https://github.com/danielgabor99/FLCD/tree/master/Lab4

Automata
Vector<string>(states)
Vector<string>(alphabet)
String (initialState)
Vector<string>(finalStates)
Vector<Transition>(transitions)
+ void readFa()
+ bool isDeterministic()
+void checkSequence()

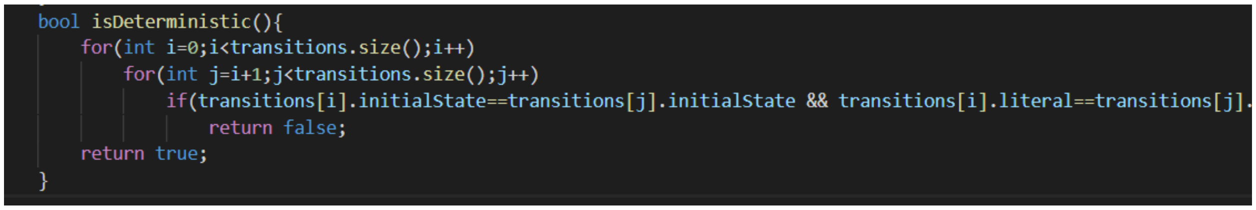
My Automata consists in a list of states, alphabets, final states and transitions;

bool isDeterministic():

Checks if it is deterministic of not, if yes, will return true, false otherwise.

Pre: the fa.

Post: true or false depending if it is deterministic or not.



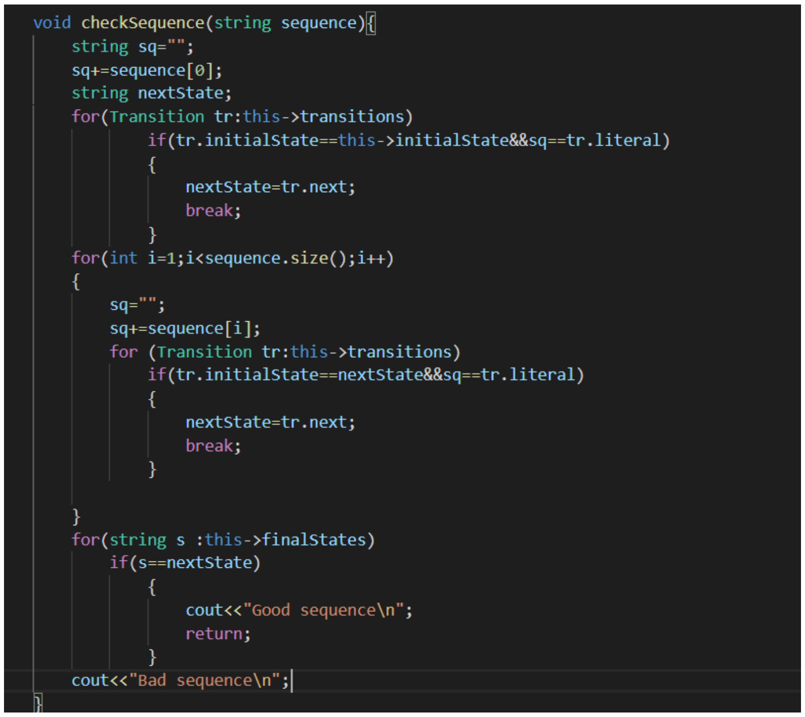
```
bool isDeterministic(){
    for(int i=0;i<transitions.size();i++)
        for(int j=i+1;j<transitions.size();j++)
            if(transitions[i].initialState==transitions[j].initialState && transitions[i].literal==transitions[j].
                return false;
    return true;
}
```

void checkSequence(sequence):

Checks if the sequence is valid

Pre: the sequence

Post: the validity of the sequence



```
void checkSequence(string sequence){
    string sq="";
    sq+=sequence[0];
    string nextState;
    for(Transition tr:this->transitions)
            if(tr.initialState==this->initialState&&sq==tr.literal)
            {
                nextState=tr.next;
                break;
            }
    for(int i=1;i<sequence.size();i++)
    {
        sq="";
        sq+=sequence[i];
        for (Transition tr:this->transitions)
            if(tr.initialState==nextState&&sq==tr.literal)
            {
                nextState=tr.next;
                break;
            }

    }
    for(string s :this->finalStates)
        if(s==nextState)
            {
                cout<<"Good sequence\n";
                return;
            }
    cout<<"Bad sequence\n";
}
```

q0,q1,q2,q3,q4

a,b,c

q0

q3

q0,a,q1

q1,b,q2

q2,c,q3

q3,a,q4