МИНОБРНАУКИ РОССИИ  
федеральное государственное бюджетное образовательное учреждение  
высшего образования

«ЧЕРЕПОВЕЦКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»

|  |  |
| --- | --- |
| Институт (факультет) | Институт информационных технологий |
| Кафедра | Кафедра математического и программного обеспечения ЭВМ |

ЛАБОРАТОРНАЯ РАБОТА № 1-2

Построение модели автомата Мили и Мура

Выполнил: Овчинников Максим Владимирович

студент группы: 1ПИб-02-2оп 22

Проверил: Ганичева Оксана Георгиевна

Череповец  
2023

Введение

**Часть 1**

Автомат выдачи напитков является механизмом, созданным для автоматизации процесса предоставления различных напитков. Рассмотрим его работу с точки зрения теории автоматов:

Начало: пользователь начинает взаимодействие вставляя монеты номиналом 5 (событие x5) или 10 (событие x10) рублей. Это переводит автомат в состояние s5 или s10 соответственно. После того, как автомат достигнет состояния S40, S45 или S50, можно будет выбрать напиток.

Выбор напитка: Пользователь может выбрать напиток кофе (событие Xcoff) или шоколад (событие Xchoc), переводя банкомат в состояния Sopr. После выбора напитка можно добавить или убавить сахар

Выбор сахара: после выбора напитка пользователь может выбрать добавить (событие Xplus) или убавить (событие Xminus) сахар.

Завершение операции: По окончании операции, банкомат выдает выбранный напиток и возвращает сдачу (если она есть), переводя автомат в состояние S0.

Завершение работы: Пользователь может завершить работу, если при S45, S50 будет переполнение (событие X10, X5).

Работа автомата выдачи напитков, рассмотренная с точки зрения теории автоматов, показывает, как устройство обрабатывает входные сигналы, перемещается между различными состояниями и проводит операции в соответствии с установленными правилами. Это типичный пример автомата с ограниченным набором состояний, применяемый в автоматах выдачи напитков.

Формальное описание модели

Модель автомата Мили называется шестерка компонентов следующего вида:

А = (X, Y, S, S0, δ, λ)

X – конечное не пустое множество входных сигналов (входной алфавит);

Y – конечное не пустое множество выходящих сигналов (выходной алфавит);

S – конечное не пустое множество состояний;

S0 – начальное состояние;

δ: S×X -> S – функция переходов;

λ: S×X -> Y – функция выходов.

S:

S0-Принята сумма 0 р

S5-Принята сумма 5 р

S10-Принята сумма 10 р

S15-Принята сумма 15 р

S20-Принята сумма 20 р

S25-Принята сумма 25 р

S30-Принята сумма 30 р

S35-Принята сумма 35 р

S40-Принята сумма 40 р

S45-Принята сумма 45 р

S50-Принята сумма 50 р

Sopr-Напиток определён, возможность выбора сахара

X:

X5-Опустить монету 5 р

X10-Опустить монету 10 р

Xcoff-Нажать кнопку "Коффе"

Xchoc-Нажать кнопку "Шоколад"

Xplus-Нажать кнопку "+сахар"

Xminus-Нажать кнопку "-сахар"

Xchange-Нажать кнопку "Выдача"

Y:

Y1-Выбран коффе

Y2-Выбран шоколад

Y3-Выдан напиток и сдача

Y5-Принято 5 р

Y10-Принято 10 р

Yplus-Сахар добавлен

Yminus-Сахар уменьшен

Ychange-Автомат переполнен, выдача сдачи

Таблицы и граф автомата Мили:

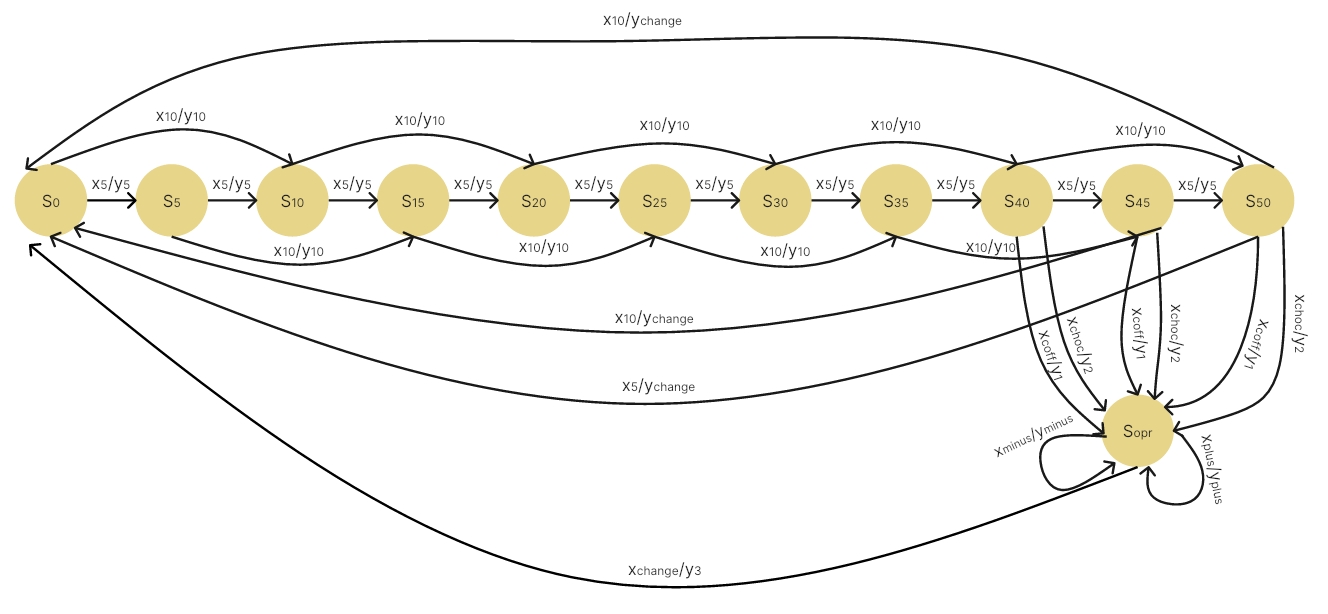


Рис. 1. Автомат Мили

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| X/S | S0 | S5 | S10 | S15 | S20 | S25 | S30 | S35 | S40 | S45 | S50 | Sopr |
| X5 | S5 | S10 | S15 | S20 | S25 | S30 | S35 | S40 | S45 | S50 | S0 | - |
| X10 | S10 | S15 | S20 | S25 | S30 | S35 | S40 | S45 | S50 | S0 | S0 | - |
| Xchoc | - | - | - | - | - | - | - | - | Sopr | Sopr | Sopr | - |
| Xcoff | - | - | - | - | - | - | - | - | Sopr | Sopr | Sopr | - |
| Xplus | - | - | - | - | - | - | - | - | - | - | - | Sopr |
| Xminus | - | - | - | - | - | - | - | - | - | - | - | Sopr |
| Xchange | - | - | - | - | - | - | - | - | - | - | - | S0 |

δ: S×X 🡪 S (функция переходов)

λ: S×X 🡪 Y (функция выходов)

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| X/S | S0 | S5 | S10 | S15 | S20 | S25 | S30 | S35 | S40 | S45 | S50 | Sopr |
| X5 | Y5 | Y5 | Y5 | Y5 | Y5 | Y5 | Y5 | Y5 | Y5 | Y5 | Ychange | - |
| X10 | Y10 | Y10 | Y10 | Y10 | Y10 | Y10 | Y10 | Y10 | Y10 | Ychange | Ychange | - |
| Xchoc | - | - | - | - | - | - | - | - | Y2 | Y2 | Y2 | - |
| Xcoff | - | - | - | - | - | - | - | - | Y1 | Y1 | Y1 | - |
| Xplus | - | - | - | - | - | - | - | - | - | - | - | Yplus |
| Xminus | - | - | - | - | - | - | - | - | - | - | - | Yminus |
| Xchange | - | - | - | - | - | - | - | - | - | - | - | Y3 |

δ, λ:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| X/S | S0 | S5 | S10 | S15 | S20 | S25 |
| X5 | S5/Y5 | S10/Y5 | S15/Y5 | S20/Y5 | S25/Y5 | S30/Y5 |
| X10 | S10/Y10 | S15/Y10 | S20/Y10 | S25/Y10 | S30/Y10 | S35/Y10 |
| Xchoc | - | - | - | - | - | - |
| Xcoff | - | - | - | - | - | - |
| Xplus | - | - | - | - | - | - |
| Xminus | - | - | - | - | - | - |
| Xchange | - | - | - | - | - | - |
| X/S | S30 | S35 | S40 | S45 | S50 | Sopr |
| X5 | S35/Y5 | S40/Y5 | S45/Y5 | S50/Y5 | S0/Ychange | - |
| X10 | S40/Y10 | S45/Y10 | S50/Y10 | S0/Ychange | S0/Ychange | - |
| Xchoc | - | - | Sopr/Y2 | Sopr/Y2 | Sopr/Y2 | - |
| Xcoff | - | - | Sopr/Y1 | Sopr/Y1 | Sopr/Y1 | - |
| Xplus | - | - | - | - | - | Sopr/Yplus |
| Xminus | - | - | - | - | - | Sopr/Yminus |
| Xchange | - | - | - | - | - | S0/Y3 |

Протокол работы автомата Мили

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| t | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 |
| X | X5 | X10 | X10 | Xchoc | X5 | X10 | X10 | X5 | Xcoff | Xminus | Xchange |  |
| S | S0 | S5 | S15 | S25 | S25 | S30 | S40 | S50 | S50 | Sopr | Sopr | S0 |
| Y | Y5 | Y10 | Y10 | - | Y5 | Y10 | Y10 | Ychange | Y1 | Yminus | Y3 |  |

Автомат Мура

Модель автомата Мура

А = (X, Y, S, S0, δ, λ)

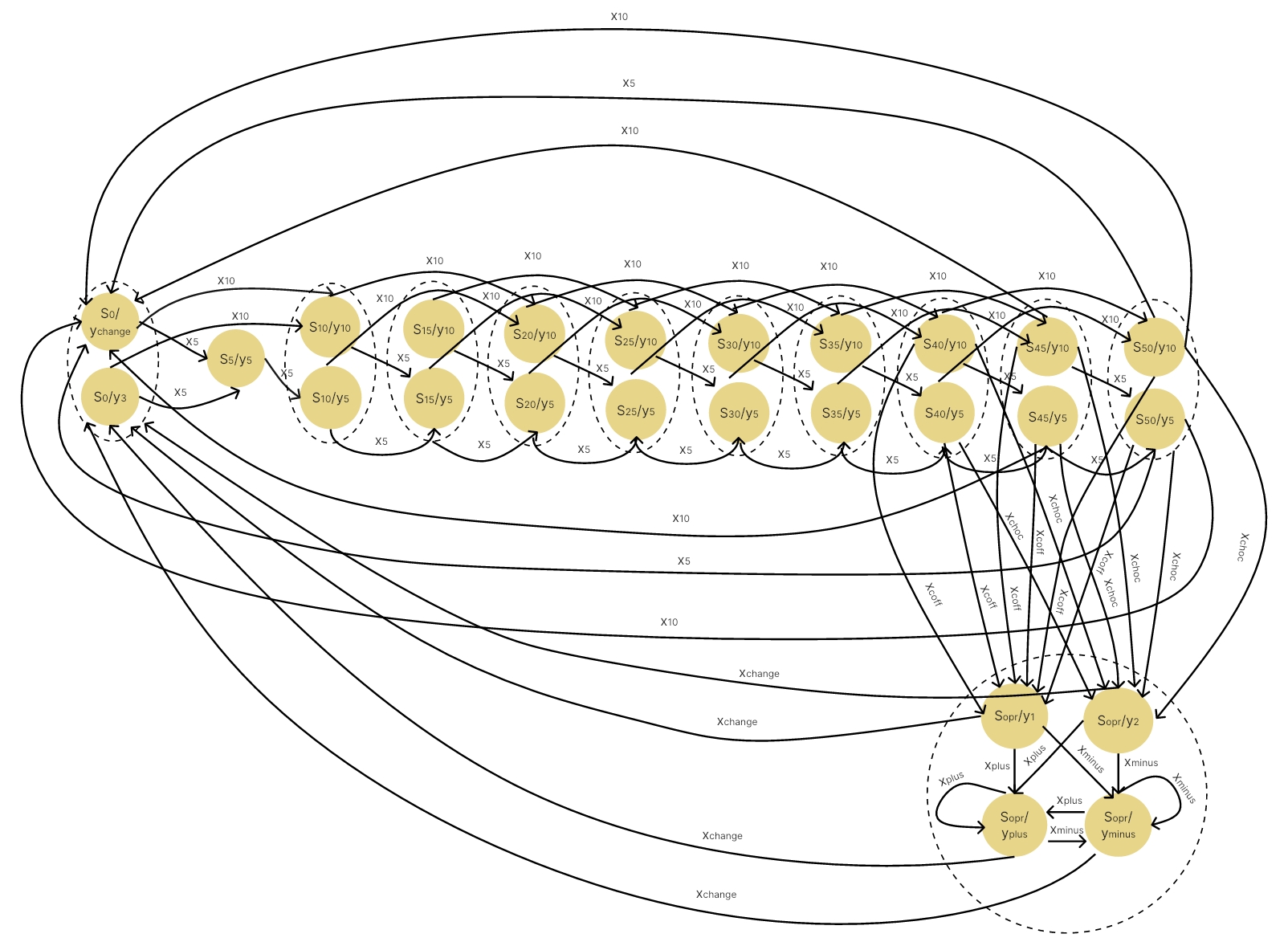


Рис. 2. Автомат Мура

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| t | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 |
| X | X10 | Xminus | X10 | Xchoc | X5 | X10 | Xchoc | X10 | Xcoff | Xminus | Xchange |  |
| S | S0 | S5 | S15 | S25 | S25 | S30 | S40 | S50 | S50 | Sopr | Sopr | S0 |
| Y | Y5 | Y10 | Y10 | - | Y5 | Y10 | Y10 | Ychange | Y1 | Yminus | Y3 |  |

Протокол работы автомата Мура

**Часть 2**

Программное моделирование работы автомата Мили

Текст программы

Файл .сpp

#include "MyForm.h"

using namespace System;

using namespace System::Windows::Forms;

[STAThreadAttribute]

int main(array<String^>^ args)

{

Application::EnableVisualStyles();

Application::SetCompatibleTextRenderingDefault(false);

Project13::MyForm form;

Application::Run(% form);

}

Файл .h

#pragma once

namespace Project13 {

using namespace System;

using namespace System::ComponentModel;

using namespace System::Collections;

using namespace System::Windows::Forms;

using namespace System::Data;

using namespace System::Drawing;

/// <summary>

/// Сводка для MyForm

/// </summary>

public ref class MyForm : public System::Windows::Forms::Form

{

public:

MyForm(void)

{

InitializeComponent();

//

//TODO: добавьте код конструктора

//

this->checkInput = gcnew array <String^>{

"X5",

"X10",

"Xcoff",

"Xchoc",

"Xplus",

"Xminus",

"Xchange",

};

this->currentState = State::S0;

}

protected:

/// <summary>

/// Освободить все используемые ресурсы.

/// </summary>

~MyForm()

{

if (components)

{

delete components;

}

}

private: System::Windows::Forms::PictureBox^ pictureBox1;

int count = 0;

String^ oneInput;

enum class State { S0, S5, S10, S15, S20, S25, S30, S35, S40, S45, S50, Sopr };

array<String^>^ input;

array<String^>^ checkInput;

State currentState;

protected:

private: System::Windows::Forms::Button^ button1;

private: System::Windows::Forms::Button^ button2;

private: System::Windows::Forms::Button^ button3;

private: System::Windows::Forms::RichTextBox^ richTextBox1;

private: System::Windows::Forms::DataGridView^ dataGridView1;

private: System::Windows::Forms::TextBox^ textBox1;

private: System::Windows::Forms::Label^ label1;

private: System::Windows::Forms::Label^ label2;

private:

/// <summary>

/// Обязательная переменная конструктора.

/// </summary>

System::ComponentModel::Container ^components;

#pragma region Windows Form Designer generated code

/// <summary>

/// Требуемый метод для поддержки конструктора — не изменяйте

/// содержимое этого метода с помощью редактора кода.

/// </summary>

void InitializeComponent(void)

{

System::ComponentModel::ComponentResourceManager^ resources = (gcnew System::ComponentModel::ComponentResourceManager(MyForm::typeid));

this->pictureBox1 = (gcnew System::Windows::Forms::PictureBox());

this->button1 = (gcnew System::Windows::Forms::Button());

this->button2 = (gcnew System::Windows::Forms::Button());

this->button3 = (gcnew System::Windows::Forms::Button());

this->richTextBox1 = (gcnew System::Windows::Forms::RichTextBox());

this->dataGridView1 = (gcnew System::Windows::Forms::DataGridView());

this->textBox1 = (gcnew System::Windows::Forms::TextBox());

this->label1 = (gcnew System::Windows::Forms::Label());

this->label2 = (gcnew System::Windows::Forms::Label());

(cli::safe\_cast<System::ComponentModel::ISupportInitialize^>(this->pictureBox1))->BeginInit();

(cli::safe\_cast<System::ComponentModel::ISupportInitialize^>(this->dataGridView1))->BeginInit();

this->SuspendLayout();

//

// pictureBox1

//

this->pictureBox1->Image = (cli::safe\_cast<System::Drawing::Image^>(resources->GetObject(L"pictureBox1.Image")));

this->pictureBox1->Location = System::Drawing::Point(0, 0);

this->pictureBox1->Name = L"pictureBox1";

this->pictureBox1->Size = System::Drawing::Size(798, 388);

this->pictureBox1->SizeMode = System::Windows::Forms::PictureBoxSizeMode::StretchImage;

this->pictureBox1->TabIndex = 0;

this->pictureBox1->TabStop = false;

//

// button1

//

this->button1->Location = System::Drawing::Point(973, 480);

this->button1->Name = L"button1";

this->button1->Size = System::Drawing::Size(137, 57);

this->button1->TabIndex = 2;

this->button1->Text = L"Пошаговый режим";

this->button1->UseVisualStyleBackColor = true;

this->button1->Click += gcnew System::EventHandler(this, &MyForm::button1\_Click);

//

// button2

//

this->button2->Location = System::Drawing::Point(823, 480);

this->button2->Name = L"button2";

this->button2->Size = System::Drawing::Size(144, 57);

this->button2->TabIndex = 3;

this->button2->Text = L"Автоматический режим";

this->button2->UseVisualStyleBackColor = true;

this->button2->Click += gcnew System::EventHandler(this, &MyForm::button2\_Click);

//

// button3

//

this->button3->Location = System::Drawing::Point(973, 552);

this->button3->Name = L"button3";

this->button3->Size = System::Drawing::Size(137, 52);

this->button3->TabIndex = 5;

this->button3->Text = L"Отчистить";

this->button3->UseVisualStyleBackColor = true;

this->button3->Click += gcnew System::EventHandler(this, &MyForm::button3\_Click);

//

// richTextBox1

//

this->richTextBox1->Location = System::Drawing::Point(814, 0);

this->richTextBox1->Name = L"richTextBox1";

this->richTextBox1->Size = System::Drawing::Size(296, 388);

this->richTextBox1->TabIndex = 6;

this->richTextBox1->Text = resources->GetString(L"richTextBox1.Text");

//

// dataGridView1

//

this->dataGridView1->ColumnHeadersHeightSizeMode = System::Windows::Forms::DataGridViewColumnHeadersHeightSizeMode::AutoSize;

this->dataGridView1->Location = System::Drawing::Point(10, 452);

this->dataGridView1->Name = L"dataGridView1";

this->dataGridView1->Size = System::Drawing::Size(798, 152);

this->dataGridView1->TabIndex = 7;

//

// textBox1

//

this->textBox1->Location = System::Drawing::Point(868, 436);

this->textBox1->Name = L"textBox1";

this->textBox1->Size = System::Drawing::Size(203, 20);

this->textBox1->TabIndex = 0;

this->textBox1->TextChanged += gcnew System::EventHandler(this, &MyForm::textBox1\_TextChanged);

//

// label1

//

this->label1->AutoSize = true;

this->label1->Location = System::Drawing::Point(12, 436);

this->label1->Name = L"label1";

this->label1->Size = System::Drawing::Size(59, 13);

this->label1->TabIndex = 9;

this->label1->Text = L"Результат";

//

// label2

//

this->label2->AutoSize = true;

this->label2->Location = System::Drawing::Point(865, 420);

this->label2->Name = L"label2";

this->label2->Size = System::Drawing::Size(116, 13);

this->label2->TabIndex = 10;

this->label2->Text = L"Ввод X через пробел ";

this->label2->Click += gcnew System::EventHandler(this, &MyForm::label2\_Click);

//

// MyForm

//

this->AutoScaleDimensions = System::Drawing::SizeF(6, 13);

this->AutoScaleMode = System::Windows::Forms::AutoScaleMode::Font;

this->ClientSize = System::Drawing::Size(1122, 616);

this->Controls->Add(this->label2);

this->Controls->Add(this->label1);

this->Controls->Add(this->textBox1);

this->Controls->Add(this->dataGridView1);

this->Controls->Add(this->richTextBox1);

this->Controls->Add(this->button3);

this->Controls->Add(this->button2);

this->Controls->Add(this->button1);

this->Controls->Add(this->pictureBox1);

this->Name = L"MyForm";

this->Text = L"Коффейный автомат";

this->Load += gcnew System::EventHandler(this, &MyForm::MyForm\_Load);

(cli::safe\_cast<System::ComponentModel::ISupportInitialize^>(this->pictureBox1))->EndInit();

(cli::safe\_cast<System::ComponentModel::ISupportInitialize^>(this->dataGridView1))->EndInit();

this->ResumeLayout(false);

this->PerformLayout();

}

#pragma endregion

private: System::Void MyForm\_Load(System::Object^ sender, System::EventArgs^ e) {

if (textBox1->Text->Length == 0) {

button1->Enabled = false;

button2->Enabled = false;

}

}

private: System::Void button3\_Click(System::Object^ sender, System::EventArgs^ e) { //реализация кнопки "Отчистить"

currentState = State::S0;

dataGridView1->Rows->Clear();

dataGridView1->Columns->Clear();

textBox1->Clear();

button1->Show();

button2->Show();

count = 0;

}

private: System::Void textBox1\_TextChanged(System::Object^ sender, System::EventArgs^ e) {

input = textBox1->Text->Split(gcnew array<String^>{" "},

StringSplitOptions::RemoveEmptyEntries);

bool flag = false;

for (int i = 0; i < input->Length; i++)

for (int j = 0; j < checkInput->Length; j++) {

if (input[i] == checkInput[j]) {

flag = true;

break;

}

else

flag = false;

}

if (textBox1->Text->Length == 0) {

button1->Enabled = false;

button2->Enabled = false;

}

else {

button1->Enabled = flag;

button2->Enabled = flag;

}

}

private: System::Void button2\_Click(System::Object^ sender, System::EventArgs^ e) { // реализация автоматического режима

button1->Hide();

currentState = State::S0;

dataGridView1->Columns->Clear();

dataGridView1->Columns->Add("", "");

dataGridView1->Rows->Add(3);

dataGridView1->Rows[0]->Cells[0]->Value = "t";

dataGridView1->Rows[1]->Cells[0]->Value = "X";

dataGridView1->Rows[2]->Cells[0]->Value = "S";

dataGridView1->Rows[3]->Cells[0]->Value = "Y";

for (int i = 0; i <= input->Length; i++) {

dataGridView1->Columns->Add("", "");

dataGridView1->Rows[0]->Cells[i+1]->Value = i;

}

dataGridView1->Rows[2]->Cells[1]->Value = currentState;

currentState = State::S0;

for (int i = 0; i < input->Length; i++) {

bool flag1 = true;

String^ X = input[i];

String^ Y;

switch (currentState) {

case State::S0:

if (X == "X5") {

currentState = State::S5;

Y = "Y5";

}

else if (X == "X10") {

currentState = State::S10;

Y = "Y10";

}

else flag1 = false;

break;

case State::S5:

if (X == "X5") {

currentState = State::S10;

Y = "Y5";

}

else if (X == "X10") {

currentState = State::S15;

Y = "Y10";

}

else flag1 = false;

break;

case State::S10:

if (X == "X5") {

currentState = State::S15;

Y = "Y5";

}

else if (X == "X10") {

currentState = State::S20;

Y = "Y10";

}

else flag1 = false;

break;

case State::S15:

if (X == "X5") {

currentState = State::S20;

Y = "Y5";

}

else if (X == "X10") {

currentState = State::S25;

Y = "Y10";

}

else flag1 = false;

break;

case State::S20:

if (X == "X5") {

currentState = State::S25;

Y = "Y5";

}

else if (X == "X10") {

currentState = State::S30;

Y = "Y10";

}

else flag1 = false;

break;

case State::S25:

if (X == "X5") {

currentState = State::S30;

Y = "Y5";

}

else if (X == "X10") {

currentState = State::S35;

Y = "Y10";

}

else flag1 = false;

break;

case State::S30:

if (X == "X5") {

currentState = State::S35;

Y = "Y5";

}

else if (X == "X10") {

currentState = State::S40;

Y = "Y10";

}

else flag1 = false;

break;

case State::S35:

if (X == "X5") {

currentState = State::S40;

Y = "Y5";

}

else if (X == "X10") {

currentState = State::S45;

Y = "Y10";

}

else flag1 = false;

break;

case State::S40:

if (X == "X5") {

currentState = State::S45;

Y = "Y5";

}

else if (X == "X10") {

currentState = State::S50;

Y = "Y10";

}

else if (X == "Xcoff") {

currentState = State::Sopr;

Y = "Y1";

}

else if (X == "Xchoc") {

currentState = State::Sopr;

Y = "Y2";

}

else flag1 = false;

break;

case State::S45:

if (X == "X5") {

currentState = State::S50;

Y = "Y5";

}

else if (X == "X10") {

currentState = State::S0;

Y = "Ychange";

}

else if (X == "Xcoff") {

currentState = State::Sopr;

Y = "Y1";

}

else if (X == "Xchoc") {

currentState = State::Sopr;

Y = "Y2";

}

else flag1 = false;

break;

case State::S50:

if (X == "X5") {

currentState = State::S0;

Y = "Ychange";

}

else if (X == "X10") {

currentState = State::S0;

Y = "Ychange";

}

else if (X == "Xcoff") {

currentState = State::Sopr;

Y = "Y1";

}

else if (X == "Xchoc") {

currentState = State::Sopr;

Y = "Y2";

}

else flag1 = false;

break;

case State::Sopr:

if (X == "Xplus") {

currentState = State::Sopr;

Y = "Yplus";

}

else if (X == "Xminus") {

currentState = State::Sopr;

Y = "Yminus";

}

else if (X == "Xchange") {

currentState = State::S0;

Y = "Y3";

}

else flag1 = false;

break;

}

dataGridView1->Rows[1]->Cells[i + 1]->Value = X;

if (flag1) {

dataGridView1->Rows[2]->Cells[i + 2]->Value = currentState;

dataGridView1->Rows[3]->Cells[i + 1]->Value = Y;

}

else {

dataGridView1->Rows[2]->Cells[i + 2]->Value = currentState;

dataGridView1->Rows[3]->Cells[i + 1]->Value = "-";

}

}

}

private: System::Void label2\_Click(System::Object^ sender, System::EventArgs^ e) {

}

private: System::Void button1\_Click(System::Object^ sender, System::EventArgs^ e) { // реализация пошагового режима

if (count == 0) {

button2->Hide();

currentState = State::S0;

dataGridView1->Columns->Clear();

dataGridView1->Columns->Add("", "");

dataGridView1->Rows->Add(3);

dataGridView1->Rows[0]->Cells[0]->Value = "t";

dataGridView1->Rows[1]->Cells[0]->Value = "x";

dataGridView1->Rows[2]->Cells[0]->Value = "S";

dataGridView1->Rows[3]->Cells[0]->Value = "y";

for (int i = 0; i <= 1; i++) {

dataGridView1->Columns->Add("", "");

dataGridView1->Rows[0]->Cells[i + 1]->Value = i;

}

dataGridView1->Rows[2]->Cells[count + 1]->Value = "S0";

}

else {

dataGridView1->Columns->Add("", "");

dataGridView1->Rows[0]->Cells[count + 2]->Value = count + 1;

}

oneInput = textBox1->Text;

dataGridView1->Rows[1]->Cells[count + 1]->Value = oneInput;

bool flag = true;

String^ X = oneInput;

String^ Y;

switch (currentState) {

case State::S0:

if (X == "X5") {

currentState = State::S5;

Y = "Y5";

}

else if (X == "X10") {

currentState = State::S10;

Y = "Y10";

}

else flag = false;

break;

case State::S5:

if (X == "X5") {

currentState = State::S10;

Y = "Y5";

}

else if (X == "X10") {

currentState = State::S15;

Y = "Y10";

}

else flag = false;

break;

case State::S10:

if (X == "X5") {

currentState = State::S15;

Y = "Y5";

}

else if (X == "X10") {

currentState = State::S20;

Y = "Y10";

}

else flag = false;

break;

case State::S15:

if (X == "X5") {

currentState = State::S20;

Y = "Y5";

}

else if (X == "X10") {

currentState = State::S25;

Y = "Y10";

}

else flag = false;

break;

case State::S20:

if (X == "X5") {

currentState = State::S25;

Y = "Y5";

}

else if (X == "X10") {

currentState = State::S30;

Y = "Y10";

}

else flag = false;

break;

case State::S25:

if (X == "X5") {

currentState = State::S30;

Y = "Y5";

}

else if (X == "X10") {

currentState = State::S35;

Y = "Y10";

}

else flag = false;

break;

case State::S30:

if (X == "X5") {

currentState = State::S35;

Y = "Y5";

}

else if (X == "X10") {

currentState = State::S40;

Y = "Y10";

}

else flag = false;

break;

case State::S35:

if (X == "X5") {

currentState = State::S40;

Y = "Y5";

}

else if (X == "X10") {

currentState = State::S45;

Y = "Y10";

}

else flag = false;

break;

case State::S40:

if (X == "X5") {

currentState = State::S45;

Y = "Y5";

}

else if (X == "X10") {

currentState = State::S50;

Y = "Y10";

}

else if (X == "Xcoff") {

currentState = State::Sopr;

Y = "Y1";

}

else if (X == "Xchoc") {

currentState = State::Sopr;

Y = "Y2";

}

else flag = false;

break;

case State::S45:

if (X == "X5") {

currentState = State::S50;

Y = "Y5";

}

else if (X == "X10") {

currentState = State::S0;

Y = "Ychange";

}

else if (X == "Xcoff") {

currentState = State::Sopr;

Y = "Y1";

}

else if (X == "Xchoc") {

currentState = State::Sopr;

Y = "Y2";

}

else flag = false;

break;

case State::S50:

if (X == "X5") {

currentState = State::S0;

Y = "Ychange";

}

else if (X == "X10") {

currentState = State::S0;

Y = "Ychange";

}

else if (X == "Xcoff") {

currentState = State::Sopr;

Y = "Y1";

}

else if (X == "Xchoc") {

currentState = State::Sopr;

Y = "Y2";

}

else flag = false;

break;

case State::Sopr:

if (X == "Xplus") {

currentState = State::Sopr;

Y = "Yplus";

}

else if (X == "Xminus") {

currentState = State::Sopr;

Y = "Yminus";

}

else if (X == "Xchange") {

currentState = State::S0;

Y = "Y3";

}

else flag = false;

break;

}

dataGridView1->Rows[1]->Cells[count + 1]->Value = X;

if (flag) {

dataGridView1->Rows[2]->Cells[count + 2]->Value = currentState;

dataGridView1->Rows[3]->Cells[count + 1]->Value = Y;

}

else {

dataGridView1->Rows[2]->Cells[count + 2]->Value = currentState;

dataGridView1->Rows[3]->Cells[count + 1]->Value = "-";

}

textBox1->Text = "";

count += 1;

}

};

}

Тесты:

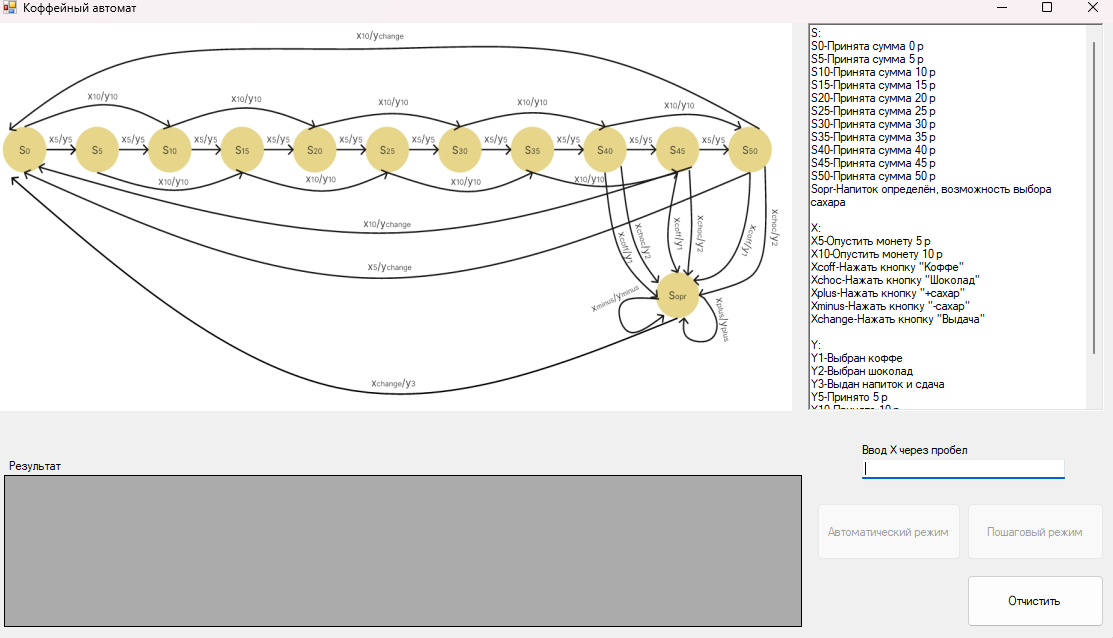


Рис. 3. Интерфейс приложения

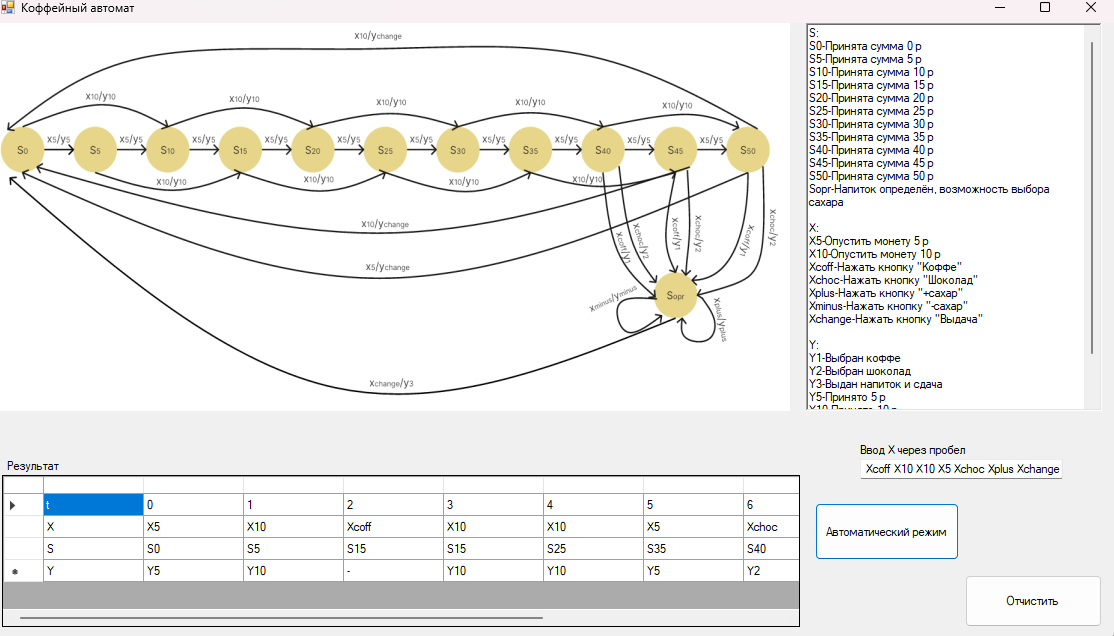


Рис. 4. Автоматический режим

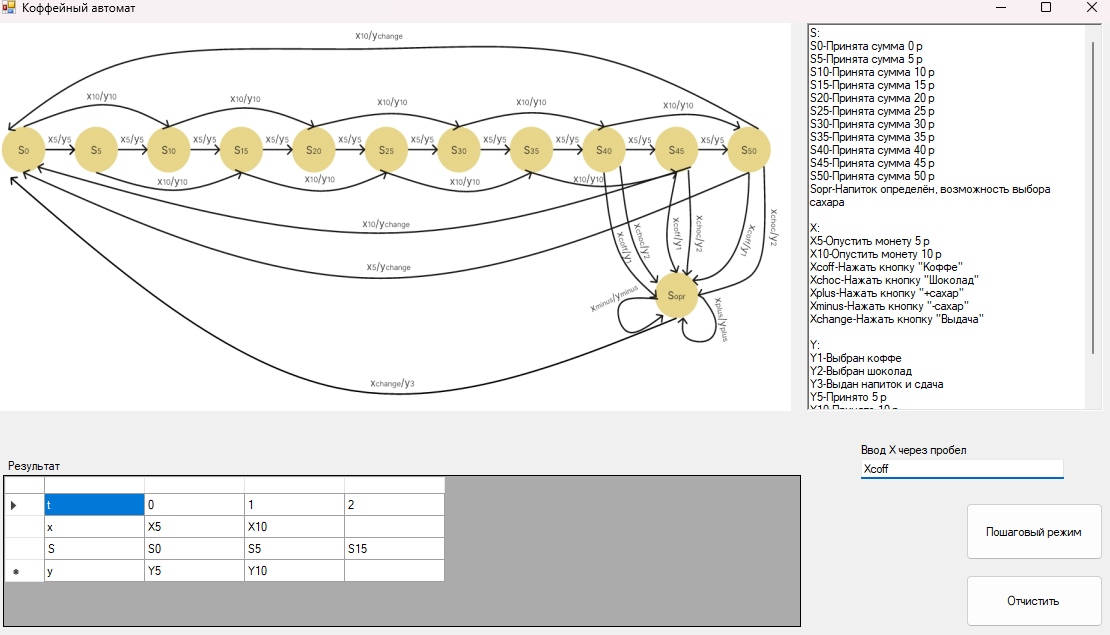


Рис. 5. Пошаговый режим

Вывод

В ходе первой лабораторной работы по теории автоматов изучен конечный детерминированный автомат Мили, который эмулирует работу автомата выдачи напитков. В этой работе разобраны, как данный автомат может быть описан с помощью состояний, событий и переходов между ними.

Начало функционирования автомата напитков начинается с его включения, когда пользователь видит главный экран (S0). Пользователь взаимодействует с автоматом, вставляя монеты (X5, X10), выбирая напиток (Xchoc, Xcoff) и осуществляя различные операции, такие как выбор сахара или запрос сдачи. Автомат перемещается между различными состояниями в зависимости от входных событий и выполняет соответствующие операции, например, выдачу напитка или сдачи.

Также созданы таблицы, описывающие переходы и выходы для автомата Мили, проведена работа с определенной последовательностью событий. Это позволило более глубоко понять, как автомат реагирует на разнообразные входные данные и какие события генерирует в ответ.

Эта лабораторная работа также подчеркнула важность использования формальных методов для моделирования и анализа сложных систем, таких как автоматы. Теория автоматов предоставляет необходимые инструменты для понимания и оптимизации работы подобных систем.

В результате данная работа помогла более глубоко понять функционирование автомата выдачи напитков с помощью конечных автоматов Мили. Она также подтвердила значимость формального описания систем для их дальнейшего анализа и усовершенствования.