# 1. Мета роботи

- закрепить понятия «алфавит», «цепочка», «формальная грамматика» и «формальный язык», «выводимость цепочек», «эквивалентная грамматика»;

- сформировать умения и навыки распознавания типов формальных языков и грамматик по классификации Хомского, построения эквивалентных грамматик.

# 2. Завдання на роботу

При выполнении лабораторной работы следует реализовать следующие действия:

1) составить грамматику, порождающую формальный язык, заданный в соответствии с вариантом;

2) определить тип формальной грамматики и языка по классификации Хомского;

3) разработать программное обеспечение, распознающее принадлежность введенного слова составленной грамматике.

# 3. Індивідуальний варіант завдання

16) L(G) = {А, ААБ, АААББВ, ААААБББВВ,...}

# 4. Виконання роботи

## 4.1 Скласти граматику

G = {T, N, P, S};

T = {А, Б, В};

N = {X, Y};

P:

1. S → АSX | A;
2. AX → АБ;
3. БX → ББY;
4. БY → БВ;
5. ВX → XY;
6. ВY → ВВ;

## 4.2 Визначити тип по Хомському

Контекстно-залежна граматика (тип 1).

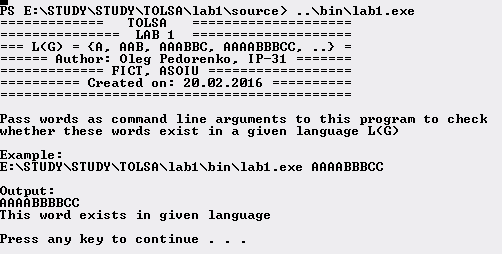
## 4.3 Приклад виведення слова за граматикою G

|  |  |  |
| --- | --- | --- |
| *№* | *Етап виведення* | *Правило* |
|  | S | Початковий символ граматики G |
|  | АSX | (1) |
|  | ААSXX | (1) |
|  | АААSXXX | (1) |
|  | ААА*АX*XX | (1) |
|  | ААА*АБ*XX | (2) |
|  | ААААБ*БY*X | (3) |
|  | ААААБ*БВ*X | (4) |
|  | ААААБ**БX**Y | (5) |
|  | ААААБ**ББY**Y | (3) |
|  | ААААБББ*ВY* | (4) |
|  | ААААБББ*ВВ* | (6) |
|  | ААААБББВВ | Кінцеве слово мови |

## 4.4 Розробка ПЗ для перевірки належності слова до складеної граматики.

Код розробленого ПЗ знаходиться у додатку до звіту. У програмі літери А, Б, В мови L були замінені на англійські літери A, B, C відповідно для полегшення роботи з введенням/виведенням даних.

Приклад роботи програми:



# 5. Висновок

У ході виконання даної лабораторної роботи були розглянуті та закріплені основні поняття і визначення, що стосуються формальних мов та граматик. Виконання завдання показало, що визначення типів мов та побудова коректної граматики – не така тяжка задача, як здається на перший погляд.

# Додаток. Лістинг програми

Допоміжний код (виведення довідки на екран) у лістингу опущено.

#include "iostream"

#include "string"

#include "cstdlib"

**using** **namespace** std**;**

/\*\*

matchGrammar

Parameters:

string s: a string that contains a word or sentence to be checked

Result:

bool: false if word s is not a legal word in the grammar G, true

otherwise.

\*/

bool matchGrammar**(**string s**)**

**{**

int n**[]** **=** **{**0**,** 0**,** 0**};**

char a**[]** **=** **{**'A'**,** 'B'**,** 'C'**};**

int i **=** 0**;**

**for(**int j **=** 0**;** j **<** 3**;** j**++)**

**{**

**for(**i**;** s**[**i**]** **==** a**[**j**]** **&&** i **<** s**.**length**();** i**++,** n**[**j**]++);**

**}**

**return** **((**n**[**0**]** **==** **(**n**[**1**]** **+** 1**))** **&&** **(**n**[**1**]** **==** **(**n**[**2**]** **+** 1**)))**

**||** **(**s **==** "A"**);**

**}**

int main**(**int argc**,** char**\*\*** argv**)**

**{**

**for(**int i **=** 1**;** i **<** argc**;** i**++)**

**{**

string expression **=** string**(**argv**[**i**]);**

cout **<<** expression **<<** endl**;**

**if(**matchGrammar**(**expression**))**

**{**

cout **<<** "This word exists in given language" **<<** endl**;**

**}**

**else**

**{**

cout **<<** "This word does not exist in given language" **<<** endl**;**

**}**

**}**

system**(**"pause"**);**

**return** 0**;**

**}**