**ДОДАТОК А** Лістинг програми

package sample;

import api.VisFx;

import graph.VisEdge;

import graph.VisGraph;

import graph.VisNode;

import javafx.stage.Stage;

import java.util.ArrayList;

import java.util.HashMap;

import java.util.List;

public class Main {

public static void main(String[] args) {

Stage primaryStage = new Stage();

VisGraph graph = new VisGraph();

VisNode[] objects = new VisNode[25];

for (int i = 0; i < GenerateSM.stateMachine.getQLength(); i++) {

objects[i] = (new VisNode(i, GenerateSM.stateMachine.getQ(i)));

graph.addNodes(new VisNode(i, GenerateSM.stateMachine.getQ(i)));

}

List<TransitionFunction> list = new ArrayList<TransitionFunction>();

for (int i = 0; i < GenerateSM.stateMachine.getTFLength(); i++) {

list.add(GenerateSM.stateMachine.getTF(i));

}

String[][] strings = new String[GenerateSM.stateMachine.getTFLength()][3];

int count = 0;

String value = "";

int index = 0;

for (int i = 0; i < list.size(); i++) {

String s1 = list.get(i).VN;

String s2 = list.get(i).Result;

for (int j = i; j < list.size(); j++) {

if (s1.equals(list.get(j).VN) && s2.equals(list.get(j).Result)) {

count++;

value += list.get(i).VT + ",";

list.remove(j);

j--;

System.out.println(i);

}

}

if (count > 0) {

strings[index][0] = s1;

strings[index][1] = s2;

strings[index][2] = value.substring(0, value.length() - 1);

System.out.println(s1 + s2 + value);

value = "";

index++;

i--;

count = 0;

}

}

try {

for (int i = 0; i < strings.length; i++) {

try {

for (int j = 0; j < objects.length; j++) {

try {

if (strings[i][0].equals(objects[j].getLabel()))

for (int q = 0; q < objects.length; q++) {

System.out.println(strings[i][0].equals(objects[j].getLabel()) + " " + strings[i][1].equals(objects[q].getLabel()));

if (strings[i][0].equals(objects[j].getLabel()) && strings[i][1].equals(objects[q].getLabel())) {

//graph.addNodes()

graph.addEdges(new VisEdge(objects[j], objects[q], "to", strings[i][2]));

}

}

} catch (Exception e) {

}

}

} catch (Exception e) {

}

}

} catch (Exception e) {

}

VisFx.graphNetwork(graph, primaryStage);

primaryStage.show();

}

}

package sample;

import java.util.ArrayList;

import java.util.List;

public class AutomationGramar

{

private List<String> VT=new ArrayList<String>();

private List<String> VN=new ArrayList<String>();

private List<String> P=new ArrayList<String>();

private String S;

public List<String> Steps=new ArrayList<String>();

public String getVN(int index)

{

return VN.get(index);

}

public String getVT(int index)

{

return VT.get(index);

}

public String getP(int index)

{

return P.get(index);

}

public String getS()

{

return S;

}

public String getSteps(int i) {

return Steps.get(i);

}

public void setS(String s)

{

S = s;

}

public void setVT(String value)

{

VT.add(value);

}

public void setVN(String value)

{

VN.add(value);

}

public void setP(String value)

{

P.add(value);

}

public int getIndex(String string) {return P.indexOf(string);}

public void RemoveP(String value)

{

P.remove(value);

}

public int getPLength()

{

return P.size();

}

public int getStepLength()

{

return Steps.size();

}

public int getVNLength()

{

return VN.size();

}

public int getVTLength()

{

return VT.size();

}

}

package sample;

public class Controller

{

ObservableList<String> list= FXCollections.observableArrayList();

public static RegGramar regGramar ;

@FXML

private ResourceBundle resources;

@FXML

private URL location;

@FXML

private TextField input\_p\_txt;

@FXML

private Button Convertbtn;

@FXML

private TextField VTtxt;

@FXML

private Button AddBtn;

@FXML

private Button Prbtn;

@FXML

private ListView<String> Plist;

@FXML

private TextField VNtxt;

@FXML

private TextField Stxt;

@FXML

void initialize()

{

regGramar = new RegGramar();

Prbtn.setOnMouseClicked(event ->

{

ObservableList<String> lists= FXCollections.observableArrayList();

VNtxt.setText("C,S,F,M,T,D,R");

VTtxt.setText("0,1,2,3,4,5,6,7,8,9,g,s");

Stxt.setText("C");

lists.add("C=D2|g"); lists.add("S=Cs|F0|F1|F2|F3|F4|F5|S0|S1|S2|S3|S4|S5|S6|S7|S8|M0|M1|M2|M3|M4|M5|M6|M7|M8|D0|D1|T0|T1|T2|T3|T4|T5|T6|T7|T8");

lists.add("F=S9");

lists.add("R=M9");

lists.add("T=R6");

lists.add("M=F6|R0|R1|R2|R3|R4|R5");

lists.add("D=T9");

Plist.setItems(lists);

list=lists;

});

input\_p\_txt.setOnKeyReleased

(

event ->

{ if((VNtxt.getText().indexOf(Character.toString(input\_p\_txt.getText().charAt(input\_p\_txt.getText().length()-1)))>=0)

|| VTtxt.getText().indexOf(Character.toString(input\_p\_txt.getText().charAt(input\_p\_txt.getText().length()-1)))>=0

|| (Character.toString(input\_p\_txt.getText().charAt(input\_p\_txt.getText().length()-1)).equals("|")&&!Character.toString(input\_p\_txt.getText().charAt(input\_p\_txt.getText().length()-2)).equals("|")&&

!Character.toString(input\_p\_txt.getText().charAt(input\_p\_txt.getText().length()-2)).equals("="))

||(( Character.toString(input\_p\_txt.getText().charAt(input\_p\_txt.getText().length()-1)).equals("=") &&input\_p\_txt.getText().length()==2)))

{

if(input\_p\_txt.getText().length()==1

&&!(VNtxt.getText().indexOf(Character.toString(input\_p\_txt.getText().charAt(input\_p\_txt.getText().length()-1)))>=0))

input\_p\_txt.clear();

if(input\_p\_txt.getText().length()>3)

{

if ((VNtxt.getText().indexOf(Character.toString(input\_p\_txt.getText().charAt(input\_p\_txt.getText().length() - 1))) >= 0)

&& !Character.toString(input\_p\_txt.getText().charAt(input\_p\_txt.getText().length() - 2)).equals("|")) {

input\_p\_txt.setText(input\_p\_txt.getText().substring(0, input\_p\_txt.getText().length() - 1));

input\_p\_txt.selectPositionCaret(input\_p\_txt.getText().length());

}

}

}

else

{ input\_p\_txt.setText(input\_p\_txt.getText().substring(0,input\_p\_txt.getText().length()-1));

input\_p\_txt.selectPositionCaret(input\_p\_txt.getText().length()+1);

}

}

);

Stxt.setOnKeyReleased

(

event->

{ if((VNtxt.getText().indexOf(Character.toString(Stxt.getText().charAt(Stxt.getText().length()-1)))<0) || Stxt.getText().length()>1) Stxt.clear();

else

{

input\_p\_txt.setText(Stxt.getText()+"=");

} } );

VTtxt.setOnKeyReleased

( event -> { if(VNtxt.getText().indexOf(Character.toString(VTtxt.getText().charAt(VTtxt.getText().length()-1)))<0 ||(Character.toString(VTtxt.getText().charAt(VTtxt.getText().length()-1))).equals(",")

||(Character.toString(VTtxt.getText().charAt(VTtxt.getText().length()-1))).equals('"')) System.out.println("");/\*regGramar.setVT(Character.toString(VTtxt.getText().charAt(VTtxt.getText().length()-1)));\*/

else { VTtxt.setText(VTtxt.getText().substring(0,VTtxt.getText().length()-1));

VTtxt.selectPositionCaret(VTtxt.getText().length()+1); }});

VNtxt.setOnKeyReleased

( event -> { if(!(VTtxt.getText().indexOf(Character.toString(VNtxt.getText().charAt(VNtxt.getText().length()-1)))>=0) ||(Character.toString(VNtxt.getText().charAt(VNtxt.getText().length()-1))).equals(",")) System.out.println("");/\*regGramar.setVT(Character.toString(VTtxt.getText().charAt(VTtxt.getText().length()-1)));\*/ else { VNtxt.setText(VNtxt.getText().substring(0,VNtxt.getText().length()-1));

VNtxt.selectPositionCaret(VNtxt.getText().length()+1); } } );

AddBtn.setOnMouseClicked

( event ->

{ if(input\_p\_txt.getText().length()>2) {

list.add(input\_p\_txt.getText().trim() );

Plist.setItems(list);

input\_p\_txt.clear(); } else

{ Alert alert= new Alert(Alert.AlertType.ERROR);

alert.setTitle("Error");

alert.setHeaderText(null);

alert.setContentText("Ви не повністю написали правило!");

alert.showAndWait(); } } );

Convertbtn.setOnMouseClicked

( event -> { if (!(list.size() < 1)) {

if (Stxt.getText().length() < 1 || VNtxt.getText().length() < 1 || VTtxt.getText().length() < 1)

{ Alert alert = new Alert(Alert.AlertType.ERROR);

alert.setTitle("Error");

alert.setHeaderText(null);

alert.setContentText("Будь ласка заповніть всі поля!");

alert.showAndWait();

} else {

regGramar.setS(Stxt.getText());

VT\_Analysator();

VN\_Analysator();

try {

P\_Analysator(list);

} catch (Exception e) {

Alert alert = new Alert(Alert.AlertType.ERROR);

alert.setTitle("Error");

alert.setHeaderText(null);

alert.setContentText(e.getMessage());

alert.showAndWait();

list.clear();

Plist.setItems(list);

VNtxt.clear();

VTtxt.clear();

Stxt.clear();

return; }

regGramar.setS(Stxt.getText())

try {

//System.out.println("tytss");

FXMLLoader fxmlLoader = new FXMLLoader(getClass().getResource("sampleSM.fxml"));

Parent root = (Parent) fxmlLoader.load();

Stage stage = new Stage();

stage.setScene(new Scene(root));

stage.setOnCloseRequest(event1 ->

System.exit(0)

);

stage.show();

Convertbtn.getScene().getWindow().hide();

ControllerSM controller = fxmlLoader.getController(); stage.setOnCloseRequest(controller.getCloseEventHandler());

} catch (IOException e) {

e.printStackTrace() } } }

else

{ Alert alert = new Alert(Alert.AlertType.ERROR);

alert.setTitle("Error");

alert.setHeaderText(null);

alert.setContentText("Додайте хоча б одне правило");

alert.showAndWait();

} } ); } public void VT\_Analysator()

{ for (int i=0;i<VTtxt.getText().length();i++)

{ if(!Character.toString(VTtxt.getText().charAt(i)).equals(","))

regGramar.setVT(Character.toString(VTtxt.getText().charAt(i))); } }

public void VN\_Analysator()

{ for (int i=0;i<VNtxt.getText().length();i++)

{ if(!Character.toString(VNtxt.getText().charAt(i)).equals(","))

regGramar.setVN(Character.toString(VNtxt.getText().charAt(i))); } }

public void P\_Analysator(ObservableList<String> list) throws Exception

{ for(String symbol:list) {

String Pravilo;

Pravilo = symbol; try {

if (regGramar.IsNotTerminal(Pravilo.substring(0, Pravilo.indexOf("=")))) { } else {

throw new Exception("Символ не є нетермінальним або неправильно задана ліва частина правила: " + Pravilo.substring(0, Pravilo.indexOf("=")));

} } catch (StringIndexOutOfBoundsException e) {

throw new Exception("Граматика повинна мати вигляд S=Pa|Aa"); }

String P1 = "";

for (int i = 2; i < Pravilo.length(); i++)

{ if (regGramar.IsNotTerminal(Character.toString(Pravilo.charAt(i))))

{ if (i > 0 && !Character.toString(Pravilo.charAt(i - 1)).equals("|") && !Character.toString(Pravilo.charAt(i - 1)).equals("="))

throw new Exception("Це не ліволінійна регулярна граматика, правило виду: " + Pravilo.substring(0) + " не є правилом ліволінійної граматики, продивіться довідку!");

else P1 += Character.toString(Pravilo.charAt(i)); } else

if (regGramar.IsTerminal(Character.toString(Pravilo.charAt(i))))

P1 += Character.toString(Pravilo.charAt(i));

else

if (!Character.toString(Pravilo.charAt(i)).equals("|"))

throw new Exception("Невідомий символ " + Pravilo.charAt(i));

else

{ regGramar.setP(Pravilo.charAt(0) + "=" + P1);

P1 = ""; } } regGramar.setP(Pravilo.charAt(0) + "=" + P1); } }}

package sample;

public class ControllerRG

{ @FXML

private ResourceBundle resources;

@FXML

private URL location;

@FXML

private WebView web;

@FXML

void initialize() throws InterruptedException {

web.setPrefHeight(758);

web.setPrefWidth(1400);

WebEngine engine = web.getEngine();

engine.load(this.getClass().getResource("index.html").toExternalForm()); }}

package sample;

public class ControllerSM {

ObservableList<String> lists= FXCollections.observableArrayList();

@FXML

private ResourceBundle resources;

@FXML

private URL location;

@FXML

private TextField VTtxt;

@FXML

private ListView<String> Plist;

@FXML

private TextField VNtxt;

@FXML

private TextField Stxt;

@FXML

private TextField Vtxt;

@FXML

private Button AGbtn;

@FXML

private Button Gbtn

@FXML

private ListView<String> blist;

@FXML

private TextField Ftxt1;

@FXML

private TextField Qtxt;

@FXML

private Button Stepbtn;

@FXML

private TextField q0txt;

@FXML

void initialize()

{ObservableList<String> list= FXCollections.observableArrayList();

RegularInAutomation regularInAutomation = new RegularInAutomation();

regularInAutomation.Step\_One();

regularInAutomation.Step\_Two();

regularInAutomation.Step\_Three();

GenerateSM generateSM= new GenerateSM();

generateSM.Step\_One();

generateSM.Step\_Two();

generateSM.Step\_Three();

generateSM.Step\_Four();

generateSM.Step\_Five();

StateMachine st=GenerateSM.stateMachine;

//----------------------------------------------------------------------

Stxt.setText(automationGramar.getS());

VNtxt.setText(automationGramar.getVN(0));

for (int i=1;i<automationGramar.getVNLength();i++)

VNtxt.setText(VNtxt.getText()+", "+automationGramar.getVN(i));

VTtxt.setText(automationGramar.getVT(0));

for (int i=1;i<automationGramar.getVTLength();i++)

VTtxt.setText(VTtxt.getText()+", "+automationGramar.getVT(i));

for (int i=0;i<automationGramar.getPLength();i++)

lists.add (automationGramar.getP(i));

Plist.setItems(lists);

//--------------------------------------------------------------------------------------

q0txt.setText(st.getQ0());

Qtxt.setText(st.getQ(0));

for (int i=1;i<st.getQLength();i++)

Qtxt.setText(Qtxt.getText()+", "+st.getQ(i));

Vtxt.setText(st.getV(0));

for (int i=1;i<st.getVLength();i++)

Vtxt.setText(Vtxt.getText()+", "+st.getV(i));

for (int i=0;i<st.getTFLength();i++)

list.add (st.getTransactionFunction(i));

blist.setItems(list);

Ftxt1.setText(st.getF(0));

for (int i=1;i<st.getFLength();i++)

Ftxt1.setText(Ftxt1.getText()+", "+st.getF(i));

Gbtn.setOnMouseClicked

( event -> Main.main(new String[]{}) );

Stepbtn.setOnMouseClicked((MouseEvent event) ->

{XMLLoader fxmlLoader = new FXMLLoader(getClass().getResource("sampleSteps.fxml"));

Parent root = null;

try { root = fxmlLoader.load();

} catch (IOException e) {

e.printStackTrace(); }

Stage stage = new Stage();

stage.setScene(new Scene(root));

stage.setTitle("Павлов П.О. Программа перетворення регулярної граматики годинника в скінченний автомат.");

stage.show(); });

AGbtn.setOnMouseClicked((MouseEvent event) ->

{ FXMLLoader fxmlLoader = new FXMLLoader(getClass().getResource("sampleRG.fxml"));

Parent root = null;

try { root = fxmlLoader.load();

} catch (IOException e) {

e.printStackTrace(); }

Stage stage = new Stage();

stage.setMaxWidth(600);

stage.setMaxHeight(620);

stage.setScene(new Scene(root));

stage.setTitle("Павлов П.О. Программа перетворення регулярної граматики годинника в скінченний автомат.");

stage.show(); }); }

private javafx.event.EventHandler<WindowEvent> closeEventHandler = new javafx.event.EventHandler<WindowEvent>() {

@Override

public void handle(WindowEvent event) {

FXMLLoader fxmlLoader = new FXMLLoader(getClass().getResource("sample.fxml"));

Parent root = null;

try { root = fxmlLoader.load();

} catch (IOException e) {

e.printStackTrace(); }

Stage stage = new Stage();

stage.setScene(new Scene(root));

stage.show(); } };

public javafx.event.EventHandler<WindowEvent> getCloseEventHandler(){

return closeEventHandler; }}

package sample;

public class ControllerSteps {

@FXML

private ResourceBundle resources;

@FXML

private URL location;

@FXML

private ListView<String> Plist;

@FXML

void initialize() throws IOException {

ObservableList<String> list= FXCollections.observableArrayList();

for(int i=0;i<RegularInAutomation.automationGramar.getStepLength();i++)

list.add(RegularInAutomation.automationGramar.getSteps(i));

Plist.setItems(list); } }

package sample;

public class GenerateSM

{ public static StateMachine stateMachine=new StateMachine();

AutomationGramar automationGramar= RegularInAutomation.automationGramar;

public void Step\_One()

{ stateMachine = new StateMachine();

automationGramar.Steps.add("ПЕРЕТВОРЕННЯ АВТОМАТНОЇ ГРАМАТИКИ В СКІНЧЕННИЙ АВТОМАТ:");

automationGramar.Steps.add("КРОК 1:");

stateMachine.setQ("H");

for(int i=0;i<automationGramar.getVNLength();i++)

{ stateMachine.setQ(automationGramar.getVN(i)); } }

public void Step\_Two()

{ for(int i=0;i<automationGramar.getVTLength();i++)

{ stateMachine.setV(automationGramar.getVT(i)); } }

public void Step\_Three()

{ automationGramar.Steps.add("КРОК 3:");

for(int i=0;i<automationGramar.getPLength();i++)

{ String value=automationGramar.getP(i);

if(value.length()>3)

{ automationGramar.Steps.add("Додавання правила переходу "+"( "+Character.toString(value.charAt(2))+","+Character.toString(value.charAt(3))+")="+Character.toString(value.charAt(0))); stateMachine.setTransactionFunction(Character.toString(value.charAt(2)),Character.toString(value.charAt(3)),Character.toString(value.charAt(0))); }

if(value.length()<4&& value.length()>2&& !regGramar.IsNotTerminal(Character.toString(value.charAt(2))))

{ automationGramar.Steps.add("Додавання правила переходу "+"( "+"H"+","+Character.toString(value.charAt(2))+")="+Character.toString(value.charAt(0))); stateMachine.setTransactionFunction("H",Character.toString(value.charAt(2)),Character.toString(value.charAt(0))); } }

for (int j=0;j<stateMachine.getTFLength();j++) System.out.println(stateMachine.getTransactionFunction(j)); }

public void Step\_Four()

{ stateMachine.setF(automationGramar.getS()); }

public void Step\_Five()

{ stateMachine.setQ0("H"); }}

package sample;

public class Main {

public static void main(String[] args) {

Stage primaryStage = new Stage();

VisGraph graph = new VisGraph();

VisNode[] objects = new VisNode[25];

for (int i = 0; i < GenerateSM.stateMachine.getQLength(); i++) {

objects[i] = (new VisNode(i, GenerateSM.stateMachine.getQ(i)));

graph.addNodes(new VisNode(i, GenerateSM.stateMachine.getQ(i))); }

List<TransitionFunction> list = new ArrayList<TransitionFunction>();

for (int i = 0; i < GenerateSM.stateMachine.getTFLength(); i++) {

list.add(GenerateSM.stateMachine.getTF(i)); }

String[][] strings = new String[GenerateSM.stateMachine.getTFLength()][3];

int count = 0;

String value = "";

int index = 0;

for (int i = 0; i < list.size(); i++) {

String s1 = list.get(i).VN;

String s2 = list.get(i).Result;

for (int j = i; j < list.size(); j++) {

if (s1.equals(list.get(j).VN) && s2.equals(list.get(j).Result)) {

count++;

value += list.get(i).VT + ",";

list.remove(j);

j--;

System.out.println(i); } }

if (count > 0) {

strings[index][0] = s1;

strings[index][1] = s2;

strings[index][2] = value.substring(0, value.length() - 1);

System.out.println(s1 + s2 + value);

value = "";

index++;

i--;

count = 0; } }

try {

for (int i = 0; i < strings.length; i++) {

try {

for (int j = 0; j < objects.length; j++) {

try {

if (strings[i][0].equals(objects[j].getLabel()))

for (int q = 0; q < objects.length; q++) {

System.out.println(strings[i][0].equals(objects[j].getLabel()) + " " + strings[i][1].equals(objects[q].getLabel()));

if (strings[i][0].equals(objects[j].getLabel()) && strings[i][1].equals(objects[q].getLabel())) {

//graph.addNodes()

graph.addEdges(new VisEdge(objects[j], objects[q], "to", strings[i][2])); } }

} catch (Exception e) { } }

} catch (Exception e) { } }

} catch (Exception e) { }

VisFx.graphNetwork(graph, primaryStage);

primaryStage.show(); }}

package sample;

public class MainSM extends Application {

@Override

public void start(Stage primaryStage) throws Exception{

Parent root = FXMLLoader.load(getClass().getResource("sample.fxml"));

primaryStage.setTitle("Павлов П.О. Программа перетворення регулярної граматики годинника в скінченний автомат.");

primaryStage.setScene(new Scene(root));

//primaryStage.setMaximized(true);

primaryStage.show(); }

public static void main(String[] args) {

launch(args); }}

package sample;

public class RegGramar{

private List<String> VT=new ArrayList<String>();

private List<String> VN=new ArrayList<String>();

private List<String> P=new ArrayList<String>();

private String S;

public String getVN(int index)

{ return VN.get(index); }

public String getVT(int index)

{ return VT.get(index); }

public String getP(int index)

{ return P.get(index); }

public String getS()

{ return S; }

public void setS(String s)

{ S = s; }

public void setVT(String value)

{ VT.add(value); }

public void setVN(String value)

{ VN.add(value); }

public void setP(String value)

{ P.add(value); }

public int getPLength()

{ return P.size(); }

public int getVNLength()

{ return VN.size(); }

public int getVTLength()

{ return VT.size(); }

public boolean IsTerminal(String value) {

for (String symbol:VT)

if(symbol.equals(value))

return true;

return false; }

public boolean IsNotTerminal(String value) {

for (String symbol:VN)

if(symbol.equals(value))

return true;

return false; }}

package sample;

import static sample.Controller.regGramar;

public class RegularInAutomation

{ String[] S=new String[]{"S","Q","W","E","R","T","Y","U","I","O","P","L","K","J","H","G","F","D","A","M","N","B","C","X","Z" ,"S0","Q0","W0","E0","R0","T0","Y0","U0","I0","O0","P0","L0","K0","J0","H0","G0","F0","D0","A0","M0","N0","B0","C0","X0","Z0"};

int index=1;

List<String> list;

static AutomationGramar automationGramar= new AutomationGramar();

public void Step\_One()

{ automationGramar=new AutomationGramar();

automationGramar.Steps.add("ПЕРЕТВОРЕННЯ РЕГУЛЯРНОЇ ГРАМАТИКИ В АВТОМАТНУ:");

for(int i = 0; i< regGramar.getVNLength(); i++)

{ automationGramar.setVN(regGramar.getVN(i)); }

for(int i = 0; i< regGramar.getVTLength(); i++)

{ automationGramar.setVT(regGramar.getVT(i)); }

automationGramar.setS(regGramar.getS());

} public void Step\_Two()

{ automationGramar.Steps.add("КРОК 2:");

for(int i=0;i<regGramar.getPLength();i++) {

list= new ArrayList<String>();

String value=regGramar.getP(i).substring(2);

System.out.println(value);

String first=String.valueOf(regGramar.getP(i).charAt(0));

final int length=value.length(); if(!(value.length()==0)&&regGramar.IsTerminal(Character.toString(value.charAt(0)))) { if(value.length()>1)

{ value = firstTerminal(value);

index++;

if(value.length()>2)

{ for (int x = 0; x < length-2; x++)

{ value = firstNotTerminal(value);

index++; }

automationGramar.Steps.add("Перенесення правила "+first+"="+value+" до множини правил автоматної граматики");

automationGramar.setP(first + "=" + value);

for(String s:list)

{ automationGramar.Steps.add("Перенесення правила "+s+" до множини правил автоматної граматики");

automationGramar.setP(s); } } else

{ automationGramar.Steps.add("Перенесення правила "+first+"="+value+" до множини правил автоматної граматики");

automationGramar.setP(first+"="+value); } } else

{ automationGramar.Steps.add("Перенесення правила "+first+"="+value+" до множини правил автоматної граматики");

automationGramar.setP(first+"="+value); } }

else if(!(value.length()==0)&&regGramar.IsNotTerminal(Character.toString(value.charAt(0))))

{ if(value.length()>2)

{ for (int z = 0; z < length-2; z++)

{ value = firstNotTerminal(value);

index++; }

automationGramar.Steps.add("Перенесення правила "+first+"="+value+" до множини правил автоматної граматики");

automationGramar.setP(first + "=" + value);

for(String string:list)

{ automationGramar.setP(string);

automationGramar.Steps.add("Перенесення правила "+string+" до множини правил автоматної граматики"); } }

else

{ automationGramar.setP(first+"="+value);

automationGramar.Steps.add("Перенесення правила "+first+"="+value+" до множини правил автоматної граматики"); } }

else

{ automationGramar.Steps.add("Перенесення правила "+first+"="+value+" до множини правил автоматної граматики");

automationGramar.setP(first+"="+value); } } }

public String firstTerminal(String value)

{ if(!regGramar.IsNotTerminal(S[index])&&!regGramar.IsTerminal(S[index]))

{ list.add(S[index]+"="+value.substring(0,1));

automationGramar.setVN(S[index]);

automationGramar.Steps.add("Заміна правила "+ value+" на правило "+S[index]+value.substring(1));

return S[index]+value.substring(1); }

else

{ index++;

return firstTerminal(value); } }

public String firstNotTerminal(String value)

{ if(!regGramar.IsNotTerminal(S[index])&&!regGramar.IsTerminal(S[index]))

{ list.add(S[index]+"="+value.substring(0,S[index-1].length()+1));

automationGramar.setVN(S[index]);

automationGramar.Steps.add("Заміна правила "+ value+" на правило "+S[index]+value.substring(S[index-1].length()+1));

return S[index]+value.substring(S[index-1].length()+1); }

else

{ index++;

return firstNotTerminal(value); } }

public void Step\_Three()

{ Three\_Step();

/\*for(String s:automationGramar.Steps)System.out.println(s);

automationGramar.setS(regGramar.getS());

for (int i=0;i<automationGramar.getPLength();i++) System.out.println(automationGramar.getP(i));\*/ }

public boolean Three\_Step()

{ automationGramar.Steps.add("КРОК 3:");

boolean value=false;

int i=0;

while(i<automationGramar.getPLength())

{ System.out.println("tyt"); if(automationGramar.getP(i).length()<4&&automationGramar.getP(i).length()>2)

{ String rule= automationGramar.getP(i);

String symbol=Character.toString(rule.charAt(2)); if(regGramar.IsNotTerminal(Character.toString(automationGramar.getP(i).charAt(2)))&&!Chracter.toString(rule.charAt(0)).equals(symbol))

{ System.out.println("tyt");

value=true;

for(int j=0;j<automationGramar.getPLength();j++)

{ System.out.println("tyt"); if(Character.toString(automationGramar.getP(j).charAt(0)).equals(symbol))

{ String s=automationGramar.getP(j); automationGramar.setP(rule.substring(0,2)+automationGramar.getP(j).substring(2));

automationGramar.Steps.add("Заміна правила "+ rule +" на правило "+(rule.substring(0,2)+automationGramar.getP(j).substring(2)));

automationGramar.Steps.add("Видалення правила "+rule);

automationGramar.RemoveP(rule);

//automationGramar.Steps.add("Видалення правила "+s);

//automationGramar.RemoveP(s);

i=-1;

j=automationGramar.getIndex(s); } } } }

else

if(automationGramar.getP(i).length()==2)

{ System.out.println("tyt");

value=true;

String rule= automationGramar.getP(i);

String symbol=Character.toString(rule.charAt(0)); if(!Character.toString(automationGramar.getP(i).charAt(0)).equals(automationGramar.getS()))

for(int j=0;j<automationGramar.getPLength();j++) { if(!automationGramar.getP(j).equals(rule)&&Character.toString(automationGramar.getP(j).charAt(2)).equals(symbol) )

{ System.out.println("tyt");

String s=automationGramar.getP(j);

automationGramar.setP(s.substring(0,2)+s.charAt(s.length()-1));

automationGramar.Steps.add("Заміна правила "+ s +" на правило "+s.substring(0,2)+s.charAt(s.length()-1));

automationGramar.Steps.add("Видалення правила "+rule);

automationGramar.RemoveP(rule);

//automationGramar.Steps.add("Видалення правила "+s);

//automationGramar.RemoveP(s);

i=-1;

j=automationGramar.getIndex(s); } } }

i++; }

return value; }}

package sample;

public class StateMachine

{ private List<String> Q=new ArrayList<String>();

private List<String> V=new ArrayList<String>();

private List<TransitionFunction> transition\_functions=new ArrayList<TransitionFunction>();

private String q0="";

private List<String> F=new ArrayList<String>();

public String getQ(int index)

{ return Q.get(index); }

public String getV(int index)

{ return V.get(index); }

public String getTransactionFunction(int index)

{ return transition\_functions.get(index).toString(); }

public TransitionFunction getTF(int index){return transition\_functions.get(index);}

public String getF(int index)

{ return F.get(index); }

public String getQ0()

{ return q0; }

public void setQ0(String s)

{ q0 = s; }

public void setQ(String value)

{ Q.add(value); }

public void setV(String value)

{ V.add(value); }

public void setF(String value)

{ F.add(value); }

public void setTransactionFunction(String VN,String VT, String Result) {

transition\_functions.add(new TransitionFunction(VN,VT,Result)); }

public int getQLength()

{ return Q.size(); }

public int getFLength()

{ return F.size(); }

public int getVLength()

{ return V.size(); }

public int getTFLength()

{ return transition\_functions.size(); }}

package sample;

public class TransitionFunction

{ String VN="";

String VT="";

String Result;

TransitionFunction(String VN,String VT,String result) {

this.VN=VN;

this.VT=VT;

this.Result=result; }

@Override

public String toString() {

return ("b("+VN+","+VT+")="+Result); }}

Sample.fxml

<?xml version="1.0" encoding="UTF-8"?>

<Pane maxHeight="-Infinity" maxWidth="-Infinity" minHeight="-Infinity" minWidth="-Infinity" prefHeight="626.0" prefWidth="1054.0" style="-fx-background-color: #e7e2e2;" xmlns="http://javafx.com/javafx/8" xmlns:fx="http://javafx.com/fxml/1" fx:controller="sample.Controller">

<children>

<TextField fx:id="VTtxt" layoutX="226.0" layoutY="32.0" prefHeight="37.0" prefWidth="146.0">

<font>

<Font name="Book Antiqua" size="16.0" />

</font></TextField>

<TextField fx:id="VNtxt" layoutX="444.0" layoutY="32.0" prefHeight="37.0" prefWidth="178.0">

<font>

<Font name="Book Antiqua" size="16.0" />

</font></TextField>

<Label layoutX="173.0" layoutY="32.0" text="G({">

<font>

<Font name="Berlin Sans FB Bold" size="27.0" />

</font>

</Label>

<Label layoutX="399.0" layoutY="32.0" text="},{">

<font>

<Font name="Berlin Sans FB Bold" size="27.0" />

</font>

</Label>

<Label layoutX="756.0" layoutY="32.0" text=")">

<font>

<Font name="Berlin Sans FB Bold" size="27.0" />

</font>

</Label>

<TextField fx:id="Stxt" layoutX="695.0" layoutY="32.0" prefHeight="37.0" prefWidth="44.0">

<font>

<Font name="Book Antiqua" size="16.0" />

</font></TextField>

<Label layoutX="633.0" layoutY="32.0" text="}, P,">

<font>

<Font name="Berlin Sans FB Bold" size="27.0" />

</font>

</Label>

<Label layoutX="77.0" layoutY="86.0" text="P:">

<font>

<Font name="Berlin Sans FB Bold" size="32.0" />

</font>

</Label>

<TextField fx:id="input\_p\_txt" layoutX="116.0" layoutY="95.0" prefHeight="25.0" prefWidth="353.0">

<font>

<Font name="Book Antiqua" size="16.0" />

</font></TextField>

<ListView fx:id="Plist" layoutX="101.0" layoutY="134.0" prefHeight="459.0" prefWidth="542.0" />

<Button fx:id="AddBtn" layoutX="487.0" layoutY="88.0" mnemonicParsing="false" style="-fx-background-radius: 100; -fx-background-color: #7cbcdd;" text="Додати правило">

<font>

<Font name="Berlin Sans FB Demi Bold" size="21.0" />

</font>

</Button>

<Button fx:id="Convertbtn" layoutX="803.0" layoutY="274.0" mnemonicParsing="false" style="-fx-background-radius: 100; -fx-background-color: #7cbcdd;" text="Перетворити до СА">

<font>

<Font name="Berlin Sans FB Demi Bold" size="21.0" />

</font>

</Button>

<Button fx:id="Prbtn" alignment="CENTER" layoutX="877.0" layoutY="364.0" mnemonicParsing="false" prefHeight="68.0" prefWidth="73.0" style="-fx-background-radius: 150; -fx-background-color: #7cbcdd;" text="Приклад">

<font>

<Font name="Times New Roman Bold" size="13.0" />

</font>

</Button>

</children>

</Pane>

sampleRG.fxml

<?xml version="1.0" encoding="UTF-8"?>

<Pane maxHeight="-Infinity" maxWidth="-Infinity" minHeight="-Infinity" minWidth="-Infinity" prefHeight="620.0" prefWidth="600.0" style="-fx-padding: 0; -fx-background-color: #abeab4;" styleClass="clock" stylesheets="@clock.css" xmlns="http://javafx.com/javafx/8" xmlns:fx="http://javafx.com/fxml/1" fx:controller="sample.ControllerRG">

<children>

<WebView fx:id="web" layoutY="-1.0" prefHeight="620.0" prefWidth="600.0" />

</children>

</Pane>

sampleSM.fxml

<?xml version="1.0" encoding="UTF-8"?>

<Pane centerShape="false" focusTraversable="true" maxHeight="-Infinity" maxWidth="-Infinity" minHeight="-Infinity" minWidth="-Infinity" nodeOrientation="LEFT\_TO\_RIGHT" prefHeight="626.0" prefWidth="1054.0" style="-fx-background-color: #e7e2e2;" xmlns="http://javafx.com/javafx/8" xmlns:fx="http://javafx.com/fxml/1" fx:controller="sample.ControllerSM">

<children>

<TextField fx:id="Qtxt" editable="false" layoutX="339.0" layoutY="14.0" prefHeight="37.0" prefWidth="158.0">

<font>

<Font name="Book Antiqua" size="16.0" />

</font></TextField>

<TextField fx:id="Vtxt" editable="false" layoutX="557.0" layoutY="14.0" prefHeight="37.0" prefWidth="178.0">

<font>

<Font name="Book Antiqua" size="16.0" />

</font></TextField>

<Label layoutX="286.0" layoutY="14.0" text="M({">

<font>

<Font name="Berlin Sans FB Bold" size="27.0" />

</font>

</Label>

<Label layoutX="512.0" layoutY="14.0" text="},{">

<font>

<Font name="Berlin Sans FB Bold" size="27.0" />

</font>

</Label>

<Label layoutX="989.0" layoutY="14.0" text=")">

<font>

<Font name="Berlin Sans FB Bold" size="27.0" />

</font>

</Label>

<TextField fx:id="q0txt" editable="false" layoutX="808.0" layoutY="14.0" prefHeight="37.0" prefWidth="44.0">

<font>

<Font name="Book Antiqua" size="16.0" />

</font></TextField>

<Label layoutX="746.0" layoutY="14.0" text="}, б,">

<font>

<Font name="Berlin Sans FB Bold" size="27.0" />

</font>

</Label>

<ListView fx:id="blist" layoutX="20.0" layoutY="153.0" prefHeight="459.0" prefWidth="316.0" />

<Label layoutX="852.0" layoutY="14.0" text=" ,">

<font>

<Font name="Berlin Sans FB Bold" size="27.0" />

</font>

</Label>

<TextField fx:id="Ftxt1" editable="false" layoutX="868.0" layoutY="14.0" prefHeight="37.0" prefWidth="118.0">

<font>

<Font name="Book Antiqua" size="16.0" />

</font>

</TextField>

<Button fx:id="Gbtn" layoutX="804.0" layoutY="219.0" mnemonicParsing="false" prefHeight="48.0" prefWidth="208.0" style="-fx-background-radius: 20; -fx-background-color: #7cbcdd;" text="Показати граф СА">

<font>

<Font name="Franklin Gothic Demi Cond" size="18.0" />

</font></Button>

<Button fx:id="Stepbtn" layoutX="798.0" layoutY="313.0" mnemonicParsing="false" prefHeight="48.0" prefWidth="219.0" style="-fx-background-radius: 20; -fx-background-color: #7cbcdd;" text="Кроки переведення до СА">

<font>

<Font name="Franklin Gothic Demi Cond" size="18.0" />

</font></Button>

<Button fx:id="AGbtn" layoutX="798.0" layoutY="407.0" mnemonicParsing="false" prefHeight="48.0" prefWidth="219.0" style="-fx-background-radius: 20; -fx-background-color: #7cbcdd;" text="Показати приклад">

<font>

<Font name="Franklin Gothic Demi Cond" size="18.0" />

</font></Button>

<ListView fx:id="Plist" layoutX="345.0" layoutY="153.0" prefHeight="459.0" prefWidth="351.0" />

<Label layoutX="735.0" layoutY="59.0" prefHeight="37.0" prefWidth="75.0" style="-fx-background-color: #93d7fb;" text="}, P,">

<font>

<Font name="Berlin Sans FB Bold" size="27.0" />

</font>

</Label>

<TextField fx:id="Stxt" editable="false" layoutX="808.0" layoutY="59.0" prefHeight="37.0" prefWidth="56.0" style="-fx-background-color: #93d7fb;">

<font>

<Font name="Book Antiqua" size="16.0" />

</font>

</TextField>

<Label layoutX="854.0" layoutY="59.0" prefHeight="37.0" prefWidth="27.0" style="-fx-background-color: #93d7fb; -fx-background-radius: 10;" text=")">

<font>

<Font name="Berlin Sans FB Bold" size="27.0" />

</font>

</Label>

<Label layoutX="483.0" layoutY="59.0" prefHeight="37.0" prefWidth="75.0" style="-fx-background-color: #93d7fb;" text="},{">

<font>

<Font name="Berlin Sans FB Bold" size="27.0" />

</font>

</Label>

<Label layoutX="281.0" layoutY="59.0" prefHeight="37.0" prefWidth="63.0" style="-fx-background-color: #93d7fb;" text=" G({">

<font>

<Font name="Berlin Sans FB Bold" size="27.0" />

</font>

</Label>

<TextField fx:id="VNtxt" editable="false" layoutX="557.0" layoutY="59.0" prefHeight="37.0" prefWidth="178.0" style="-fx-background-color: #93d7fb;">

<font>

<Font name="Book Antiqua" size="16.0" />

</font>

</TextField>

<TextField fx:id="VTtxt" editable="false" layoutX="339.0" layoutY="59.0" prefHeight="37.0" prefWidth="146.0" style="-fx-background-color: #93d7fb;">

<font>

<Font name="Book Antiqua" size="16.0" />

</font>

</TextField>

<Label layoutX="345.0" layoutY="109.0" style="-fx-background-color: #93d7fb; -fx-background-radius: 10;" text="P автом. грам.:">

<font>

<Font name="Times New Roman Bold" size="32.0" />

</font>

</Label>

<Label layoutX="48.0" layoutY="59.0" prefHeight="37.0" prefWidth="240.0" style="-fx-background-color: #93d7fb; -fx-background-radius: 10;" text="Автом. грам.:">

<font>

<Font name="Times New Roman Bold" size="32.0" />

</font>

</Label>

<Label layoutX="35.0" layoutY="109.0" text="Функції переходу б:">

<font>

<Font name="Times New Roman Bold" size="32.0" />

</font>

</Label>

<Label layoutX="49.0" layoutY="22.0" text="Скінч. автом.:">

<font>

<Font name="Times New Roman Bold" size="32.0" />

</font>

</Label>

</children>

</Pane>

sampleSteps.fxml

<?xml version="1.0" encoding="UTF-8"?>

<Pane maxHeight="-Infinity" maxWidth="-Infinity" minHeight="-Infinity" minWidth="-Infinity" prefHeight="626.0" prefWidth="860.0" style="-fx-background-color: #e7e2e2;" xmlns="http://javafx.com/javafx/8" xmlns:fx="http://javafx.com/fxml/1" fx:controller="sample.ControllerSteps">

<children>

<Label layoutX="50.0" layoutY="14.0" text="Кроки побудови скінченного автомату:">

<font>

<Font name="Berlin Sans FB Bold" size="32.0" />

</font>

</Label>

<ListView fx:id="Plist" layoutX="31.0" layoutY="71.0" prefHeight="537.0" prefWidth="781.0" />

</children>

</Pane>

Index.html

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<title>Павлов П.О. Программа перетворення регулярної граматики годинника в скінченний автомат.</title>

<link rel="stylesheet" href="stylesheets/style.css">

</head>

<body>

<div class="clock">

<div class="arrow-hour">h</div>

<div class="arrow-minute">m</div>

<div class="arrow-second">s</div>

</div>

</body>

</html>

Style.css

/\*Animation\*/

@keyframes seconds{

0%{

transform:rotate(0deg); }

100%{

transform:rotate(360deg); }}

/\*@-webkit-keyframes seconds{

0%{

transform:rotate(0deg); }

100%{

transform:rotate(450deg); }}\*/

@keyframes minutes{

0%{

transform:rotate(-1deg); }

100%{

transform:rotate(360deg); }}

/\*@-webkit-keyframes minutes{

0%{

transform:rotate(120deg); }

100%{

transform:rotate(480deg); }}\*/

@keyframes hours{

0%{

transform:rotate(1deg); }

100%{

transform:rotate(510deg); }}

/\*@-webkit-keyframes hours{

0%{

transform:rotate(150deg); }

100%{

transform:rotate(510deg); }}\*/

.clock{

float: left;

overflow: hidden;

width: 600px;

height: 600px;

position: relative;

top:0;

background: transparent url('../img/clockface.jpg') no-repeat 0 0;}

.arrow-second{

z-index: 1;

position: absolute;

width: 30px;

height: 600px;

text-indent: -9999px;

left: 50%;

margin-left: -15px;

top: 0;

background: transparent url('../img/sechand.png') no-repeat 0 0;

/\* -webkit-animation: seconds 60s steps(60, end) infinite;\*/

animation: seconds 60s steps(60, end) infinite;}

.arrow-minute{

z-index: 3;

position: absolute;

width: 20px;

height: 600px;

text-indent: -9999px;

left: 50%;

margin-left: -10px;

top: -7px;

background: transparent url('../img/minhand.png') no-repeat 0 0;

/\* -webkit-animation: minutes 3600s steps(3600, end) infinite;\*/

animation: minutes 3600s steps(3600, end) infinite;}

.arrow-hour{

z-index: 3;

position: absolute;

width: 25px;

height: 600px;

text-indent: -9999px;

left: 50%;

margin-left: -15px;

top: 0;

background: transparent url('../img/hourhand.png') no-repeat 0 0;

/\*-webkit-animation: hours 216000s steps(216000, end) infinite;\*/

animation: hours 216000s steps(216000, end) infinite;}