МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное

образовательное учреждение высшего образования

«ЧЕРЕПОВЕЦКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»

Институт Информационных Технологий

Кафедра МПО ЭВМ

Дисциплина «Теория автоматов и формальных языков»

Лабораторная работа №1-2

«Построение модели автомата Мили и Мура»

Выполнил:

студент группы 1ПИб-02-3оп-22

Маркелов Сергей Александрович

Проверил:

доцент, к.т.н.

Ганичева Оксана Георгиевна

Череповец, 2023 год

1. Словесное описание работы автомата

В рамках выполнения лабораторной работы «Построение модели автомата Мили и Мура» был рассмотрен автомат «Банкомат Сбербанка»

Изначально банкомат отображает начальный экран ожидания карты. Пользователь может либо вставить карту (контактный вход), либо приложить её к NFC-сканеру (бесконтактный вход).

Далее банкомат запрашивает PIN-код от карты. В случае ввода неверного PIN-кода он вновь запрашивает PIN-код, и будет делать это до тех пор, пока не будет введен верный PIN-код.

После этого банкомат отобразит главный экран выбора операции. Доступно 3 операции: снятие денег, внесение денег, оплата услуг. Также с этого же экрана можно выйти обратно на начальный (если вход был бесконтактным), или же запросить возврат карты и уже затем вернуться на начальный экран (если вход был контактным).

После выбора операции банкомат, в зависимости от выбранной операции, попросит либо ввести сумму для снятия, либо внести в банкомат нужную сумму денег, либо выбрать нужную услугу. Если операция выбрана неверно, можно вернуться на главный экран. Если она выбрана верно, необходимо совершить запрашиваемое действие и затем подтвердить операцию.

Далее банкомат выполняет операцию и после этого выводит окно результата операции вместе с балансом.

Затем он возвращается на главный экран, где можно выбрать еще какую-либо еще операцию, либо вернуть карту (при контактном входе) и выйти на начальный экран.

1. Формальное описание работы автомата

|  |  |
| --- | --- |
| **S:** | **Y:** |
| S0 – ожидание карты | y0 – сообщение «Карта считана», отображение экрана ввода пин-кода |
| S1 – принята карта | y1 – сообщение «Карта принята», отображение экрана ввода пин-кода |
| S2 – пин-код принят | у2 – сообщение «Пин-код неверный» |
| S3 – выбрана операция снятия денег | у3 – сообщение «Пин-код верный», отображение главного экрана |
| S4 – выбрана операция внесения денег | у4 – сообщение «Операция выбрана» |
| S5 – выбрана операция оплаты услуг | у5 – отображение главного экрана |
| S6 – введена сумма, которую необходимо снять | у6 – отображение главного экрана |
| S7 – внесена сумма, которую необходимо положить | у7 – отображение главного экрана |
| S8 – выбрана необходимая услуга | у8 – отображение экрана операции |
| S9 – операция подтверждена | у9 – вывод введенной суммы |
| S10 – операция выполнена | у10 – вывод внесенной суммы |
| S11 – карта возвращена | у11 – вывод выбранной услуги |
| **X:** | у12 – вывод результатов операции и баланса |
| x0 – приложена карта (бесконтактный вход) | у13 – отображение главного экрана |
| x1 – вставлена карта (контактный вход) | у14 – сообщение «Заберите карту» |
| x2 – ввод неверного пин-кода | у15 – вывод начального экрана |
| x3 – ввод верного пин-кода | у16 – вывод начального экрана |
| x4 – выбор операции снятия денег |  |
| x5 – выбор операции внесения денег |  |
| x6 – выбор операции оплаты услуг |  |
| x7 – нажата кнопка «Вернуться на главный экран» |  |
| x8 – ввод суммы, которую необходимо снять |  |
| x9 – внесение суммы, которую необходимо положить |  |
| x10 – выбор услуги |  |
| x11 – подтверждение операции |  |
| х12 – проведение операции |  |
| х13 – запрос на возврат карты |  |
| х14 – выход из системы |  |

Функция переходов δ: S \* X → S

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| X S | S0 | S1 | S2 | S3 | S4 | S5 | S6 | S7 | S8 | S9 | S10 | S11 |
| x0 | S1 |  |  |  |  |  |  |  |  |  |  |  |
| x1 | S1 |  |  |  |  |  |  |  |  |  |  |  |
| x2 |  | S1 |  |  |  |  |  |  |  |  |  |  |
| x3 |  | S2 |  |  |  |  |  |  |  |  |  |  |
| x4 |  |  | S3 |  |  |  |  |  |  |  |  |  |
| x5 |  |  | S4 |  |  |  |  |  |  |  |  |  |
| x6 |  |  | S5 |  |  |  |  |  |  |  |  |  |
| x7 |  |  |  | S2 | S2 | S2 |  |  |  |  | S2 |  |
| x8 |  |  |  | S6 |  |  |  |  |  |  |  |  |
| x9 |  |  |  |  | S7 |  |  |  |  |  |  |  |
| x10 |  |  |  |  |  | S8 |  |  |  |  |  |  |
| x11 |  |  |  |  |  |  | S9 | S9 | S9 |  |  |  |
| x12 |  |  |  |  |  |  |  |  |  | S10 |  |  |
| x13 |  |  | S11 |  |  |  |  |  |  |  |  |  |
| x14 |  |  | S0 |  |  |  |  |  |  |  |  | S0 |

Функция выводов λ: S \* X → Y

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| X S | S0 | S1 | S2 | S3 | S4 | S5 | S6 | S7 | S8 | S9 | S10 | S11 |
| x0 | y0 |  |  |  |  |  |  |  |  |  |  |  |
| x1 | y1 |  |  |  |  |  |  |  |  |  |  |  |
| x2 |  | y2 |  |  |  |  |  |  |  |  |  |  |
| x3 |  | y3 |  |  |  |  |  |  |  |  |  |  |
| x4 |  |  | y4 |  |  |  |  |  |  |  |  |  |
| x5 |  |  | y4 |  |  |  |  |  |  |  |  |  |
| x6 |  |  | y4 |  |  |  |  |  |  |  |  |  |
| x7 |  |  |  | y5 | y6 | y7 |  |  |  |  | y13 |  |
| x8 |  |  |  | y8 |  |  |  |  |  |  |  |  |
| x9 |  |  |  |  | y8 |  |  |  |  |  |  |  |
| x10 |  |  |  |  |  | y8 |  |  |  |  |  |  |
| x11 |  |  |  |  |  |  | y9 | y10 | y11 |  |  |  |
| x12 |  |  |  |  |  |  |  |  |  | y12 |  |  |
| x13 |  |  | y14 |  |  |  |  |  |  |  |  |  |
| x14 |  |  | y15 |  |  |  |  |  |  |  |  | y16 |

Совмещенная функция δ / λ:

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| X S | S0 | S1 | S2 | S3 | S4 | S5 | S6 | S7 | S8 | S9 | S10 | S11 |
| x0 | S1  y0 |  |  |  |  |  |  |  |  |  |  |  |
| x1 | S1  y1 |  |  |  |  |  |  |  |  |  |  |  |
| x2 |  | S1  y2 |  |  |  |  |  |  |  |  |  |  |
| x3 |  | S2  y3 |  |  |  |  |  |  |  |  |  |  |
| x4 |  |  | S3  y4 |  |  |  |  |  |  |  |  |  |
| x5 |  |  | S4  y4 |  |  |  |  |  |  |  |  |  |
| x6 |  |  | S5  y4 |  |  |  |  |  |  |  |  |  |
| x7 |  |  |  | S2  y5 | S2  y6 | S2  y7 |  |  |  |  | S2  y13 |  |
| x8 |  |  |  | S6  y8 |  |  |  |  |  |  |  |  |
| x9 |  |  |  |  | S7  y8 |  |  |  |  |  |  |  |
| x10 |  |  |  |  |  | S8  y8 |  |  |  |  |  |  |
| x11 |  |  |  |  |  |  | S9  y9 | S9  y10 | S9  y11 |  |  |  |
| x12 |  |  |  |  |  |  |  |  |  | S10  y12 |  |  |
| x13 |  |  | S11  y14 |  |  |  |  |  |  |  |  |  |
| x14 |  |  | S0  y15 |  |  |  |  |  |  |  |  | S0  y16 |

5. Программное моделирование работы автомата Мили

Программная модель представлена на рис. 1.

Реализация с помощью программного кода:

void Mili(int x[], int y[], int s[], int i, int j, int& err) {

switch (s[i]) {

case 0: {

if (x[i] == 0) {

s[i + 1] = 1;

y[i] = 0;

}

else if (x[i] == 1) {

s[i + 1] = 1;

y[i] = 1;

}

else err++;

}

break;

case 1: {

if (x[i] == 2) {

s[i + 1] = 1;

y[i] = 2;

}

else if (x[i] == 3) {

s[i + 1] = 2;

y[i] = 3;

}

else err++;

}

break;

case 2: {

if (x[i] == 4) {

s[i + 1] = 3;

y[i] = 4;

}

else if (x[i] == 5) {

s[i + 1] = 4;

y[i] = 4;

}

else if (x[i] == 6) {

s[i + 1] = 5;

y[i] = 4;

}

else if (x[i] == 13) {

s[i + 1] = 11;

y[i] = 14;

}

else if (x[i] == 14) {

s[i + 1] = 0;

y[i] = 15;

}

else err++;

}

break;

case 3: {

if (x[i] == 7) {

s[i + 1] = 2;

y[i] = 5;

}

else if (x[i] == 8) {

s[i + 1] = 6;

y[i] = 8;

}

else err++;

}

break;

case 4: {

if (x[i] == 7) {

s[i + 1] = 2;

y[i] = 6;

}

else if (x[i] == 9) {

s[i + 1] = 7;

y[i] = 8;

}

else err++;

}

break;

case 5: {

if (x[i] == 7) {

s[i + 1] = 2;

y[i] = 7;

}

else if (x[i] == 10) {

s[i + 1] = 8;

y[i] = 8;

}

else err++;

}

break;

case 6: {

if (x[i] == 11) {

s[i + 1] = 9;

y[i] = 9;

}

else err++;

}

break;

case 7: {

if (x[i] == 11) {

s[i + 1] = 9;

y[i] = 10;

}

else err++;

}

break;

case 8: {

if (x[i] == 11) {

s[i + 1] = 9;

y[i] = 11;

}

else err++;

}

break;

case 9: {

if (x[i] == 12) {

s[i + 1] = 10;

y[i] = 12;

}

else err++;

}

break;

case 10: {

if (x[i] == 7) {

s[i + 1] = 2;

y[i] = 13;

}

else err++;

}

break;

case 11: {

if (x[i] == 14) {

s[i + 1] = 0;

y[i] = 16;

}

else err++;

}

break;

}

}

#pragma endregion

int click = 0;

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

setlocale(LC\_ALL, "Russian");

int n = 0;

String^ sx = textBox2->Text;

string str\_x = marshal\_as<string>(sx), str\_y = "", str\_s = "";

for (int i = 0; i < str\_x.length(); i++)

if (i != 0)

if ((str\_x[i] == ' ' || str\_x[i] == '\0') && str\_x[i - 1] != ' ') n++;

int k = 0, j = 0, err = 0;

int\* x = new int[n + 1];

int\* y = new int[n + 1];

int\* s = new int[n + 2];

for (int i = 0; i < str\_x.length(); i++) {

k \*= 10;

if (str\_x[i] == '0' && err == 0) k = k;

else if (str\_x[i] == '1' && err == 0) k++;

else if (str\_x[i] == '2' && err == 0) k += 2;

else if (str\_x[i] == '3' && err == 0) k += 3;

else if (str\_x[i] == '4' && err == 0) k += 4;

else if (str\_x[i] == '5' && err == 0) k += 5;

else if (str\_x[i] == '6' && err == 0) k += 6;

else if (str\_x[i] == '7' && err == 0) k += 7;

else if (str\_x[i] == '8' && err == 0) k += 8;

else if (str\_x[i] == '9' && err == 0) k += 9;

else if (str\_x[i] == ' ' && err == 0 && i != 0) {

if (str\_x[i - 1] == ' ') {

if (err == 0) err++;

}

if (str\_x[i + 1] != '\0') {

k /= 10;

x[j] = k;

j++;

k = 0;

}

else k /= 10;

}

else {

if (err == 0) err++;

}

}

x[j] = k;

s[0] = 0;

if (err == 0 && click <= j)

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

Mili(x, y, s, i, click, err);

}

else err++;

if (err == 0) {

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

if (s[i] == 0) str\_s += "0 ";

else if (s[i] == 1) str\_s += "1 ";

else if (s[i] == 2) str\_s += "2 ";

else if (s[i] == 3) str\_s += "3 ";

else if (s[i] == 4) str\_s += "4 ";

else if (s[i] == 5) str\_s += "5 ";

else if (s[i] == 6) str\_s += "6 ";

else if (s[i] == 7) str\_s += "7 ";

else if (s[i] == 8) str\_s += "8 ";

else if (s[i] == 9) str\_s += "9 ";

else if (s[i] == 10) str\_s += "10 ";

else if (s[i] == 11) str\_s += "11 ";

}

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

if (y[i] == 0) str\_y += "0 ";

else if (y[i] == 1) str\_y += "1 ";

else if (y[i] == 2) str\_y += "2 ";

else if (y[i] == 3) str\_y += "3 ";

else if (y[i] == 4) str\_y += "4 ";

else if (y[i] == 5) str\_y += "5 ";

else if (y[i] == 6) str\_y += "6 ";

else if (y[i] == 7) str\_y += "7 ";

else if (y[i] == 8) str\_y += "8 ";

else if (y[i] == 9) str\_y += "9 ";

else if (y[i] == 10) str\_y += "10 ";

else if (y[i] == 11) str\_y += "11 ";

else if (y[i] == 12) str\_y += "12 ";

else if (y[i] == 13) str\_y += "13 ";

else if (y[i] == 14) str\_y += "14 ";

else if (y[i] == 15) str\_y += "15 ";

else if (y[i] == 16) str\_y += "16 ";

}

}

String^ ss = marshal\_as<String^>(str\_s);

String^ sy = marshal\_as<String^>(str\_y);

if (err == 0) {

textBox4->Text = sx;

textBox1->Text = sy;

textBox3->Text = ss;

}

else {

textBox4->Text = "Ошибка!";

textBox1->Text = "";

textBox3->Text = "";

}

click++;

delete[] x;

delete[] y;

delete[] s;

}

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

setlocale(LC\_ALL, "Russian");

int n = 0;

String^ sx = textBox2->Text;

string str\_x = marshal\_as<string>(sx), str\_y = "", str\_s = "";

for (int i = 0; i < str\_x.length(); i++)

if (i != 0)

if ((str\_x[i] == ' ' || str\_x[i] == '\0') && str\_x[i - 1] != ' ') n++;

int k = 0, j = 0, err = 0;

int\* x = new int[n + 1];

int\* y = new int[n + 1];

int\* s = new int[n + 2];

for (int i = 0; i < str\_x.length(); i++) {

k \*= 10;

if (str\_x[i] == '0' && err == 0) k = k;

else if (str\_x[i] == '1' && err == 0) k++;

else if (str\_x[i] == '2' && err == 0) k += 2;

else if (str\_x[i] == '3' && err == 0) k += 3;

else if (str\_x[i] == '4' && err == 0) k += 4;

else if (str\_x[i] == '5' && err == 0) k += 5;

else if (str\_x[i] == '6' && err == 0) k += 6;

else if (str\_x[i] == '7' && err == 0) k += 7;

else if (str\_x[i] == '8' && err == 0) k += 8;

else if (str\_x[i] == '9' && err == 0) k += 9;

else if (str\_x[i] == ' ' && err == 0 && i != 0) {

if (str\_x[i - 1] == ' ') {

if (err == 0) err++;

}

if (str\_x[i + 1] != '\0') {

k /= 10;

x[j] = k;

j++;

k = 0;

}

else k /= 10;

}

else {

if (err == 0) err++;

}

}

x[j] = k;

s[0] = 0;

if (err == 0)

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

Mili(x, y, s, i, j, err);

}

if (err == 0) {

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

if (s[i] == 0) str\_s += "0 ";

else if (s[i] == 1) str\_s += "1 ";

else if (s[i] == 2) str\_s += "2 ";

else if (s[i] == 3) str\_s += "3 ";

else if (s[i] == 4) str\_s += "4 ";

else if (s[i] == 5) str\_s += "5 ";

else if (s[i] == 6) str\_s += "6 ";

else if (s[i] == 7) str\_s += "7 ";

else if (s[i] == 8) str\_s += "8 ";

else if (s[i] == 9) str\_s += "9 ";

else if (s[i] == 10) str\_s += "10 ";

else if (s[i] == 11) str\_s += "11 ";

}

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

if (y[i] == 0) str\_y += "0 ";

else if (y[i] == 1) str\_y += "1 ";

else if (y[i] == 2) str\_y += "2 ";

else if (y[i] == 3) str\_y += "3 ";

else if (y[i] == 4) str\_y += "4 ";

else if (y[i] == 5) str\_y += "5 ";

else if (y[i] == 6) str\_y += "6 ";

else if (y[i] == 7) str\_y += "7 ";

else if (y[i] == 8) str\_y += "8 ";

else if (y[i] == 9) str\_y += "9 ";

else if (y[i] == 10) str\_y += "10 ";

else if (y[i] == 11) str\_y += "11 ";

else if (y[i] == 12) str\_y += "12 ";

else if (y[i] == 13) str\_y += "13 ";

else if (y[i] == 14) str\_y += "14 ";

else if (y[i] == 15) str\_y += "15 ";

else if (y[i] == 16) str\_y += "16 ";

}

}

String^ ss = marshal\_as<String^>(str\_s);

String^ sy = marshal\_as<String^>(str\_y);

if (err == 0) {

textBox4->Text = sx;

textBox1->Text = sy;

textBox3->Text = ss;

}

else {

textBox4->Text = "Ошибка!";

textBox1->Text = "";

textBox3->Text = "";

}

delete[] x;

delete[] y;

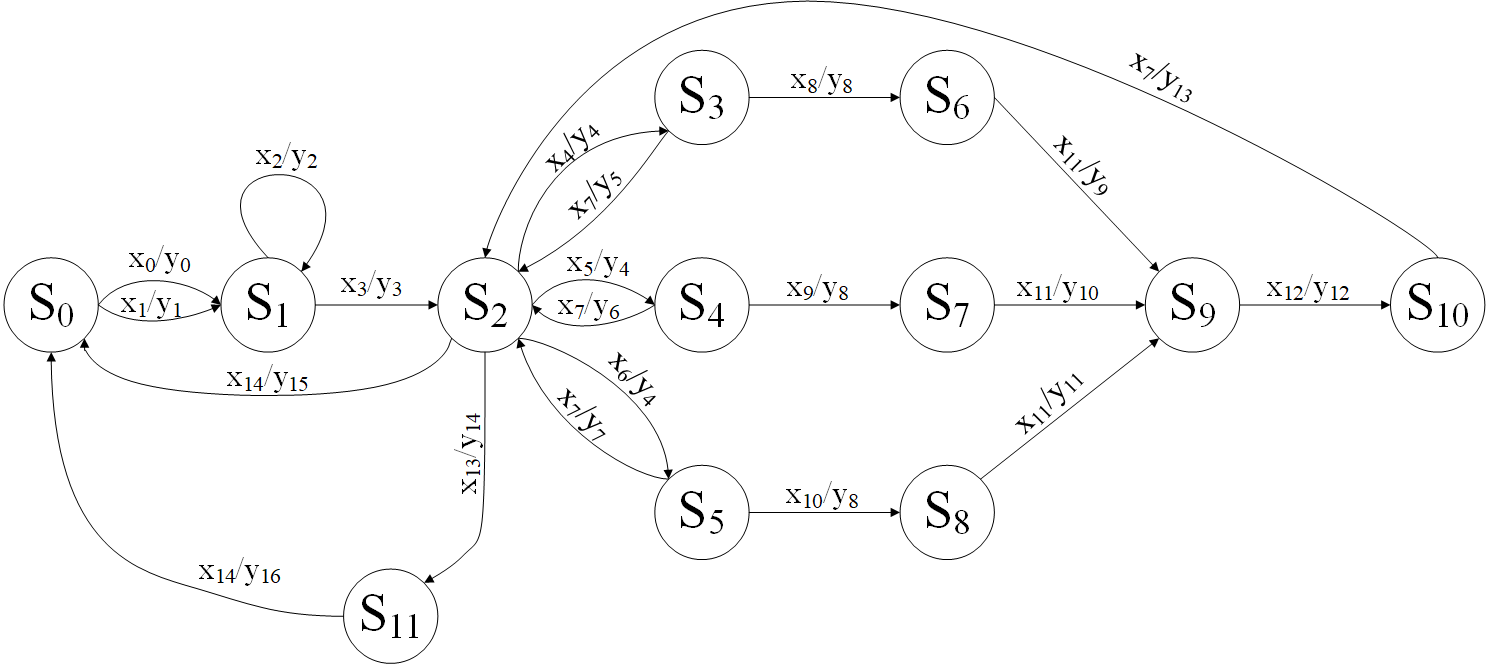
delete[] s;

}

6. Вывод

В ходе выполнения лабораторной работы были построены автоматы Мили и Мура для автомата «Банкомат Сбербанка», даны их словесное и формальное описание, рассмотрен протокол работы на примере входной последовательности из 20 входных символов. Автомат Мили был смоделирован с помощью языка программирования С++.

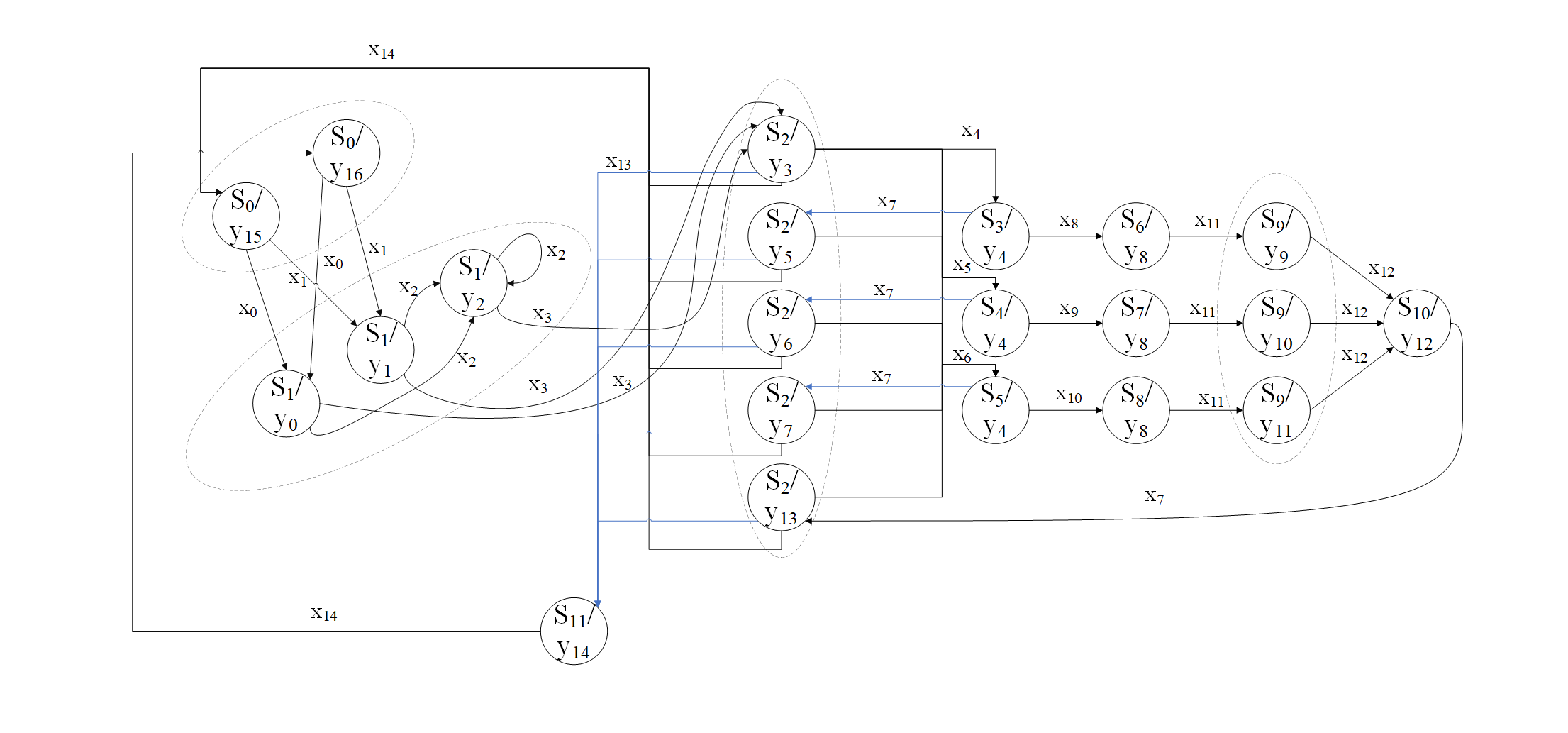
1. Автомат Мили



Протокол работы автомата Мили:

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| t | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 |
| X | x1 | x2 | x3 | x6 | x7 | x5 | x7 | x4 | x8 | x11 | x12 | x7 | x5 | x7 | x6 | x10 | x11 | x12 | x7 | x14 |  |
| S | S0 | S1 | S1 | S2 | S5 | S2 | S4 | S2 | S3 | S6 | S9 | S10 | S2 | S4 | S2 | S5 | S8 | S9 | S10 | S2 | S0 |
| Y | y1 | y2 | y3 | y4 | y7 | y4 | y6 | y4 | y8 | y9 | y12 | y13 | y4 | y6 | y4 | y8 | y11 | y12 | y13 | y15 |  |

1. Автомат Мура



Протокол работы автомата Мура:

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| t | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 |
| X | x1 | x2 | x3 | x6 | x7 | x5 | x7 | x4 | x8 | x11 | x12 | x7 | x5 | x7 | x6 | x10 | x11 | x12 | x7 | x14 |  |
| S | S0 | S1 | S1 | S2 | S5 | S2 | S4 | S2 | S3 | S6 | S9 | S10 | S2 | S4 | S2 | S5 | S8 | S9 | S10 | S2 | S0 |
| Y | y1 | y2 | y3 | y4 | y7 | y4 | y6 | y4 | y8 | y9 | y12 | y13 | y4 | y6 | y4 | y8 | y11 | y12 | y13 | y15 |  |

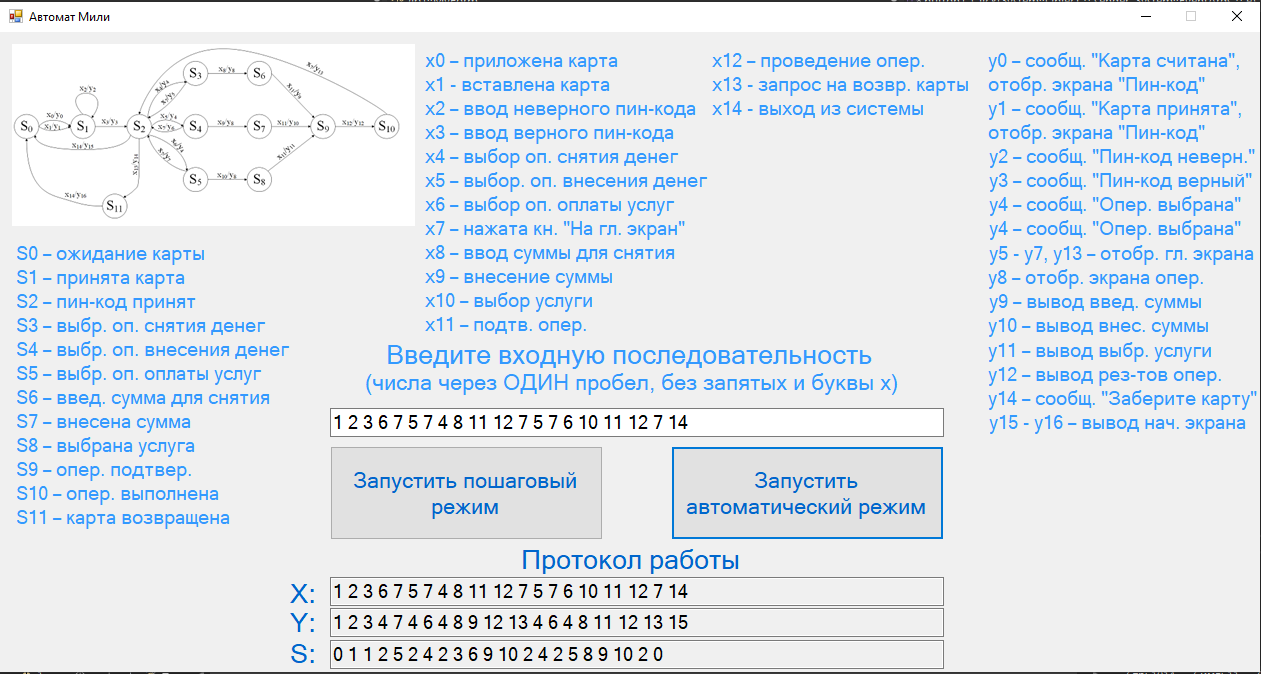


Рис. 1. Программная модель автомата Мили