УДК 519.71:519.95

РАЗРАБОТКА ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ АВТОМАТИЗАЦИИ ПРЕОБРАЗОВАНИЯ КОНЕЧНОГО АВТОМАТА В СЕТЬ ПЕТРИ

ГУСЕЙНЗАДЕ ШАХЛА СУРХАЙ гызы
Сумгаитский государственный университет, доцент
shahla.huseynzade@gmail.com

Ключевые слова: *сети Петри, конечный автомат, матрица инцидентностей, таблицы переходов и выходов, С++.*

Предложена компьютерная реализация процесса автоматизации преобразования конечного автомата (КА) в сети Петри (СП). Программный модуль разработан кроссплатформенным объектно-ориентированным языком программирования С ++, на основе детального алгоритма в виде блок схемы. Представлен алгоритм илистинг программного модуля. В интерактивном режиме вводятся матрицы переходов и выходов КА, выводятся полученные в результате выполнения программы матрицы входных и выходных инциденцийСП, имитирующей КА.

Введение. КА относятся к важнейшим основным понятиям информатики, служат для описания и анализа различных систем и процессов. [1]. КА рассматривается как абстрактные модели устройств, обрабатывающих данные, обращая в основном внимание на входно-выходное поведение, т. е. на определяемое автоматом отображение между входным и выходным множествами слов. При этом особое значение дается конструктивным и алгоритмическим аспектам проблемы.

В настоящее время практически нет систем управления без использования микроконтроллеров, где верхний уровень управления представляется аппаратно реализуемым сложным автоматом. Такая система более надежна, но сложна для программирования, т.к. содержит интеллектуальные алгоритмы управления для принятия решений [2].

Автоматный подход при нескольких десятках состояний и логических условий приводит к необходимости реализации очень сложной программной модели. Для автоматного программирования возникает необходимость совершенствования, как программных моделей, так и структурной организации автоматов [3] .

Способность описания класса сетей Петри больше, чем характеристика класса конечных автоматов, а класс ограниченных сетей Петри обладает равной способностью описания с классом конечного автомата, а поведенческие свойства ограниченной сети Петри эквивалентны поведенческим свойствам конечного автомата, особенно модели перехода состояния [4].

Поэтому представляет интерес методика построения СП по автоматному представлению. Имеются некоторые общие подходы к этой задаче. Но конкретные решения этой задачи и автоматизация решения в литературе не встречаются.Цель статьи – привлечь инженеров к технике, которая может дать эффективные решения для практических проблем, полностью используя скрытую форму линейной алгебры, которая часто превосходит интуицию.

В работе [5] рассмотрена задача автоматизации преобразования КА в СП, произведен теоретический анализ преобразования КА в СП, исследована топология, определены основные этапы автоматизации преобразования КА в СП.построены этапы (обобщенный алгоритм) автоматизации преобразования КА в СП:

Этап 1. Создание матриц переходов $C = \{c_{j,i}\}$ и выходов $B = \{b_{j,i}\}$, соответственно на основе таблиц переходов и выходов КА:

$$c_{j,i} = \begin{cases} k, \textit{если } \varphi\colon (x_i, u_j) \to x_k; \\ 0, \textit{если } \varphi\colon (x_i, u_j) \to \varepsilon; \end{cases} \text{ где } i = \overline{1,z};\ k = \overline{1,z};\ j = \overline{1,m}. \quad (1)$$

$$b_{j,i} = \begin{cases} k, \textit{если } \psi\colon (x_i, u_j) \to y_k; \\ 0, \textit{если } \psi\colon (x_i, u_j) \to \varepsilon; \end{cases} \text{ где } i = \overline{1,z};\ k = \overline{1,g};\ j = \overline{1,m}. \quad (2)$$

Этап 2. Определение матрицы входных и выходных инциденций СП$F = \{f_{i,j}\}$,$H = \{h_{j,i}\}$, где $i = \overline{1,n}; j = \overline{1,m}; n = m + z + g$; (n – число позиций, m – число переходов):

$$f_{i,j} = \begin{cases} 1, & \textit{если } p_i \in {}^{\bullet}t_j; \\ 0, & \textit{в противном случае.} \end{cases}$$

$$h_{j,i} = \begin{cases} 1, & \textit{если } p_i \in t_j^{\bullet}; \\ 0, & \textit{в противном случае.} \end{cases}$$

Этап 3. Формирование элементов матрицы входных и выходных инциденций СП $F = \{f_{i,j}\}$ и $H = \{h_{j,i}\}$:

3.1. обнуление всех элементов матрицы $f_{i,j} = 0$, где$i = \overline{1,n}; j = \overline{1,m}$;

3.2. обнуление всех элементов матрицы $h_{j,i} = 0$, где$j = \overline{1,m};\ i = \overline{1,n}$;

3.3. если $c_{j,i} \neq 0$ тогда $f_{j,j} = 1$ и$f_{m+i,j} = 1,\ h_{j,m+l} = 1,$ где$i = \overline{1,z}; j = \overline{1,m}; l = c_{j,i}$.

3.4. если $b_{j,i} \neq 0$ тогда $h_{j,m+z+l} = 1,$ где $j = \overline{1,m};\ i = \overline{1,z};\ l = b_{j,i}$.

Где $U = (u_1,\ u_2, \dots,\ u_m)$– конечное множество входных сигналов; $Y = (y_1,\ y_2, \dots,\ y_g)$ – конечное множество выходных сигналов; $X = (x_1,\ x_2, \dots,\ x_z)$ – множество внутренних состояний автомата.

Алгоритм автоматизации преобразования КА в СП.

С целью разработки программного обеспечения, алгоритм подробно описан в виде блок схемы, чтобы пользователи могли попробовать разные реализации (рис.1).

С ++ – популярный в мире кроссплатформенный объектно-ориентированный язык программирования, который можно использовать для создания высокопроизводительных приложений, компьютерных программ. С++ является переносимым и дает четкую структуру программам, позволяет повторно использовать код. Учитывая эти возможности программная реализация алгоритма выполнена на С ++ [6].

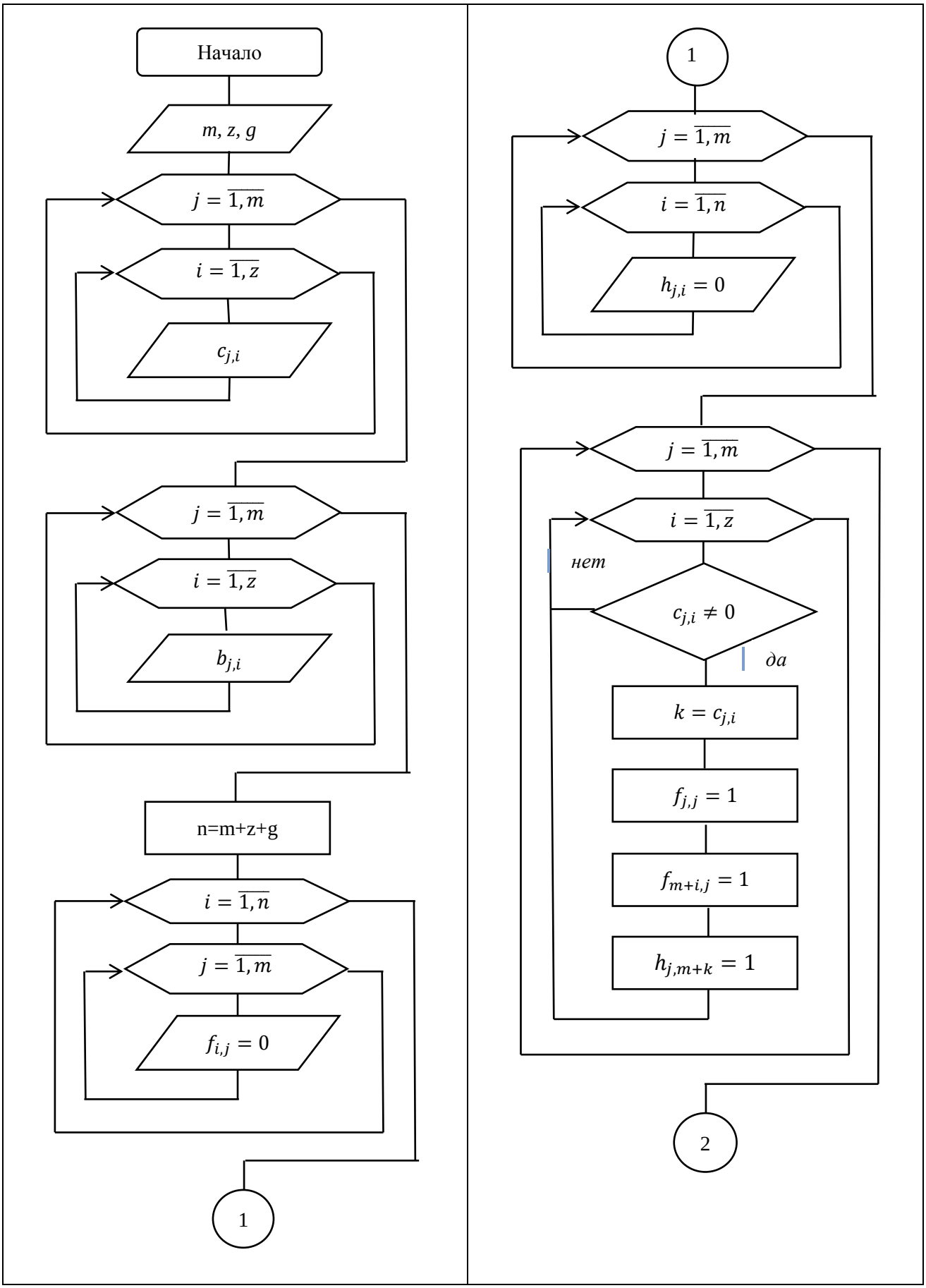

***Рис.**. Алгоритм автоматизации преобразования КА в СП*

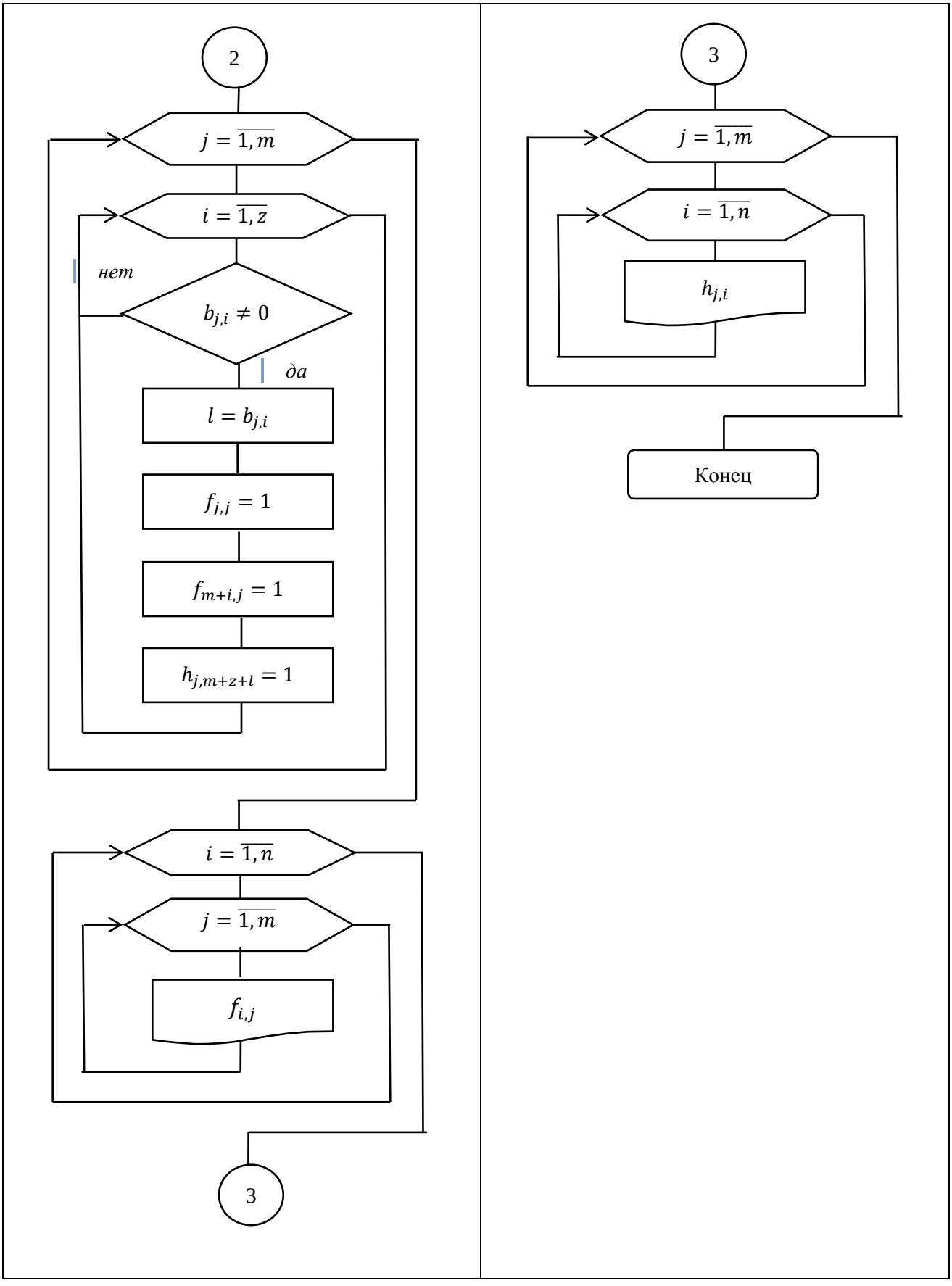

Рис. (продолжение). *Алгоритм автоматизации преобразования КА в СП*

Листинг программы:

```
#include  <iostream>
#include  <conio.h>
usingnamespacestd;
int main ()
{
// Объявление индексов
inti,j,k,l;
// Определение чисел входных сигналов,
//  внутренних состояний и выходных сигналов
intm,z,g;
cout<<"Vvoditechislovxodnixsiqnalov"<<"\n";
cin>>m;
cout<<"Vvoditechislovnutrennixsostoyaniy"<<"\n";
cin>>z;
cout<<"Vvoditechislovxodnixsiqnalov"<<"\n";
cin>>g;
int n=m+z+g;
// Определение матриц переходов и выходов
intC[m][z],B[m][z],F[n][m],H[m][n];
// Ввод элементов матрицы переходов
cout<<"Vvoditeelementovmatritsiperexodov"<<"\n";
for(j=1;j<=m;j++)
for(i=1;i<=z;i++){
cin>>C[j][i];}
// Ввод элементов матрицы выходов
cout<<"Vvoditeelementovmatritsivixodov"<<"\n";
for(j=1;j<=m;j++)
for(i=1;i<=z;i++){
cin>>B[j][i];}
// Вывод на печать элементов матрицы переходов
cout<<"Matritsaperexodov"<<"\n";
for(j=1;j<=m;j++){
for(i=1;i<=z;i++)
{cout<<C[j][i]<<" ";}
cout<<"\n\n";}
// Вывод на печать элементов матрицы выходов
cout<<"Matritsavixodov"<<"\n";
for(j=1;j<=m;j++){
for(i=1;i<=z;i++)
{cout<<B[j][i]<<" ";}
cout<<"\n";}
// Обнуление элементов матрицы входных инциденций сети Петри
for(i=1;i<=n;i++)
for(j=1;j<=m;j++){
F[i][j]==0;}
// Обнуление  элементов матрицы выходных инциденций сети Петри
for(j=1;j<=m;j++)
for(i=1;i<=n;i++){
```

```
   H[j][i]==0;}
// Вычисление элементов матрицы входных инциденций сети Петри
for(j=1;j<=m;j++)
for(i=1;i<=z;i++){
if(C[j][i]!=0){
  k=C[j][i];F[j][j]=1; F[m+i][j]=1; H[j][m+k]=1;}}
// Вычисление элементов матрицы выходных инциденций сети Петри
for(j=1;j<=m;j++)
for(i=1;i<=z;i++){
if(B[j][i]!=0){
l=B[j][i]; F[j][j]=1;F[m+i][j]=1; H[j][m+z+l]=1;}}
// Вывод на печать матрицы входных инциденций сети Петри
cout<<"Matritsavxodnixinsidensiyceti Petri"<<"\n";
for(i=1;i<=n;i++){
        for(j=1;j<=m;j++)
cout<<F[i][j]<<" ";
cout<<"\n\n";}
// Вывод на печать матрицы выходных инциденций сети Петри
cout<<"Matritsavixodnixinsidensiyceti Petri"<<"\n";
for(j=1;j<=m;j++){
for(i=1;i<=n;i++)
cout<<H[j][i]<<" ";
cout<<"\n\n";}
return 0;}
```

На примере рассматрена модель КА с внутренними состояниями $X = (x_1,\ x_2,\ x_3)$ (число внутренних состояний $z = 3$); входными сигналами $U = (u_1,\ u_2, u_3)$ (число входных сигналов $m = 3$); выходными сигналами $Y = (y_1,\ y_2, y_3)$: (число выходных сигналов $g = 3$). Функции перехода и выхода задаются следующими преобразованиями:

$$\begin{cases} \varphi\colon (x_1, u_3) \to x_2; \\ \varphi\colon (x_2, u_1) \to x_3; \\ \varphi\colon (x_3, u_2) \to x_1. \end{cases} \tag{3}$$

$$\begin{cases} \psi\colon (x_1, u_3) \to y_2; \\ \psi\colon (x_2, u_1) \to y_1; \\ \psi\colon (x_3, u_2) \to y_3. \end{cases} \tag{4}$$

Таблицы переходов и выходов автомата, заданного функциями перехода (3) и выхода (4) представлены в таб. 1.(а) и 1.(б).

Таблица .

Таблицы переходов (а) и выходов (б) автомата.

	x_1	x_2	x_3
u_1	ε	x_3	ε
u_2	ε	ε	x_1
u_3	x_2	ε	ε

а)

	x_1	x_2	x_3
u_1	ε	y_1	ε
u_2	ε	ε	y_3
u_3	y_2	ε	ε

б)

Соответственно (1) и (2)матрицы переходов и выходов имеют вид:

$$C(3,3) = \begin{vmatrix} 0 & 3 & 0 \\ 0 & 0 & 1 \\ 2 & 0 & 0 \end{vmatrix}, B(3,3) = \begin{vmatrix} 0 & 1 & 0 \\ 0 & 0 & 3 \\ 2 & 0 & 0 \end{vmatrix}.$$

В результате выполнения программного модуля соответственно матрицы входных и выходных инциденций СП $F(n,m) = \{f_{i,j}\}$ u $H(m,n) = \{h_{j,i}\}$, (где$i = \overline{1,n}; j = \overline{1,m}$; m $= 3$; z $= 2$; g $= 2$; $n = m + z + g = 9$.) формируются в нижеследующих видах:

$$F(9,3) = \begin{vmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{vmatrix}, H(3,9) = \begin{vmatrix} 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 \end{vmatrix}$$

Заключение. Разработан программный модуль преобразования конечного автомата (КА) в сети Петри (СП). Проведенные компьютерные эксперименты доказывают достоверность разработанного алгоритма и программного модуля. Предложенное программное обеспечение может быть полезным при анализе проектированных с применением КА дискретных систем, особенно при больших размерностях входных, выходных сигналов и внутренних состояний.

ЛИТЕРАТУРА

1. Брауэр В. Введение в теорию конечных автоматов. М.: Радио и связь, 1987, 392 с. https://www.twirpx.com/file/2576720/grant/
2. Колкер А.Б., Ливенец Д.А., Кошелева А.И., Жмудь В.А. Исследование вариантов создания интеллектуальных систем робототехники на базе одноплатных компьютеров и свободных операционных систем // Автоматика и программная инженерия. № 1. 2012, с. 84-98
3. Мухопад Ю.Ф. Анализ и синтез управляющих автоматов сложных технических систем. XII всероссийское совещание по проблемам управления ВСПУ-2014, 16-19 июня, М., 2014, с.7295-7306
4. Питерсон Дж. Теория сетей Петри и моделирование систем. М.: Мир, 1984, 264 с.
5. Ахмедов М.А., Гусейнзаде Ш.С., Насирова Е.А. Разработка алгоритма автоматизации преобразования конечного автомата в сеть Петри // Автоматизация. Современные технологии. т. 73. №3. 2019, с.108-112
6. C++ Tutorial. https://www.w3schools.com/cpp/

XÜLASƏ

SONLU AVTOMATIN PETRİ ŞƏBƏKƏSİNƏ ÇEVRİLMƏSİNİN AVTOMATLAŞDIRILMASININ PROQRAM TƏMİNATININ İŞLƏNMƏSİ

Hüseynzadə Ş.S.

Açar sözlər: *Petri şəbəkəsi, sonlu avtomat, insidentlik matrisi, keçid və çıxış cədvəlləri,* ***C++.***

Petri şəbəkəsinin (PŞ) sonlu avtomata (SA) çevrilməsinin avtomatlaşdırılması prosesinin kompüter reallaşdırılması təklif olunur. Proqram modulu, blok sxem şəklində ətraflı alqoritm əsasında krossplatformalı obyektyönümlü C ++ proqramlaşdırma dilində işlənmişdir. Proqram modulunun alqoritmi və listinqi təqdim olunur. İnteraktiv rejimdə SA-nın keçid və çıxış matrisləri daxil edilir, proqramın icrası nəticəsində əldə

edilən SA-nı simulyasiya edən PŞ-nin giriş və çıxış hadisələrinin insidentlik matrisləri hesablanaraq xaric edilir.

SUMMARY

AUTOMATION SOFTWARE DEVELOPMENT FOR TRANSFORMATION OF THE FINITE AUTOMATA INTO PETRI NET

Huseynzade Sh.S.

Key words: *Petri net, finite automata, incidence matrix, transition and output tables, C++.*

A computer implementation of the process of automation the transformation of the finite automata (FA) into Petri Net (PN) is proposed. The software module is developed by the cross-platform object-oriented programming language C ++, based on a detailed algorithm in the form of a block diagram. The algorithm and listing of the program module are presented. In the interactive mode, the matrices of transitions and outputs of the finite automata are entered, the matrices are displayed of input and output incidents of the PN which simulates the finite automata, obtained as a result of the execution of the program.

Daxilolma tarixi: İlkin variant 24.07.2020
Son variant 16.09.2020