МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ МОЛДОВЫ

ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ МОЛДОВЫ

ФАКУЛЬТЕТ ВЫЧИСЛИТЕЛЬНОЙ ТЕХНИКИ, ИНФОРМАТИКИ И МИКРОЭЛЕКТРОНИКИ

КАФЕДРА ВЫЧИСЛИТЕЛЬНОЙ ТЕХНИКИ

**ОТЧЕТ**

По лабораторной работе №1

по дисциплине «Формальные языки и программирование компиляторов»

Тема: «Регулярная грамматика»

Выполнил: студент группы TI – 155 Буянов Евгений

Проверила: преподаватель Скороходова Т.А.

Кишинев 2016

Задание

***Variant 5.***

G= (VN, VT, P, S);

VN={S, C, D}; VT= {a, b};

P= {

S → aD

D → bS

D → bC

D → b

C → aC

C → a

}

1. Для заданной грамматики G=(VN, VT, P, S) построить 4 цепочки, принадлежащие порождаемому этой грамматикой языку L(G). Длина цепочки должна быть не меньше, чем количество символов в алфавите VN плюс 2.
2. а) S→aD →abS→abaD→ababC→ababaC→ababaa;

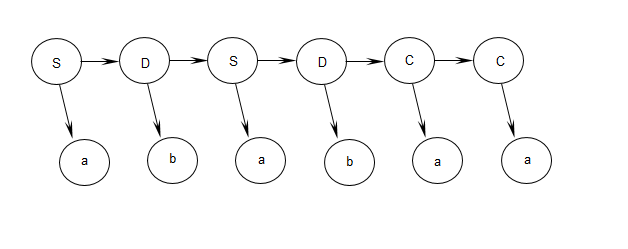
б) S→aD→abC→abaC→abaaC→abaaaC→abaaaa;

в)S→aD→abS→abaD→ababS→ababaD→ababab;

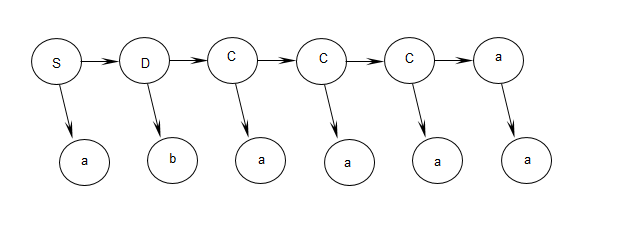
г)S→aD→abS→abaD→ababS→ababaD→abababS→abababaD→abababab;

1. Для каждой из этих цепочек построить дерево вывода.

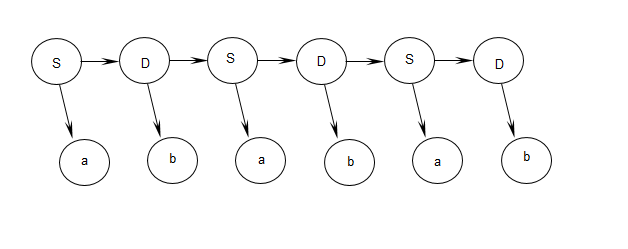
a)



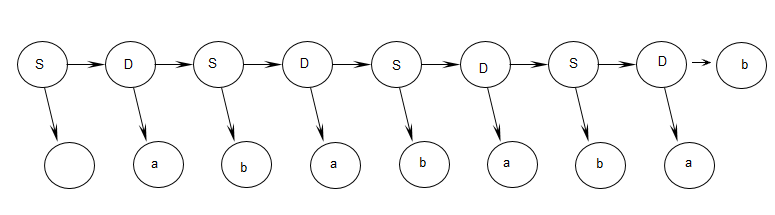
b)



c)



d)



1. К какому классу грамматик по Хомскому относится данная грамматика и почему?

По Хомскому данная грамматика относится к языку **типа 3(регулярный)**. Т.к. данная грамматика имеет вид:A→bC;

A→b.

1. Построить конечный автомат эквивалентный данной грамматике.

KA=(Q, ∑, δ, q0, F);

1) Q= VN ᴜ {x}={S, C, D, x};

2) Ʃ= VT ={a, b};

3) q0=S;

4) δ=ᴓ;

5)F={x}.

1. S → aD => δ(S,a)= δ(S,a) ᴜ {D}= ᴓ ᴜ {D}={D};
2. D → bS => δ(D,b)= δ(D,b) ᴜ {S}= ᴓ ᴜ {S}={S};
3. D → bC => δ(D,b)= δ(D,b) ᴜ {C}= ᴓ ᴜ {C}={C};
4. D → b => δ(D,b)= δ(D,b) ᴜ {x}= ᴓ ᴜ {x}={x};
5. C → aC => δ(C,a)= δ(C,a) ᴜ {C}= ᴓ ᴜ {C}={C};
6. C → a => δ(C,a)= δ(C,a) ᴜ {x}= ᴓ ᴜ {x}={x};
7. Построенный конечный автомат является детерминированным или недетерминированным и почему?

Построенный конечный автомат является недетерминированным, потому что из состояния q2(D) по b мы можем попасть в три различные состояния S, C, x.

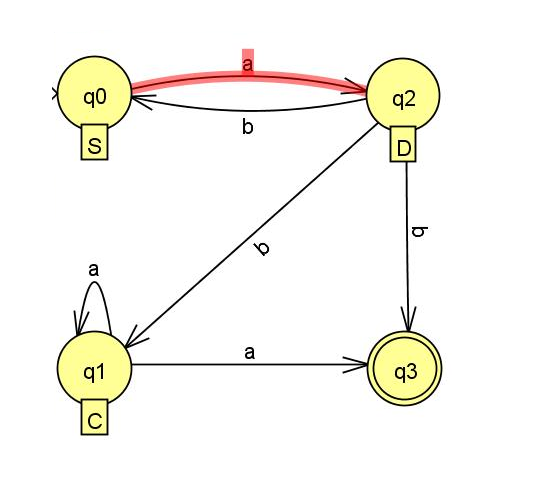
1. Построить ориентированный граф для полученного конечного автомата. 

Рисунок 1-Конечный автомат

1. Написать общий вид цепочек, принадлежащих порождаемому этой грамматикой языку L(G).

**L(G)={(ab)i, a\* |i=1,2,…};**

1. Написать регулярное выражение **e**, эквивалентное заданной грамматике **G** и показать, что L(e) = L(G).

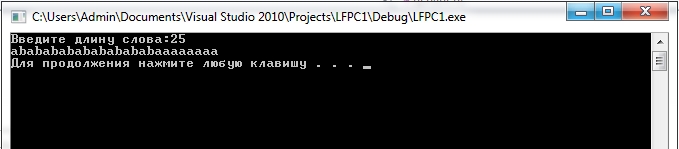
Выражение, эквивалентное заданной грамматике **G**:

**e= (ab)\*a\*;**

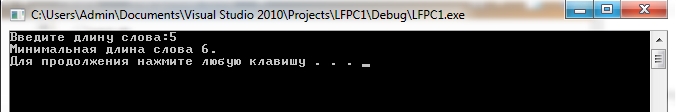
**L(e) =L((ab)ia\*)=L(ab) i ∙ (L(a)\* = (L(ab)) i ∙ ((L(a))\* =**

**= (L(a) ∙ L(b)) i ∙ ((L(a))\*=(a∙b) i ∙a\*, где i=1,2…**

1. Программа генерирующая слова принадлежащие данному языку.
2. Описание: данная программа генерирует слово принадлежащее данному языку заданной длины (минимальная длина слова 6 символов). Пользователь вводит длину слова после чего программа выводит на экран одно слово заданной длины.



Если слово меньше 6-ти символов программа выдаёт ошибку:



Код программы:

#include<iostream>

#include<conio.h>

#include<string.h>

#include<stdlib.h>

#include<vector>

using namespace std;

vector<char>word;

int num;

void OutVector(vector<char> vec){

for(vector<char>::const\_iterator i = vec.begin(); i != vec.end(); ++i)

cout << \*i ;

cout<<endl;

}

void gen(int n){

int r=rand()%n+1;

for(int i=0;i<r;i+=2){

word.push\_back('a');

word.push\_back('b');

}

for(int i=r;i<n;i++){

word.push\_back('a');

}

}

int main(){

setlocale (LC\_ALL, "RUS");

while(1){

word.clear();

cout<<"Введите длину слова:";

cin>>num;

if(num<6){

cout<<"Минимальная длина слова 6."<<endl;

goto end;

}

gen(num);

if(word.empty())

cout<<"Слово пустое."<<endl;

OutVector(word);

end:

system("pause");

system("cls");

}

return 0;

}

**Вывод:**

В ходе проделанной лабораторной работы мной были применены на практике знания об преобразовании регулярной грамматики в эквивалентный конечный автомат. Помимо этого была изучена программа JFLAP – графическим редактором, способным строить конечные автоматы по заданным правилам, а так же упрощать регулярную грамматику и выводить результат. Данная программа значительно облегчила выполнение лабораторной работы.