')
        {
            x = 0;
        }
        else if (st.Ret() == 'E')
        {
            x = 1;
        }
        else if (st.Ret() == 'Z')
        {
            x = 2;
        }
        else if (st.Ret() == 'T')

```

```

        {
            x = 3;
        }
    else if (st.Ret() == 'H')
    {
        x = 4;
    }
    else if (st.Ret() == 'F')
    {
        x = 5;
    }
    else if (st.Ret() == 'x')
    {
        x = 6;
    }
    else if (st.Ret() == '5')
    {
        x = 7;
    }
    else if (st.Ret() == '6')
    {
        x = 8;
    }
    else if (st.Ret() == '+')
    {
        x = 9;
    }
    else if (st.Ret() == '*')
    {
        x = 10;
    }
    else if (st.Ret() == '(')
    {
        x = 11;
    }
    else if (st.Ret() == ')')
    {
        x = 12;
    }
    else if (st.Ret() == '$')
    {
        x = 13;
    }
}
if (str3[el] == '6')
{
    y = 2;
    if (st.Ret() == 'S')
    {
        x = 0;
    }
    else if (st.Ret() == 'E')

```

```

{
    x = 1;
}
else if (st.Ret() == 'Z')
{
    x = 2;
}
else if (st.Ret() == 'T')
{
    x = 3;
}
else if (st.Ret() == 'H')
{
    x = 4;
}
else if (st.Ret() == 'F')
{
    x = 5;
}
else if (st.Ret() == 'x')
{
    x = 6;
}
else if (st.Ret() == '5')
{
    x = 7;
}
else if (st.Ret() == '6')
{
    x = 8;
}
else if (st.Ret() == '+')
{
    x = 9;
}
else if (st.Ret() == '*')
{
    x = 10;
}
else if (st.Ret() == '(')
{
    x = 11;
}
else if (st.Ret() == ')')
{
    x = 12;
}
else if (st.Ret() == '$')
{
    x = 13;
}
}

```



```
if (str3[el] == '+')
{
    y = 3;
    if (st.Ret() == 'S')
    {
        x = 0;
    }
    else if (st.Ret() == 'E')
    {
        x = 1;
    }
    else if (st.Ret() == 'Z')
    {
        x = 2;
    }
    else if (st.Ret() == 'T')
    {
        x = 3;
    }
    else if (st.Ret() == 'H')
    {
        x = 4;
    }
    else if (st.Ret() == 'F')
    {
        x = 5;
    }
    else if (st.Ret() == 'x')
    {
        x = 6;
    }
    else if (st.Ret() == '5')
    {
        x = 7;
    }
    else if (st.Ret() == '6')
    {
        x = 8;
    }
    else if (st.Ret() == '+')
    {
        x = 9;
    }
    else if (st.Ret() == '*')
    {
        x = 10;
    }
    else if (st.Ret() == '(')
    {
        x = 11;
    }
    else if (st.Ret() == ')')
```

```

        {
            x = 12;
        }
        else if (st.Ret() == '$')
        {
            x = 13;
        }
    }
    if (str3[el] == '*')
    {
        y = 4;
        if (st.Ret() == 'S')
        {
            x = 0;
        }
        else if (st.Ret() == 'E')
        {
            x = 1;
        }
        else if (st.Ret() == 'Z')
        {
            x = 2;
        }
        else if (st.Ret() == 'T')
        {
            x = 3;
        }
        else if (st.Ret() == 'H')
        {
            x = 4;
        }
        else if (st.Ret() == 'F')
        {
            x = 5;
        }
        else if (st.Ret() == 'x')
        {
            x = 6;
        }
        else if (st.Ret() == '5')
        {
            x = 7;
        }
        else if (st.Ret() == '6')
        {
            x = 8;
        }
        else if (st.Ret() == '+')
        {
            x = 9;
        }
        else if (st.Ret() == '*')
    }

```

```

        {
            x = 10;
        }
    else if (st.Ret() == '(')
    {
        x = 11;
    }
    else if (st.Ret() == ')')
    {
        x = 12;
    }
    else if (st.Ret() == '$')
    {
        x = 13;
    }
}
if (str3[el] == '(')
{
    y = 5;
    if (st.Ret() == 'S')
    {
        x = 0;
    }
    else if (st.Ret() == 'E')
    {
        x = 1;
    }
    else if (st.Ret() == 'Z')
    {
        x = 2;
    }
    else if (st.Ret() == 'T')
    {
        x = 3;
    }
    else if (st.Ret() == 'H')
    {
        x = 4;
    }
    else if (st.Ret() == 'F')
    {
        x = 5;
    }
    else if (st.Ret() == 'x')
    {
        x = 6;
    }
    else if (st.Ret() == '5')
    {
        x = 7;
    }
    else if (st.Ret() == '6')

```

```

        {
            x = 8;
        }
        else if (st.Ret() == '+')
        {
            x = 9;
        }
        else if (st.Ret() == '*')
        {
            x = 10;
        }
        else if (st.Ret() == '(')
        {
            x = 11;
        }
        else if (st.Ret() == ')')
        {
            x = 12;
        }
        else if (st.Ret() == '$')
        {
            x = 13;
        }
    }
    if (str3[el] == ')')
    {
        y = 6;
        if (st.Ret() == 'S')
        {
            x = 0;
        }
        else if (st.Ret() == 'E')
        {
            x = 1;
        }
        else if (st.Ret() == 'Z')
        {
            x = 2;
        }
        else if (st.Ret() == 'T')
        {
            x = 3;
        }
        else if (st.Ret() == 'H')
        {
            x = 4;
        }
        else if (st.Ret() == 'F')
        {
            x = 5;
        }
        else if (st.Ret() == 'x')
    }

```

```

        {
            x = 6;
        }
        else if (st.Ret() == '5')
        {
            x = 7;
        }
        else if (st.Ret() == '6')
        {
            x = 8;
        }
        else if (st.Ret() == '+')
        {
            x = 9;
        }
        else if (st.Ret() == '*')
        {
            x = 10;
        }
        else if (st.Ret() == '(')
        {
            x = 11;
        }
        else if (st.Ret() == ')')
        {
            x = 12;
        }
        else if (st.Ret() == '$')
        {
            x = 13;
        }
    }
    if (str3[el] == '$')
    {
        y = 7;
        if (st.Ret() == 'S')
        {
            x = 0;
        }
        else if (st.Ret() == 'E')
        {
            x = 1;
        }
        else if (st.Ret() == 'Z')
        {
            x = 2;
        }
        else if (st.Ret() == 'T')
        {
            x = 3;
        }
        else if (st.Ret() == 'H')

```

```

        {
            x = 4;
        }
        else if (st.Ret() == 'F')
        {
            x = 5;
        }
        else if (st.Ret() == 'x')
        {
            x = 6;
        }
        else if (st.Ret() == '5')
        {
            x = 7;
        }
        else if (st.Ret() == '6')
        {
            x = 8;
        }
        else if (st.Ret() == '+')
        {
            x = 9;
        }
        else if (st.Ret() == '*')
        {
            x = 10;
        }
        else if (st.Ret() == '(')
        {
            x = 11;
        }
        else if (st.Ret() == ')')
        {
            x = 12;
        }
        else if (st.Ret() == '$')
        {
            x = 13;
        }
    }

    if (array[x][y] == 0)
    {
        system("cls");
        cout << "Ошибка. Неподходящие слово" << endl;
        break;
    }
    else if (array[x][y] == 1)
    {
        cout << "1" << endl;
        st.Del();
        st.Creat('E');
    }

```

```

    }
    else if (array[x][y] == 2)
    {
        cout << "2" << endl;
        st.Del();
        st.Creat('Z');
        st.Creat('T');
    }
    else if (array[x][y] == 3)
    {
        cout << "3" << endl;
        st.Del();
        st.Creat('Z');
        st.Creat('T');
        st.Creat('+');
    }
    else if (array[x][y] == 4)
    {
        cout << "4" << endl;
        st.Del();
        st.Creat('H');
        st.Creat('F');
    }
    else if (array[x][y] == 5)
    {
        cout << "5" << endl;
        st.Del();
        st.Creat('H');
        st.Creat('F');
        st.Creat('*');
    }
    else if (array[x][y] == 6)
    {
        cout << "6" << endl;
        st.Del();
        st.Creat('x');
    }
    else if (array[x][y] == 7)
    {
        cout << "7" << endl;
        st.Del();
        st.Creat('5');
    }
    else if (array[x][y] == 8)
    {
        cout << "8" << endl;
        st.Del();
        st.Creat('6');
    }
    else if (array[x][y] == 9)
    {
        cout << "9" << endl;

```

```

        st.Del();
        st.Creat('');
        st.Creat('E');
        st.Creat('(');
    }
    else if (array[x][y] == 10)
    {
        cout << "10" << endl;
        st.Del();
    }
    else if (array[x][y] == 11)
    {
        cout << "11" << endl;
        st.Del();
    }
    else if (array[x][y] == 88)
    {
        st.Del();
        el++;
    }
    else if (array[x][y] == 100)
    {
        st.Del();
        el++;
    }
}
}

```

```

string Ans(string polynomial) {
    string s = polynomial;
    int z1 = 0;
    int z2 = 0;
    for (int i = 0; i <= s.length(); i++)
    {
        if (s[i] == '(')
        {
            z1++;
        }
        else if (s[i] == ')')
        {
            z2++;
        }
    }
    while (z1 > 0)
    {
        s.erase(s.find('('), 1);
        z1--;
    }
    while (z2 > 0)
    {
        s.erase(s.find(')'), 1);
    }
}

```



```

        z2--;
    }
    // Ключ - степень икса, значение - коэффициент при икс
    map<int, int> poly;
    map<int, int> dxpoly;
    if (polynomial[0] != '-') s = '+' + s;
    const regex r("[+|-]\\d+)([\\*]x([\\^](\\d+))?)?");
    const regex rr("[+|-])x");
    const string fmt("$011*x");
    s = regex_replace(s, rr, fmt);

    for (sregex_iterator i(s.begin(), s.end(), r), end; i != end; ++i) {
        //cout << " " << (*i)[0] << "\t" << (*i)[1] << "\t" << (*i)[2] << "\t" << (*i)[3] <<
        "\t" << (*i)[4] << endl;
        // Только число. Коэффициент при x^0
        if ((*i)[2] == "") {
            int an = atoi((*i)[1].str().c_str());
            poly[0] += an;
            // x^1
        }
        else if ((*i)[2] == "*x") {
            int an = atoi((*i)[1].str().c_str());
            poly[1] += an;
        }
        else {
            int an = atoi((*i)[1].str().c_str());
            int xn = atoi((*i)[4].str().c_str());
            poly[xn] += an;
        }
    }
    // Derivate
    for (auto it = poly.begin(); it != poly.end(); ++it) {
        dxpoly[it->first - 1] = it->second * it->first;
    }
    // Grap result
    stringstream ss;
    for (auto it = dxpoly.rbegin(); it != dxpoly.rend(); ++it) {
        if (it->first < 0) continue;
        if (it->first == 0) {
            ss << showpos << it->second;
        }
        else if (it->first == 1) {
            if (abs(it->second) == 1) ss << (it->second > 0 ? "+x" : "-x");
            else ss << showpos << it->second << "*x";
        }
        else {
            if (abs(it->second) == 1) ss << (it->second > 0 ? "+x" : "-x") << "^" <<
            noshowpos << it->first;
            else ss << showpos << it->second << "*x^" << noshowpos << it->first;
        }
    }
}

```

```

        string result = ss.str();
        if (result.size() == 0) return "0";
        if (result[0] == '+') return result.substr(1);

        return result;
    }

int main()
{
    setlocale(LC_ALL, "rus");
    string str1, str2 = "$", str3;
    Stack st;
    int a, flag;
    int array[14][8] = {
        // x 5 6 + * ( ) #
        { 1, 1, 1, 0, 0, 1, 0, 0 },           //S
        { 2, 2, 2, 0, 0, 2, 0, 0 },           //E
        { 0, 0, 0, 3, 0, 0, 10, 10 },         //Z
        { 4, 4, 4, 0, 0, 4, 0, 0 },           //T
        { 0, 0, 0, 11, 5, 0, 11, 11 },        //H
        { 6, 7, 8, 0, 0, 9, 0, 0 },           //F
        { 88, 0, 0, 0, 0, 0, 0, 0 },          //x
        { 0, 88, 0, 0, 0, 0, 0, 0 },          //5
        { 0, 0, 88, 0, 0, 0, 0, 0 },          //6
        { 0, 0, 0, 88, 0, 0, 0, 0 },          //+
        { 0, 0, 0, 0, 88, 0, 0, 0 },          //*
        { 0, 0, 0, 0, 0, 88, 0, 0 },          //(
        { 0, 0, 0, 0, 0, 0, 88, 0 },          //)
        { 0, 0, 0, 0, 0, 0, 0, 100 }         //#
    };

    cout << "Введите строку: ";
    cin >> str1;
    for (a = 0; a <= (str1.size() - 1); a++)
    {
        if ((str1[a] == 'x') || (str1[a] == '5') || (str1[a] == '6') || (str1[a] == '*') || (str1[a] ==
'+' ) || (str1[a] == '(') || (str1[a] == ')'))
        {
            flag = 0;
        }
        else
        {
            flag = 1;
        }
    }
    if (flag == 0)
    {
        cout << "right" << endl;
        str3 = str1 + str2;
        LL1(str3, array, st);
        str3 = Ans(str1);
        cout << "\n" << "Результат = " << str3 << endl;
    }
}

```

```
    }  
    else  
    {  
        cout << "false" << endl;  
    }  
    system("PAUSE");  
    return 0;  
}
```