Министерство Образования Республики Беларусь

Учреждение образования

«Гомельский государственный технический университет

имени П.О.Сухого»

Кафедра «Информатика»

**Лабораторная работа № 4**

по дисциплине: **«Методы трансляции»**

**Разработка КС-грамматик**

Выполнил студент

группы ИП-32

*Суховенко Э.С.*

Проверил доцент:

*Кравченко О.А.*

Гомель 2022

***Цель работы***: *получить навыки создания КС-грамматик для заданного языка, разработка алгоритма и программы формирования множеств левых и правых символов для нетерминальных символов грамматики.*

***Задание***

Лабораторная работа предполагает выполнение двух этапов.

На первом этапе требуется разработать КС-грамматику, порождающую заданный язык.

На втором этапе необходимо разработать программу, которая для каждого нетерминального символа определяет множество его левых или правых символов:

L(*U*)={*W*|Ǝ(*U*→*Wx*) ˅ Ǝ(*U*→*Zx*) & *W*∈L(*Z*)} – множество левых символов нетерминального символа *U* состоит из символов, которые начинают цепочки, выводимые из *U*.

R(*U*)={*W*|Ǝ(*U*→*xW*) ˅ Ǝ(*U*→*xZ*) & *W*∈R(*Z*)} – множество правых символов нетерминального символа *U* состоит из символов, которые заканчивают цепочки, выводимые из *U*.

Вариант 12

Язык представляет собой множество арифметических выражений со скобочной структурой, операнды которого – идентификаторы неограниченной длины, состоящие из латинских строчных букв, а операции – вычитание и умножение.

Пример предложения языка: ((*fd*-*jh*)^*vv*-*s*^*sa*-*kiu*^(*mm*-*cxz*))^*r*

Для каждого нетерминального символа разработанной КС-грамматики программа должна построить множество его левых символов.

Семантика нетерминальных символов:  
S – начальный нетерминальный символ;

А – арифметическое выражение;

T – алгебраическое слагаемое (терм) в арифметическом выражении, между термами знак “–”;

U – множитель в терме, между множителями должен стоять знак “^”;

H – первичное арифметическое выражение в множителе, H представляется в виде : выражение в скобках;

V – идентификатор – последовательность строчных латинских букв и цифр, начинающаяся со строчной буквы;

L – любая строчная латинская буква;

Разработанная КС-грамматика:

1. *S→A*

*2-3. A→T | A-T*

*4-5. T→U | T^U*

*6. U→H*

*7-8. H→V |(A)*

*9-10. V→L | VL*

1. *L→m*

Проверка разработанной КС-грамматики:

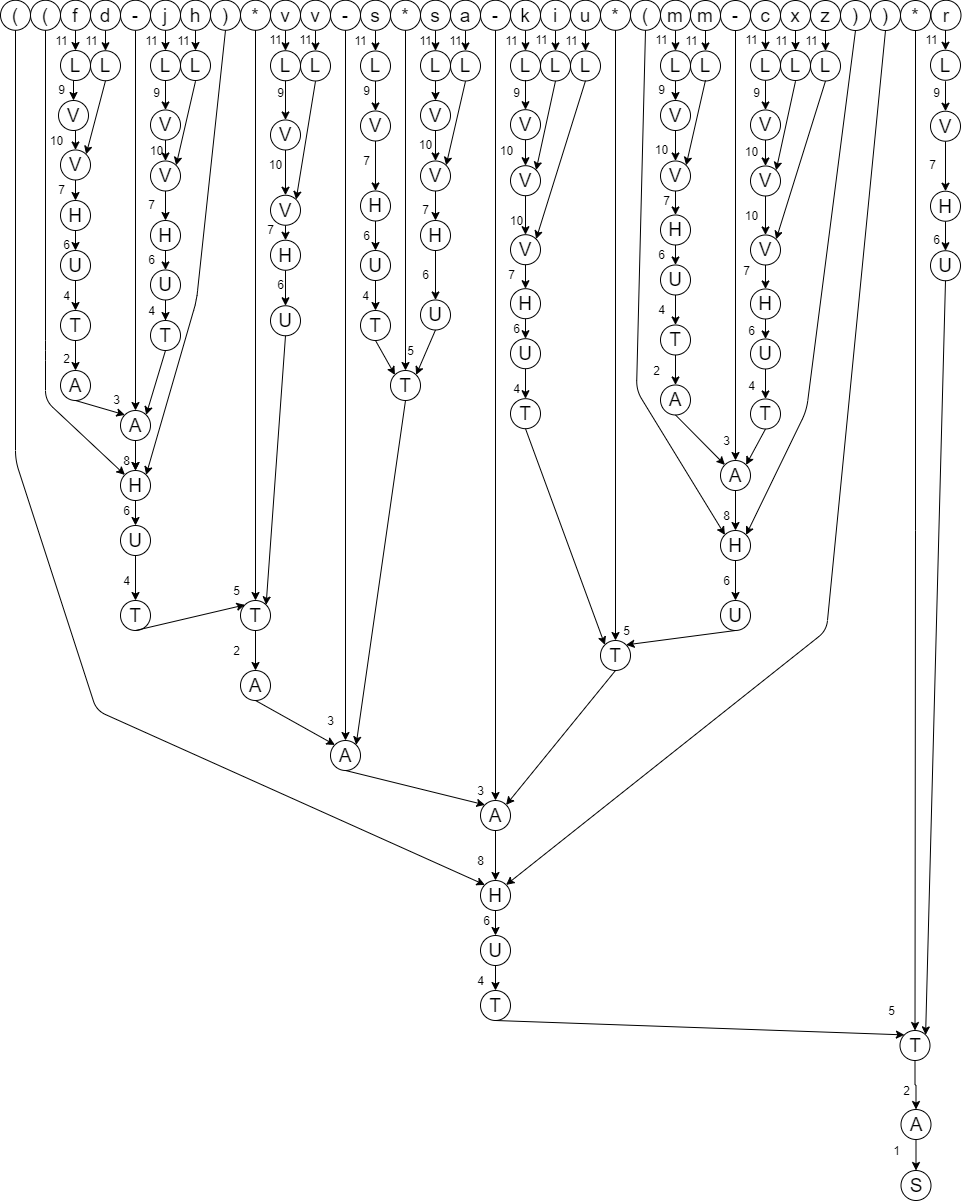


Рисунок 1 – Результат проверки разработанной КС-грамматики

Получение множества левых символов:

*L (L)={ m }*

*L (V)={ L, m , V }*

*L (H)={ V, L, m, ( }*

*L (U)={ H, V, L, m, ( }*

*L(T)={ U, H, V, L, m, ( , T }*

*L(A)={ T, U, H, V, L, m, ( , A}*

*L(S)={ A, T, U, H, V, L, m, ( }*

Получение множества правых символов:

*R(L)={ m }*

*R (V)={ L, m }*

*R (H)={ V, L, m, ) }*

*R (U)={ H, V, L, m, ) }*

*R (T)={ U, H, V, L, m, ( }*

*R (A)={ T, U, H, V, L, m, ( }*

*R (S)={ A, T, U, H, V, L, m, ( }*

Результат работы программы:



Рисунок 2- Результат работы программы по получению множества левых символов

Код программы:

using System;

using System.Collections.Generic;

using System.Linq;

namespace laba4

{

public class Program

{

private static bool \_isRightOrLeft = false;

private static void Main(string[] args)

{

var list = new List<NodeList>

{

new NodeList("S", "A"),

new NodeList("A", "A-T"),

new NodeList("A", "T"),

new NodeList("T", "T\*U"),

new NodeList("T", "U"),

new NodeList("U", "H"),

new NodeList("H", "(A)"),

new NodeList("H", "V"),

new NodeList("V", "VL"),

new NodeList("V", "L"),

new NodeList("L", "m"),

};

list.Reverse();

var rules = list.GroupBy(rule => rule.Key).ToDictionary(rule => rule.Key, rule => rule.ToList());

foreach (var key in rules)

{

var result = new List<char>();

AppendDataInNodeList(key.Key, rules, result);

var flag = \_isRightOrLeft ? "R" : "L";

Console.WriteLine($"{flag}({key.Key}) -> {{{string.Join(", ", result.Distinct())}}}");

}

Console.ReadLine();

}

private static void AppendDataInNodeList(string key, IReadOnlyDictionary<string, List<NodeList>> rules, ICollection<char> result)

{

if (!rules.ContainsKey(key)) return;

var values = rules[key];

foreach (var value in values)

{

var newKey = \_isRightOrLeft ? value.Value.Last() : value.Value.First();

if (newKey.ToString().Equals(key))

{

result.Add(newKey);

continue;

}

result.Add(newKey);

AppendDataInNodeList(newKey.ToString(), rules, result);

}

}

}

public class NodeList

{

public NodeList(string key, string value)

{

Key = key;

Value = value;

}

public string Key { get; }

public string Value { get; }

}

}

**Вывод:** получили навыки создания КС-грамматик для заданного языка, проверили разработанную КС-грамматику – она была создана правильно, без ошибок, разработали программу формирования множеств левых символов для нетерминальных символов грамматики.