НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО

Факультет Программной Инженерии и Компьютерной Техники

Формальные языки и грамматики

Лабораторная работа № 2

Выполнила студентка:

Борисова Анастасия Денисовна

Группа № P33222

Преподаватель: Лаздин Артур Вячеславович

г. Санкт-Петербург

2023

Исходная грамматика

S → aAS | cfAc | aS | cfc

A → bA | b

Матрица для отношения =

0;0;0;0;0;0;

1;0;0;1;0;0;

1;1;0;0;0;0;

0;0;0;0;1;0;

0;1;0;1;0;0;

0;1;0;0;0;0;

Матрица для отношения <

0;0;0;0;0;0;

0;0;1;1;0;0;

0;0;1;1;0;1;

0;0;0;0;0;0;

0;0;0;0;0;1;

0;0;0;0;0;1;

Матрица для отношения >

0;0;0;0;0;0;

1;0;1;1;0;0;

0;0;0;0;0;0;

0;0;0;0;0;0;

0;0;0;0;0;0;

1;0;1;1;0;0;

ГРАММАТИКА НЕ ЯВЛЯЕТСЯ ГРАММАТИКОЙ ПРОСТОГО ПРЕДШЕСТВОВАНИЯ!

Конфликтные отношения:

<A> = <S> <A> > <S>

<A> < "a" <A> > "a"

<A> = "c" <A> < "c" <A> > "c"

Матрица предшествования:

|  | S | B | D | A | C | c | f | a | b |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| S |  |  |  |  |  |  |  |  |  |
| B | = | < |  |  |  | < |  | < |  |
| D |  |  |  |  |  | = |  |  |  |
| A | > | > |  |  |  | > |  | > |  |
| C | > | > |  |  |  | > |  | > |  |
| c |  |  |  |  |  |  | = |  |  |
| f |  |  | = | < | < | = |  |  | < |
| a | = | < |  | = |  | < |  | < | < |
| b | > | > |  | < | = | > |  | > | < |

S → BS

S → cfDc

S → aS

S → cfc

A → bC

A → b

B → aA

C → A

D → C

package org.example;

import java.util.\*;

public class Lab2 {

public static int[][] matrix = {{0, 0, 0, 0, 0, 0, 0, 0, 0, 2},

{3, 1, 0, 0, 0, 1, 0, 2, 0, 2},

{0, 0, 0, 0, 0, 3, 0, 0, 0, 2},

{2, 2, 0, 0, 0, 2, 0, 2, 0, 2},

{2, 2, 0, 0, 0, 2, 0, 2, 0, 2},

{0, 0, 0, 0, 0, 0, 3, 0, 0, 2},

{0, 0, 3, 2, 2, 3, 0, 0, 1, 2},

{3, 1, 0, 3, 0, 1, 0, 1, 1, 2},

{2, 2, 0, 1, 3, 2, 0, 2, 1, 2},

{1, 1, 1, 1, 1, 1, 1, 1, 1, 0}};

public static Map<String, Integer> grammar\_ind = Map.of("S", 0,

"B", 1,

"D", 2,

"A", 3,

"C", 4,

"c", 5,

"f", 6,

"a", 7,

"b", 8,

"#", 9);

public static Map<String, String> grammar = Map.of("BS", "S", "cfDc", "S", "aS", "S", "cfc", "S", "bC", "A", "b", "A","aA", "B","A", "C", "C", "D");

public static void main(String[] args) {

Scanner sc = new Scanner(System.in);

System.out.println("Input your expression: ");

String phrase = sc.next();

phrase += "#";

String[] sphrase = phrase.split("");

Stack<String> st = new Stack<>();

st.push("#");

Iterator<String> it = Arrays.stream(sphrase).iterator();

String sym = it.next();

while (true) {

if (sym.equals("#")) {

String check = "";

Stack<String> tm = (Stack<String>) st.clone();

while (!tm.isEmpty()) {

check = tm.pop() + check;

}

if (check.equals("#S")) {

System.out.println("Expression is correct!");

break;

}

}

int rel = matrix[grammar\_ind.get(st.peek())][grammar\_ind.get(sym)];

if (rel == 3 || rel == 1) {

st.push(sym);

sym = it.next();

} else if (rel == 2) {

String a = st.pop();

String rule = "";

rule = a + rule;

int rel\_1 = matrix[grammar\_ind.get(st.peek())][grammar\_ind.get(a)];

while (rel\_1 != 1) {

a = st.pop();

rule = a + rule;

rel\_1 = matrix[grammar\_ind.get(st.peek())][grammar\_ind.get(a)];

}

String nonT = grammar.get(rule);

if (nonT == null) {

System.out.println("Wrong expression!");

break;

}

st.push(nonT);

} else {

System.out.println("Wrong expression!");

break;

}

}

}

}