МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ

Учреждение образования  
«Гомельский государственный технический университет имени П.О. Сухого»

Факультет автоматизированных и информационных систем

Кафедра «Информационные технологии»

Специальность 1-40 04 01 «Информатика и технологии программирования»

Отчет  
 по лабораторной работе №4  
по дисциплине «Методы трансляции»  
на тему: «Разработка КС-грамматик»

Составил: студент гр. ИП-31 Коваленко А.И.

(подпись, дата)

Принял доцент кафедры ИТ Кравченко О.А.

(подпись, дата)

Гомель 2023

**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*.

Вариант 15

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*

((*fd*-*jh*)^*vv*-*s*^*sa*-*kiu*^(*mm*-*cxz*))^*r*

Проверка разработанной КС-грамматики:

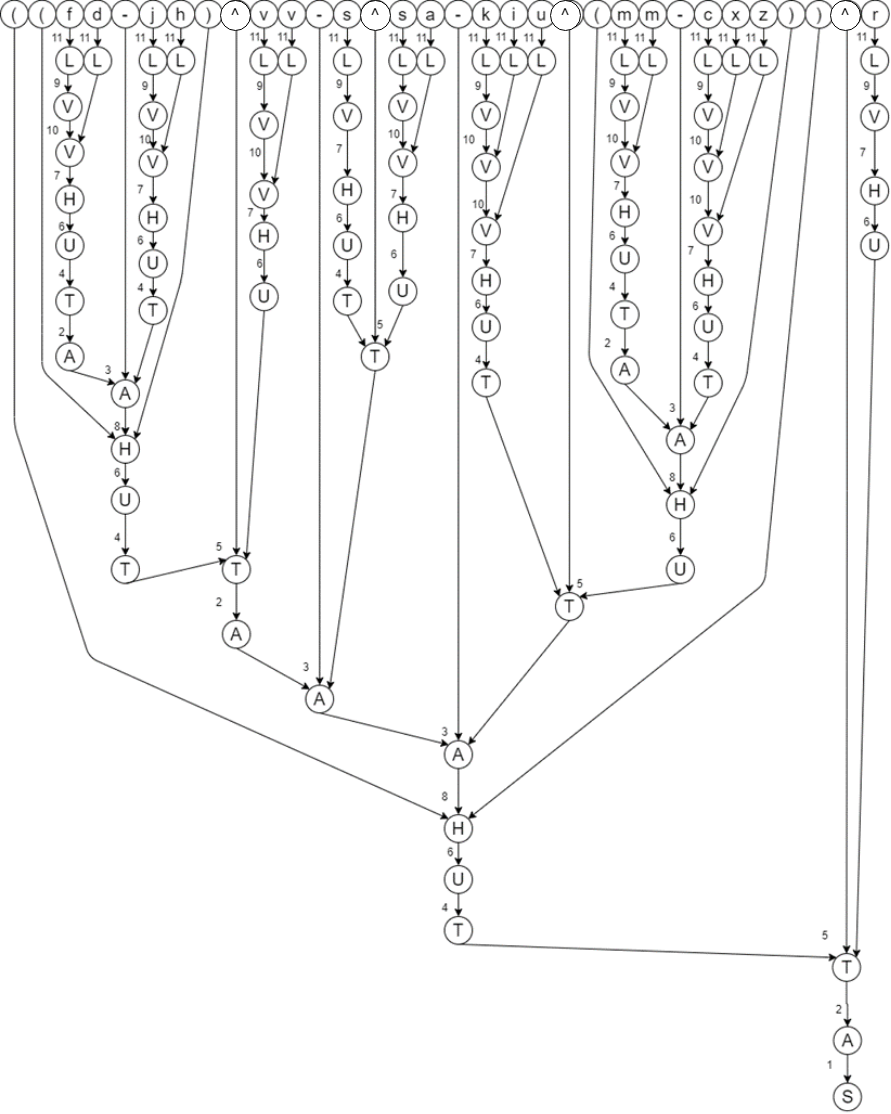


Рисунок 1 – Результат проверки разработанной КС-грамматики

Получение множества левых символов:

*L(L) -> [m]*

*L(V) -> [V, L, m]*

*L(H) -> [(, V, L, m]*

*L(U) -> [H, (, V, L, m]*

*L(T) -> [T, U, H, (, V, L, m]*

*L(A) -> [A, T, U, H, (, V, L, m]*

*L(S) -> [A, T, U, H, (, V, L, m]*

Результат работы программы:

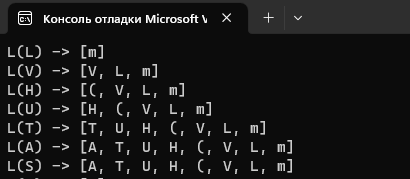


Рисунок 2 – Результат работы программы по получению множества левых символов

**Код программы:**

using System;

using System.Collections.Generic;

using System.Linq;

namespace fourth

{

public static class Program

{

private static IEnumerable<char> CollectSideChars

(

string key,

IReadOnlyDictionary<string, List<string>> rules,

Func<string, char> selector

)

{

if (!rules.ContainsKey(key))

return new HashSet<char>();

var result = new HashSet<char>();

var selectedUniqueChars = rules[key].Select(selector).Distinct();

foreach (var newKey in selectedUniqueChars)

{

result.Add(newKey);

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

result.UnionWith(CollectSideChars(newKey.ToString(), rules, selector));

}

return result;

}

private static void Main()

{

var list = new List<KeyValuePair<string, string>>

{

new ("S", "A"),

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

new ("A", "T"),

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

new ("T", "U"),

new ("U", "H"),

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

new ("H", "V"),

new ("V", "VL"),

new ("V", "L"),

new ("L", "m"),

};

var rules = list

.GroupBy(rule => rule.Key)

.ToDictionary(rule => rule.Key, rule => rule.Select(r => r.Value).ToList());

var selectors = new List<KeyValuePair<char, Func<string, char>>>

{

new('L', s => s.First()),

};

selectors.ForEach(selector =>

{

rules.Keys

.Reverse()

.ToList()

.ForEach(key =>

Console.WriteLine($"{selector.Key}({key}) -> [{string.Join(", ", CollectSideChars(key, rules, selector.Value))}]"));

});

}

}

}

**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.