

основе развития основной задачи управления производственной системой, который позволяет решать задачу прогнозирования, т. е. определяя направление изменения технологических и эксплуатационных требований (область поиска параметров) для выполнения заданных технических условий по показателям качества системы (целевые критерии). В результате дальнейшей работы решена основная задача управления для подсистем производственной системы (модуля, участка, цеха). Получены рациональные значения параметров подсистем, при которых удовлетворяются заданные технические условия на целевые критерии.

Библиографические ссылки

1. Сиразетдинов Т.К. Методы решения многокритериальных задач синтеза технических систем. М.: Машиностроение, 1988. 160 с.
2. Сиразетдинов Т.К. Динамическое моделирование экономических объектов. Казань: ФЭН, 1996. 224 с.
3. Иванов В.К. К решению основной задачи управления технологической подготовкой производства // Вестник Поволжского гос. техн. ун-та. Сер.: Материалы. Конструкции. Технологии. 2017. № 2. С. 81—86.
4. Робототехника и гибкие автоматизированные производства. Кн. 5. Моделирование робототехнических систем и гибких автоматизированных производств. М.: Высш. школа, 1986. 175 с.
5. Иванов В.К. К задаче аналитического проектирования элементов гибких производственных систем // Вестник КГТУ им. А.Н. Туполева. 2010. № 1. С. 93—96.
6. Иванов В.К. Автоматизация выбора рациональных параметров производственной системы // Автоматизация и современные технологии. 2010. № 9. С. 26—29.
7. Иванов В.К. Автоматизация моделирования возможностей гибких производственных систем // Вестник КГТУ им. А.Н. Туполева. 2010. № 4. С. 142—144.
8. Иванов В.К. Интегрированная система моделирования и управления возможностями предприятия // Автоматизация и современные технологии. 2012. № 7. С. 34—39.

Ссылка для цитирования

Иванов В.К. К решению основной задачи управления автоматизированным мелкосерийным машиностроительным производством // Автоматизация. Современные технологии. 2019. Т. 73. № 3. С. 104—108.

УДК 519.95

М.А. Ахмедов, д-р техн. наук, проф., Ш.С. Гусейнзаде, канд. техн. наук, доц., Е.А. Насирова
(Сумгаитский государственный университет, Азербайджанская Республика)

shahla.huseynzade@gmail.com

РАЗРАБОТКА АЛГОРИТМА АВТОМАТИЗАЦИИ ПРЕОБРАЗОВАНИЯ КОНЕЧНОГО АВТОМАТА В СЕТЬ ПЕТРИ

Рассмотрена задача автоматизации преобразования конечного автомата (КА) в сети Петри (СП). Разработан алгоритм преобразования таблиц переходов и выходов КА в матрицу входных и выходных инцидентов СП.

Продemonстрировано функционирование алгоритма на основе простого примера модели транспортной системы (ТС) гибкого автоматизированного участка (ГАУ). На основе матрицы инцидентностей, полученной в результате функционирования алгоритма, построена модель СП в виде графа, имитирующего КА.

Ключевые слова: сеть Петри; конечный автомат; преобразование конечного автомата; матрица инцидентностей; таблицы переходов и выходов; множество входных сигналов; выходной алфавит автомата; внутренние состояния автомата; функция переходов автомата; функция инцидентности позиций и переходов.

The transformation automation task of a finite automatic machine (FAM) into the Petri net (PN) is considered. The algorithm of the tables transformation for FAM transition and outlet into a matrix of the PN input and output incidents is developed. The functioning of the algorithm is demonstrated on a simple example of a transport system (TS) model for a flexible automated area (FAA). The PN model in the form of a graph simulating a FAM is constructed on the basis of the incidence matrix, obtained as a result of the algorithm functioning.

Keywords: Petri net; finite automatic machine; transformation of the finite automatic machine; incidence matrix; transition and outlet tables; many input signals; output alphabet automatic machine; internal states of the automatic machine; transition function of the automatic machine; incidence function for positions and transitions.

Введение. Мощным инструментом для изучения поведения технических и экономических объектов является аппарат имитационного моделирования сети Петри (СП). Для многих объектов и процессов традиционно используются автоматные представления. Описательная способность класса СП больше, чем у класса конечных автоматов [1].

Моделирование сложных динамических систем (ДС) можно рассматривать на нескольких уровнях, и СП могут моделировать каждый из этих уровней. На низком уровне модули системы часто описываются конечными автоматами (КА). Аппарат КА предназначен прежде всего для моделирования отдельных процессов. При моделировании же сложных ДС необходимо учитывать не только сами процессы, но и их взаимодействие. Частично эту проблему позволяют решать сети КА, т. е. схемы, которые получаются в результате присоединения выходов одних автоматов к входам других [2]. На высоком уровне более удобны модели, построенные на СП. Для анализа и формальной верификации системы при моделировании сложных динамических систем необходимо использовать достаточно мощный формальный аппарат как СП [3]. При этом часто возникает задача преобразования конечного автомата в СП.

СП лучше всего подходят для описания любых асинхронных систем, тогда как КА — для последовательных систем. Наглядность динамики и композиционные возможности СП выше, чем у КА [4].

Правила построения корректных схем и необходимость использования единого алфавита накладывают существенные ограничения на процесс разработки моделей при использовании сетей КА в качестве моделей сложных ДС [5].

Поэтому представляет интерес методика построения СП по автоматному представлению. Имеются некоторые общие подходы к решению этой задачи. Но конкретные решения задачи и их автоматизация в литературе не встречаются.

В работе представлена методика построения структурных элементов и матриц инцидентностей СП, которая моделирует КА, описываемый таблицами переходов и выходов.

Теоретический анализ задачи преобразования конечного автомата в СП. Классический КА связан с понятиями алгоритма и последовательной алгоритмической системы, для которых характерен последовательный способ

функционирования: система (автомат) последовательно переходит из одного состояния в другое в соответствии с заданной функцией перехода и осуществляет очередной (последовательный) шаг алгоритма. Обычный автомат имеет функции переходов и выходов, которые явно не зависят от времени [5].

Абстрактный автомат A задаётся совокупностью шести объектов [6]: $A = (X, U, Y, x_0, \varphi, \psi)$, где $U = (u_1, u_2, \dots, u_m)$ — конечное множество входных сигналов, называемое входным алфавитом автомата; $Y = (y_1, y_2, \dots, y_g)$ — конечное множество выходных сигналов, называемое выходным алфавитом автомата; $X = (x_1, x_2, \dots, x_z)$ — произвольное множество, называемое множеством внутренних состояний автомата; x_0 — элемент из множества X , называемый начальным состоянием автомата; $\varphi(x, u)$ и $\psi(x, u)$ — две функции, задающие однозначные отображения множества пар (x, u) , где $u \in U$ и $x \in X$, в множества X и Y . Функция $\varphi(x, u)$ называется функцией переходов автомата, а функция $\psi(x, u)$ — функцией выходов:

$$\varphi: (X \times U) \rightarrow X; \psi: (X \times U) \rightarrow Y.$$

КА может быть задан в виде графа, таблиц переходов и выходов и аналитическими способами. Функции переходов и выходов автомата $\varphi(x, u)$ и $\psi(x, u)$ могут быть представлены табличным способом.

Сеть Петри формально представляется как набор вида $N = (P, T, F, H, \mu^0)$, где $P = \{p_1, p_2, \dots, p_n\}$, $n > 0$ — конечное непустое множество позиций (иначе состояний или мест); $T = \{t_1, t_2, \dots, t_m\}$, $m > 0$ — конечное непустое множество переходов (событий); $F: P \times T \rightarrow \{0, 1, 2, \dots\}$ и $H: P \times T \rightarrow \{0, 1, 2, \dots\}$ — функции входных и выходных инцидентностей соответственно, а отображение $\mu^0: P \rightarrow \{0, 1, 2, \dots\}$ — начальная маркировка (разметка каждой позиции) [7].

Графическим изображением сети Петри является двудольный ориентированный граф с двумя типами вершин. Вершины $p \in P$ изображаются кружками, а вершины $t \in T$ — чёрточками (барьерами). Дуги соответствуют функциям инцидентности позиций и переходов [7].

Топология СП, имитирующей КА, определяется следующим образом: входному и выходному сигналам автомата ставятся в соответствие свои позиции, каждому состоянию автомата ставятся в соответствие позиция и переход. Обязательным условием, обеспечивающим детерминированность автомата,

является наличие в любой момент времени в позициях состояний только одного маркера, отмечающего нахождение моделируемого автомата в соответствующем состоянии [8]. Каждый переход связан входной дугой с позицией соответствующего состояния и двусторонними дугами — с входной и выходной позициями. Выходные дуги перехода связаны с позициями новых состояний автомата, отличных от прежнего.

Для КА $A = (X, U, Y, x_0, \varphi, \psi)$ определена СП $N = (P, T, F, H, \mu^0)$ следующим образом: $P = U \cup X \cup Y$, $T = \{t_x, x/x \in X \text{ и } u \in U\}$, $I(t_x, u) = \{x, u\}$, $O(t_x, u) = \{\varphi(x, u), \psi(x, u)\}$. Полученная СП является моделью конечного автомата [6].

На основе заданной топологии представляется определение структурных элементов СП — переходов и позиций по следующим правилам: для каждой пары (состояние и входной символ) определяется переход; объединение множеств входных сигналов $U = \{u_1, u_2, \dots, u_m\}$, выходных сигналов $Y = \{y_1, y_2, \dots, y_g\}$, внутренних состояний $X = \{x_1, x_2, \dots, x_z\}$ автомата составляет множество позиций СП. Переходы и позиции СП определяются как $T = \{t_1, t_2, \dots, t_m\}$, $P = \{p_1, p_2, \dots, p_n\}$, где $n = m + z + g$.

Предлагается методика построения СП по автоматному представлению, основанная на преобразовании таблиц переходов и выходов КА в матрицу инцидентностей СП.

Алгоритм преобразования таблиц переходов и выходов КА в матрицы входных и выходных инцидентностей СП.

Начало алгоритма.

Шаг 1. Создание матрицы переходов $C = \{c_{j,i}\}$ на основе таблицы переходов:

$$c_{j,i} = \begin{cases} k, & \text{если } \varphi:(x_i, u_j) \rightarrow x_k; \\ 0, & \text{если } \varphi:(x_i, u_j) \rightarrow \varepsilon, \end{cases}$$

где $i = \overline{1, z}$; $k = \overline{1, z}$; $j = \overline{1, m}$; ε — пустой элемент.

Шаг 2. Создание матрицы выходов $B = \{b_{j,i}\}$ на основе таблицы выходов:

$$b_{j,i} = \begin{cases} k, & \text{если } \psi:(x_i, u_j) \rightarrow y_k; \\ 0, & \text{если } \psi:(x_i, u_j) \rightarrow \varepsilon, \end{cases}$$

где $i = \overline{1, z}$; $k = \overline{1, g}$; $j = \overline{1, m}$.

Шаг 3. Определение матрицы входных инцидентностей СП $F = \{f_{i,j}\}$, где $i = \overline{1, n}$; $j = \overline{1, m}$; $n = m + z + g$ (n — число позиций, m — число переходов). Элемент f_{ij} равен числу дуг от i -й позиции к j -му переходу:

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

Шаг 4. Определение матрицы выходных инцидентностей СП $H = \{h_{j,i}\}$, где $i = \overline{1, n}$; $j = \overline{1, m}$. Элемент $h_{j,i}$ равен числу дуг от j -го перехода к i -й позиции:

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

Шаг 5. Формирование элементов матрицы входных инцидентностей СП $F = \{f_{i,j}\}$:

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

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

Шаг 6. Формирование элементов матрицы выходных инцидентностей СП $H = \{h_{j,i}\}$:

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

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

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

Шаг 7. Формирование элементов матрицы инцидентностей СП $D = \{d_{j,i}\}$:

так как $D = F - H^T$, то $d_{i,j} = f_{i,j} - h_{j,i}$, где $j = \overline{1, m}$; $i = \overline{1, n}$.

Конец алгоритма.

Для проверки достоверности продемонстрировано функционирование алгоритма на основе простого примера.

Демонстрация функционирования алгоритма на основе примера. В работе представлено решение этой задачи на конкретном примере. Здесь рассматривается модель функционирования транспортной системы (ТС) гибкого автоматизированного участка (ГАУ). ТС может находиться в одном из двух состояний $X = (x_1, x_2)$: x_1 — отключено, x_2 — включено (число внутренних состояний $z = 2$). Переход от одного состояния к другому осуществляется управляющими воздействиями $U = (u_1, u_2)$: u_1 — отключение, u_2 — включение ТС (число входных сигналов $m = 2$). Функции перехода и выхода задаются следующими преобразованиями:

$$\varphi:(x_2, u_1) \rightarrow x_1; \quad (1)$$

$$\varphi:(x_1, u_2) \rightarrow x_2,$$

$$\psi:(x_2, u_1) \rightarrow y_1; \quad (2)$$

$$\psi:(x_1, u_2) \rightarrow y_2.$$

Внешние состояния ТС определяются выходным алфавитом $Y = (y_1, y_2)$: y_1 — ТС находится в отключенном состоянии, y_2 — ТС находится во включенном состоянии (число выходных сигналов $g = 2$). Следовательно, для управления ТС необходимо определить внутренние состояния X на основе информации U с соответствующих датчиков и выработать управляющие сигналы Y в соответствии с функцией перехода. Графовое представление этого автомата задано на рис. 1. Таблицы переходов и выходов автомата, заданного функциями перехода (1) и выхода (2), представлены на рис. 2, а и б.

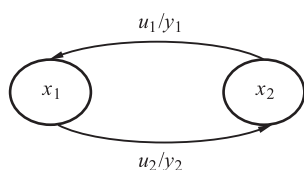


Рис. 1. Граф — представление автомата, заданного функциями перехода (1) и выхода (2)

	x_1	x_2
u_1		x_1
u_2	x_2	

а)

	x_1	x_2
u_1		y_1
u_2	y_2	

б)

Рис. 2. Таблицы переходов (а) и выходов (б) автомата, заданного функциями перехода (1) и выхода (2)

Теперь для определения переходов из одного состояния в другое можно ввести переходы сети следующим образом. Для каждой пары (состояние и входной символ) определяем переход, входные позиции которого соответствуют состоянию и входному символу, а выходные — следующему состоянию и выходу.

Таким образом, определяются переходы $T = \{t_1, t_2\}$ СП (число переходов $m = 2$).

Позиции СП определяются по следующему правилу (число позиций $n = m + z + g = 2 + 2 + 2 = 6$):

$$P = U \cup X \cup Y = \{u_1, u_2, x_1, x_2, y_1, y_2\} = \{p_1, p_2, p_3, p_4, p_5, p_6\}.$$

Начало выполнения алгоритма.

Шаг 1. Создание матрицы переходов $C(m, z) = C(2, 2) = \{c_{j,i}\}$, где $i = \overline{1,2}$; $k = \overline{1,2}$; $j = \overline{1,2}$; на основе таблицы переходов (см. рис. 2, а):

$$\begin{aligned} c_{1,1} &= 0, \text{ так как } \varphi:(x_1, u_1) \rightarrow \varepsilon; \\ c_{1,2} &= 1, \text{ так как } \varphi:(x_2, u_1) \rightarrow x_1; \\ c_{2,1} &= 2, \text{ так как } \varphi:(x_1, u_2) \rightarrow x_2; \\ c_{2,2} &= 0, \text{ так как } \varphi:(x_2, u_2) \rightarrow \varepsilon. \end{aligned}$$

В результате получается матрица переходов

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

Шаг 2. Создание матрицы выходов $B(m, z) = B(2, 2) = \{b_{j,i}\}$, где $i = \overline{1,2}$; $k = \overline{1,2}$; $j = \overline{1,2}$; на основе таблицы выходов (см. рис. 2, б):

$$\begin{aligned} b_{1,1} &= 0, \text{ так как } \psi:(x_1, u_1) \rightarrow \varepsilon; \\ b_{1,2} &= 1, \text{ так как } \psi:(x_2, u_1) \rightarrow x_1; \\ b_{2,1} &= 2, \text{ так как } \psi:(x_1, u_2) \rightarrow x_2; \\ b_{2,2} &= 0, \text{ так как } \psi:(x_2, u_2) \rightarrow \varepsilon. \end{aligned}$$

В результате получается матрица выходов

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

Шаг 3. Определение матрицы входных инцидентий СП $F(n, m) = F(6, 2) = \{f_{i,j}\}$, где $i = \overline{1,6}$; $j = \overline{1,2}$; $n = m + z + g = 6$ (n — число позиций, m — число переходов).

Шаг 4. Определение матрицы выходных инцидентий СП $H(m, n) = H(2, 6) = \{h_{j,i}\}$, где $i = \overline{1,6}$; $j = \overline{1,2}$.

Шаг 5. Формирование элементов матрицы входных инцидентий СП $F(6, 2) = \{f_{i,j}\}$:

обнуление всех элементов матрицы $f_{i,j} = 0$, где $i = \overline{1,6}$; $j = \overline{1,2}$;

если $c_{j,i} \neq 0$, тогда $f_{j,j} = 1$ и $f_{2+i,j} = 1$, где $i = \overline{1,2}$; $j = \overline{1,2}$:

так как $c_{1,2} \neq 0$, получается $f_{j,j} = f_{1,1} = 1$, $f_{2+i,j} = f_{2+2,1} = f_{4,1} = 1$;

так как $c_{2,1} \neq 0$, получается $f_{j,j} = f_{2,2} = 1$, $f_{2+i,j} = f_{2+1,2} = f_{3,2} = 1$;

$$F(6, 2) = \begin{vmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 1 \\ 1 & 0 \\ 0 & 0 \\ 0 & 0 \end{vmatrix}.$$

Шаг 6. Формирование элементов матрицы выходных инцидентий СП $H(2, 6) = \{h_{j,i}\}$:

обнуление всех элементов матрицы $h_{j,i} = 0$, где $j = \overline{1,2}$; $i = \overline{1,6}$;

если $c_{j,i} \neq 0$, тогда $h_{j,m+l} = h_{j,2+l} = 1$, где $j = \overline{1,2}; i = \overline{1,2}; l = c_{j,i}$;

так как $c_{1,2} \neq 0$, получается $h_{j,m+l} = h_{1,2+1} = h_{1,3} = 1$ ($l = c_{j,i} = c_{1,2} = 1$);

так как $c_{2,1} \neq 0$, получается $h_{j,m+l} = h_{2,2+2} = h_{2,4} = 1$ ($l = c_{j,i} = c_{2,1} = 1$);

если $b_{j,i} \neq 0$, тогда $h_{j,m+z+l} = h_{j,2+2+l} = 1$, где $j = \overline{1,2}; i = \overline{1,2}; l = b_{j,i}$;

так как $b_{1,2} \neq 0$, получается $h_{j,m+z+l} = h_{1,2+2+1} = h_{1,5} = 1$ ($l = b_{j,i} = b_{1,2} = 1$);

так как $b_{2,1} \neq 0$, получается $h_{j,m+z+l} = h_{2,2+2+2} = h_{2,6} = 1$ ($l = b_{j,i} = b_{2,1} = 2$);

$$H(2,6) = \begin{vmatrix} 0 & 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 \end{vmatrix}.$$

Шаг 7. Формирование элементов матрицы инцидентий СП $D(n, m) = D(6, 2) = \{d_{i,j}\}$:

так как $D = F - H^T$, то $d_{i,j} = f_{i,j} - h_{j,i}$, где $j = \overline{1,2}; i = \overline{1,6}$;

$$D(6,2) = F(6,2) - H^T(2,6) =$$

$$= \begin{vmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 1 \\ 1 & 0 \\ 0 & 0 \\ 0 & 0 \end{vmatrix} - \begin{vmatrix} 0 & 0 \\ 0 & 0 \\ 1 & 0 \\ 0 & 1 \\ 1 & 0 \\ 0 & 1 \end{vmatrix} = \begin{vmatrix} 1 & 0 \\ 0 & 1 \\ -1 & 1 \\ 1 & -1 \\ -1 & 0 \\ 0 & -1 \end{vmatrix}.$$

Конец алгоритма.

Используя полученные результаты при функционировании алгоритма, можно построить граф СП, имитирующий КА, заданного

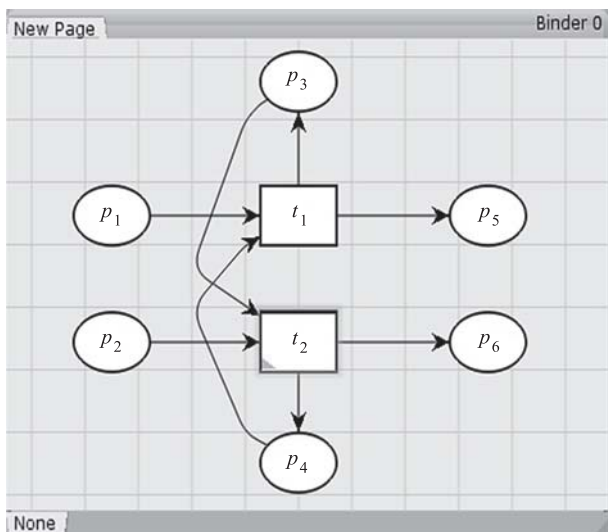


Рис. 3. Граф СП, имитирующий КА, заданного функциями перехода (1) и выхода (2)

функциями перехода (1) и выхода (2) (рис. 3). Граф построен в системе CPN Tools [9].

На основе алгоритма разработано программное обеспечение, произведены компьютерные эксперименты и получены достоверные результаты.

Заключение. Разработан алгоритм преобразования таблиц переходов и выходов КА в матрицу входных и выходных инцидентий СП. Продемонстрировано функционирование алгоритма на основе примера. По результатам функционирования алгоритма построена модель СП в виде графа, имитирующего КА. Разработанная методика может явно и строго определить преобразование КА в СП. Предложенный алгоритм может быть особенно полезен при больших размерностях множеств входных, выходных сигналов и внутренних состояний автомата.

Библиографические ссылки

1. Jensen K., Kristensen L.M. Coloured Petri Nets: modelling and validation of concurrent systems // Springer. 2009. № 11. Р. 384.
2. Кузнецов О.П., Адельсон-Вельский Г.М. Дискретная математика для инженера. 2-е изд., перераб. и доп. М.: Энергоатомиздат, 1988. 480 с.
3. Ломазова И.А. Вложенные сети Петри: моделирование и анализ распределенных систем с объектной структурой. М.: Научный мир, 2004. 208 с.
4. Мальков М.В., Малыгина С.Н. Сети Петри и моделирование // Труды Кольского научного центра РАН. 2010. № 3. С. 35—40.
5. Королёв Ю.И. Методы и программные средства моделирования сложных динамических систем на основе темпоральной модификации раскрашенных сетей Петри: дис. ... канд. техн. наук. М., 2015. 146 с.
6. Захаров Н.Г., Рогов В.Н. Синтез цифровых автоматов. Ульяновск: УлГТУ, 2003. 135 с.
7. Питерсон Дж. Теория сетей Петри и моделирование систем. М.: Мир, 1984. 264 с.
8. Механов В.Б., Кизилев Е.А., Коннов Н.Н. Моделирование конечного автомата в системе моделирования CPN Tools // Сб. статей X Междунар. науч.-метод. конф. «Проблемы информатики в образовании, управлении, экономики и технике». Пенза: ПДЗ, 2010. С. 283—287.
9. Aalst W., Stahl C. Modeling Business Processes — A Petri Net — Oriented Approach. Cambridge MA: MIT Press, 2011. 400 p.

Ссылка для цитирования

Ахмедов М.А., Гусейнзаде Ш.С., Насирова Е.А. Разработка алгоритма автоматизации преобразования конечного автомата в сеть Петри // Автоматизация. Современные технологии. 2019. Т. 73. № 3. С. 108—112.