УДК 519.6

В.В. Кожевников

МЕТОД МАТЕМАТИЧЕСКОГО МОДЕЛИРОВАНИЯ КОГНИТИВНЫХ ЦИФРОВЫХ АВТОМАТОВ[1]

Кожевников Валерий Владимирович, *кандидат технических наук, окончил Пушкинское высшее командное училище радиоэлектроники, старший научный сотрудник Научно-исследовательского технологического института Ульяновского государственного университета. Имеет публикации в области теории проектирования микроэлектронных систем. [e-mail: vvk28061955@mail.ru].*

Аннотация

Предлагается подход к решению проблемы математического моделирования когнитивных цифровых автоматов (КЦА). На первый план выдвигается задача формализации понятия когнитивности математической модели КЦА. Когнитивность математической модели, соответственно, определяется возможностью обучения и генерации решений, не предусмотренных в процессе обучения.

Особенность КЦА заключается в том, что в качестве структурной схемы автомата служит описание структуры нейронной сети (НС), а в качестве модели нейрона используется логическая функция «НЕ-И-ИЛИ». В случае образования обратных связей с выхода на входы нейронов модель нейрона представляет собой двоичный триггер с логической функцией «НЕ-И-ИЛИ» на входе. В качестве инструмента построения математической модели КЦА предлагается использовать математический аппарат сетей Петри (СП): маркированные графы, ингибиторные СП и СП с программируемой логикой. Математическая модель строится на базе представления КЦА в виде уравнения состояний СП из класса уравнений Мурата (матричных уравнений) или системы линейных алгебраических уравнений.

Задача формализации понятия когнитивности (познания) решается в результате синтеза логики (обучения) исходной структурной схемы КЦА или формирования формулы (сетевого алгоритма) КЦА. При этом возможность формирования формулы (сетевого алгоритма) КЦА зависит от критической массы (качества) обучающих наборов и алгоритмов обучения. Отсюда особое значение приобретает задача генерации минимального множества обучающих наборов для заданной или экспериментально определяемой функции КЦА. Прогнозирование, или генерация решений, в свою очередь, выполняется на основе полученной в процессе обучения математической модели КЦА.

Ключевые слова: интеллектуальная система управления, когнитивный цифровой автомат, искусственный интеллект, нейронные сети, машинное обучение, познание, сети Петри, уравнение состояний, математическое моделирование, синтез, генерация, анализ, логика.

doi: 10.35752/1991-2927-2019-2-56-101-112

THE METHOD OF MATHEMATICAL MODELING OF COGNITIVE DIGITAL AUTOMATA

Valerii Vladimirovich Kozhevnikov, *Candidate of Science in Engineering; graduated from the Pushkin Higher Command School of Radioelectronics; Senior Researcher at the Scientific Research Technological Institute of Ulyanovsk State University; an author of articles in the field of microelectronic system design theory. e-mail: vvk28061955@mail.ru.*

Abstract

An approach to solving the problem of mathematical modeling of cognitive digital automata (CDA) is proposed. The task of formalizing the concept of the cognitive nature of the CDA mathematical model comes to the fore. The cognitiveness (cognition) of the mathematical model is determined by the possibility of learning and generating solutions that are not provided for in the learning process.

A special feature of CDA is that the description of the neural network (NN) structure is used as a structural circuit of the automata, and the logical function "NOT-AND-OR" is used as the model of the neuron. In the case of the feedbacks formation from the output to the inputs of the neurons, the model of the neuron is a binary trigger with a logical function "NOT-AND-OR" at the input. As a tool for constructing a mathematical model of CDA, a mathematical apparatus of Petri nets (PNs) is proposed: marked graphs, inhibitory PNs and PNs with programmable logic. The mathematical model is built

1 Исследование выполнено при финансовой поддержке РФФИ и Правительства Ульяновской области в рамках научного проекта № 18-47-732015 р_мк.

on the basis of the representation of the CDA in the form of the state equation of the PNs from the class of Murat equations (matrix equations) or a system of linear algebraic equations.

The task of formalizing the concept of cognitiveness (cognition) is solved as a result of the logic synthesis (learning) of the initial structural circuit of CDA or the formation of the formula (network algorithm) of CDA. At the same time, the possibility of forming a formula (network algorithm) of CDA depends on the critical mass (quality) of training sets and training algorithms. Hence, the task of generating the minimum set of training sets for a given CDA function or experimentally determined function takes on particular importance. Forecasting or generation of solutions, in turn, is performed on the basis of the mathematical model of CDA obtained in the learning process.

Key words: intellectual control system, cognitive digital automata, artificial intelligence, neural networks, machine learning, cognition, Petri nets, equation of states, mathematical modeling, synthesis, generation, analysis, logic.

Введение

Актуальность проблемы математического моделирования когнитивных цифровых автоматов (КЦА) определяется необходимостью создания новых цифровых технологий, обеспечивающих возможность перехода к интеллектуальным (когнитивным) системам управления сложными объектами [1–3]. При этом на первый план выдвигается задача формализации понятия когнитивности математической модели КЦА. Когнитивность математической модели, соответственно, определяется возможностью обучения и генерации решений, не предусмотренных в процессе обучения. Предлагаемый подход к решению проблемы математического моделирования КЦА [4], по сути, является развитием теории цифровых автоматов и, соответственно, строится на основе методов математического моделирования обычных цифровых автоматов [5, 6].

Особенность КЦА заключается в том, что в качестве структурной схемы автомата служит описание структуры нейронной сети (НС), а в качестве компонентов структурной схемы или модели нейрона используется логическая функция «НЕ-И-ИЛИ». При этом логика компонентов структурной схемы автомата и структурных связей между компонентами изначально не определена и формируется в процессе обучения на основе модели синаптической памяти нейронов. Рассматриваются два вида памяти – постоянная, или синаптическая память, и оперативная, возникающая в результате образования обратных связей с выхода на входы нейронов. В случае образования обратных связей модель нейрона представляет собой двоичный триггер с логической функцией «НЕ-И-ИЛИ» на входе, который способен находиться в двух состояниях – «0» или «1».

Исходная структура автомата представляется в виде универсальной матрицы, где связи между компонентами построены по принципу «все со всеми». Возможны также различные многоуровневые конфигурации исходной структуры, где связи между компонентами построены по принципу «все со всеми» только между различными уровнями исходной структуры или по принципу сверточных НС.

Формирование конфигурации исходной структуры автомата осуществляется в результате кластерного анализа множества обучающих наборов. На основе результатов кластерного анализа или классификации множества обучающих наборов можно определить количество входов и выходов НС, количество слоев НС, количество нейронов в каждом наборе, исходную структуру связей между нейронами. Регрессионный анализ данных обучающих наборов выполняется в процессе обучения или синтеза логики исходной структуры автомата.

В качестве инструмента построения математической модели КЦА предлагается математический аппарат сетей Петри (СП) [7]: маркированные графы, ингибиторные СП [8] и СП с программируемой логикой (СППЛ) [9]. В отличие от ингибиторных СП логика запуска переходов СППЛ заранее не задана, и любая входная дуга перехода может быть ингибиторной, а переход может быть запрограммирован на выполнение любой логической функции. При этом логика компонентов исходной структуры автомата в матрице инцидентности задается неявно. Выбор (разработка) математического аппарата, принципов, методов и подходов (основ) математического моделирования КЦА обоснован необходимостью обеспечения возможности модельного представления последовательностных логических схем автомата, исчисления инвариантов сетевой модели, реляционного исчисления инвариантов сетевой модели, синтеза логики, анализа достижимости устойчивых состояний, генерации достижимых устойчивых состояний, моделирования динамики асинхронных и параллельных процессов функционирования.

Математическая модель строится на базе представления КЦА в виде уравнения состояний СП из класса уравнений Мурата (матричных уравнений) [10] или системы линейных алгебраических уравнений (СЛАУ). Возможность представления КЦА в виде уравнения состояний СП (матричных уравнений) или СЛАУ достигается в результате неявного определения логики в сетевой модели. При этом понятие неявного определения логики не имеет ничего общего с «нечеткой логикой» СП или НС. Построение сетевой модели КЦА осуществляется исходя из принципа сохранения потока однородной информации. Свойство однородности сетевой модели обеспечивает сохранение свойств ингибиторных СП в сетевой модели для моделирования логики и одновременно служит в качестве критерия достижимости. Математическая модель КЦА обеспечивает необходимую мощность моделирования и в то же время увеличивает мощность разрешения, необходимую для решения задач синтеза (обучения) и генерации решений.

Задача формализации понятия когнитивности (познания) решается в результате синтеза логики (обучения) исходной структурной схемы КЦА или формирования формулы (сетевого алгоритма) КЦА. При этом возможность формирования формулы (сетевого алгоритма) КЦА зависит от критической массы (качества) обучающих наборов и алгоритмов обучения. Отсюда особое значение приобретает задача генерации минимального множества обучающих наборов для заданной или экспериментально определяемой функции КЦА. Прогнозирование, или генерация решений, в свою очередь, выполняется на основе полученной в процессе обучения математической модели КЦА.

1 Построение математической модели КЦА

Исходной информацией для построения математической модели служит описание структурной схемы КЦА. Структурная схема КЦА представляется в виде маркированного графа (рис. 1.) путем интерпретации входов и выходов схемы и структурных компонентов позициями маркированного графа, а самих компонентов и линий соединений составными и простыми переходами соответственно. Множество входов и выходов структурной схемы интерпретируется как множество входных и выходных позиций сети. Наличие информации интерпретируется как фишка в позиции сети. На логическом уровне представления фишка в позиции сети интерпретируется как логическая единица, а ее отсутствие – как логический ноль. Перемещение информации интерпретируется как движение фишек в сети.

На рисунке 1 приведен маркированный граф двухразрядного сумматора. Формирование структурной схемы выполнено на основе результатов кластерного анализа (классификации) множества обучающих наборов или таблицы истинности двухразрядного сумматора (табл.). В первой строке таблицы приведены условные обозначения входов и выходов двухразрядного сумматора. Во второй строке – их обозначения на графе.

Таблица истинности двухразрядного сумматора

№	a_2	b_2	a_1	b_1	f_2	f_1	c_2
	P_4	P_3	P_2	P_1	P_{20}	P_{19}	
1	0	0	0	0	0	0	0
2	0	0	0	1	0	1	0
3	0	0	1	0	0	1	0
4	0	0	1	1	1	0	0
5	0	1	0	0	1	0	0
6	0	1	0	1	1	1	0
7	0	1	1	0	1	1	0
8	0	1	1	1	0	0	1
9	1	0	0	0	1	0	0
10	1	0	0	1	1	1	0
11	1	0	1	0	1	1	0
12	1	0	1	1	0	0	1
13	1	1	0	0	0	0	1
14	1	1	0	1	0	1	1
15	1	1	1	0	0	1	1
16	1	1	1	1	1	0	1

В качестве минимального множества обучающих наборов (выделены серым цветом) отобраны те

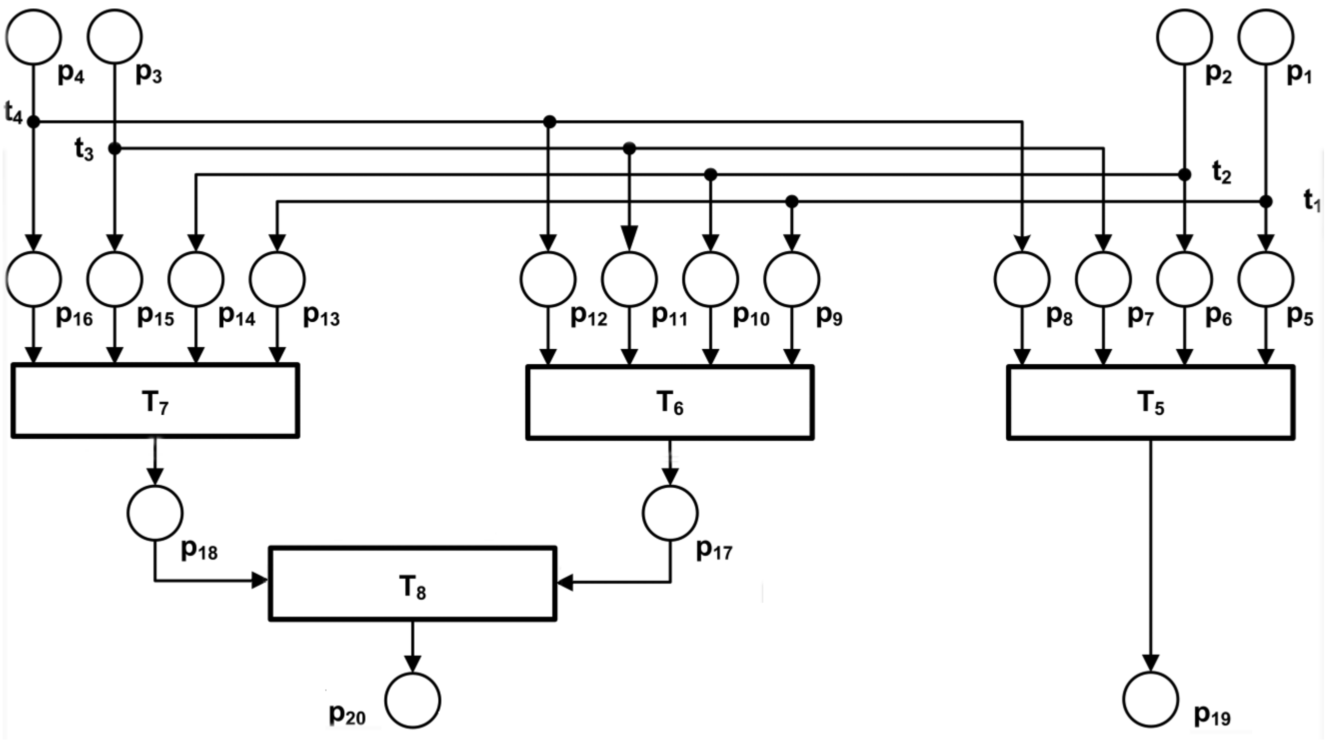

Рис. 1. Маркированный граф двухразрядного сумматора

наборы из таблицы истинности, которые дают единицу только в одном выходном разряде сумматора с учетом переноса в старший (третий) разряд. Таким образом, для каждого выходного разряда сумматора сформирован соответствующий класс обучающих наборов. Далее для каждого класса обучающих наборов отобраны наборы с одной единицей на входах сумматора, с двумя единицами и т. д. По данному принципу сформированы соответствующие подклассы обучающих наборов.

При формировании исходной структуры сумматора множество подклассов составляет первый слой структурной схемы, где для каждого подкласса отводится один компонент. Множество классов составляет второй слой структурной схемы, где для каждого класса также отводится один компонент. В результате исходная структура сумматора состоит из двух слоев. Количество компонентов первого слоя равно количеству подклассов. Количество компонентов второго слоя равно количеству классов. В случае, когда класс состоит из одного подкласса, формируется только один компонент в первом или во втором слоях.

Структура связей между входами структурной схемы и входами компонентов первого слоя, выходами компонентов первого слоя и входами компонентов второго слоя сформирована по принципу «все со всеми».

Графическая форма представления структурной схемы КЦА позволяет перейти от описания структурной схемы к ее математическому представлению в виде матрицы инцидентности: $\mathbf{A} = \mathbf{A}^{+} - \mathbf{A}^{-}$. Матрица инцидентности для двухразрядного сумматора приведена ниже:

$$\mathbf{A} = \begin{array}{cc} & \\ b_1 & p_1 \\ a_1 & p_2 \\ b_2 & p_3 \\ a_2 & p_4 \\ & p_5 \\ & p_6 \\ & p_7 \\ & p_8 \\ & p_9 \\ & p_{10} \\ & p_{11} \\ & p_{12} \\ & p_{13} \\ & p_{14} \\ & p_{15} \\ & p_{16} \\ & p_{17} \\ & p_{18} \\ f_1 & p_{19} \\ f_2 & p_{20} \end{array} \begin{array}{c} \begin{array}{cccccccc} t_1 & t_2 & t_3 & t_4 & T_5 & T_6 & T_7 & T_8 \end{array} \\ \left[\begin{array}{cccccccc} -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & -1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & -1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & -1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & -1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & -1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & -1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & -1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & -1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & -1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & -1 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{array}\right] \end{array}.$$

Представление структурной схемы КЦА в виде маркированного графа или матрицы инцидентности позволяет задать модель структурной схемы КЦА статически. Динамику в модель вносит движение фишек, регулируемое правилами запуска переходов и смены разметки сети.

Единство статического и динамического аспектов достигается в результате представления сетевой модели структурной схемы КЦА в виде фундаментального уравнения состояний СП из класса уравнений Мурата [10]:

$$\Delta\boldsymbol{\mu} = \mathbf{A} \cdot \boldsymbol{\tau}, \tag{1}$$

где $\Delta\boldsymbol{\mu} = \boldsymbol{\mu} - \boldsymbol{\mu}_0$, $\boldsymbol{\mu}_0$ – вектор начальной разметки сети, $\boldsymbol{\mu}$ – вектор конечной разметки сети, $\boldsymbol{\tau}$ – вектор покрытия переходов сети (вектор счёта последовательности срабатываний переходов или отображение Париха [7] для последовательности срабатываний переходов), который определяет только состав и не определяет последовательность срабатываний переходов, при $\tau_i = \{0,1\}$. Вектор $\Delta\boldsymbol{\mu}$ задается на множестве позиций сети $\mathbf{P}$. Вектор $\boldsymbol{\tau}$ задается на множестве переходов сети $\mathbf{T}$. Множество векторов $\Delta\boldsymbol{\mu}$ образует множество $\Delta\mathbf{M}$, где $\Delta\boldsymbol{\mu} \in \Delta\mathbf{M}$. Множество векторов покрытия переходов $\boldsymbol{\tau}$ образует покрытие сети $\mathbf{S}$, где $\boldsymbol{\tau} \in \mathbf{S}$.

Для маркированных графов структурной схемы автомата $\boldsymbol{\tau} = \mathbf{1}$. Для сетевой модели логической схемы автомата $\boldsymbol{\tau} \le \mathbf{1}$. Ограничение вектора покрытия переходов сети $\boldsymbol{\tau}$ обеспечивает безопасность (ограниченность) сетевой модели, при $\Delta\boldsymbol{\mu}_j = \{-1,0,1\}$.

Правила срабатываний переходов СП могут быть заданы следующими отношениями:

$$\boldsymbol{\mu}_k = \boldsymbol{\mu}_{k-1} + \mathbf{A} \cdot \mathbf{u}_k \tag{2}$$

и

$$\boldsymbol{\mu}_{k-1} + \mathbf{A} \cdot \mathbf{u}_k \ge \mathbf{0}, \tag{3}$$

где $\mathbf{u}_k$ – вектор запуска переходов сети, $\mathbf{u}_k \in \boldsymbol{\tau}$,

$\boldsymbol{\mu}_k$ – вектор текущей разметки сети, интерпретируется как вектор текущего состояния автомата.

Уравнение (2) определяет правило смены разметки сети, условие (3) – правило запуска переходов сети соответственно. Все переходы, каждый и все одновременно, удовлетворяющие условию (3), составляют вектор $\mathbf{u}_k$. Для каждой последовательности переходов $\Delta\boldsymbol{\mu} = \sum_{k=1}^{n} \Delta\boldsymbol{\mu}_k, \boldsymbol{\tau} = \sum_{k=1}^{n} \mathbf{u}_k$, где $\Delta\boldsymbol{\mu}_k = \boldsymbol{\mu}_k - \boldsymbol{\mu}_{k-1}$.

Множество достижимых разметок сети определяет пространство состояний или мощность моделирования сети, которая для СП ограничена собственно правилами Петри. Мощность разрешения сети зависит от возможностей ее анализа. Повышение моделирующей мощности сети может быть достигнуто путем определения дополнительных правил срабатываний переходов сети, отличных от правил Петри, применение которых приводит к неопределенной логике срабатывания переходов и соответственно к неопределенной логике функционирования сети, что выражается в нарушении уравнения (1).

Построение комплексной математической модели КЦА выполняется на базе фундаментального уравнения состояний СП (1). Если уравнение состояний СП (1) представить как $\boldsymbol{\mu} = \boldsymbol{\mu}_0 + \mathbf{A} \cdot \boldsymbol{\tau}$ и, соответственно, $\boldsymbol{\mu}_0 + \mathbf{A} \cdot \boldsymbol{\tau} - \boldsymbol{\mu} = \mathbf{0}$, то все возможные расширения СП могут быть заданы следующими отношениями [6]:

$$\boldsymbol{\mu}_0 + \mathbf{A} \cdot \boldsymbol{\tau} - \boldsymbol{\mu} \leq \mathbf{0} \tag{4}$$

и

$$\boldsymbol{\mu}_0 + \mathbf{A} \cdot \boldsymbol{\tau} - \boldsymbol{\mu} \geq \mathbf{0}. \tag{5}$$

При этом неопределенная логика запуска переходов (входная логика переходов) выражается отношением (4). Неопределенная логика смены разметки сети (выходная логика переходов) – отношением (5). Совмещение расширений СП (4) и (5) в рамках одной сети допускается только в случае введения дополнительных ограничений при построении исходной структуры сети.

Отношения (4) и (5) могут быть использованы для определения ингибиторных СП и их расширений. Введение специальных обозначений для ингибиторных дуг в матрице инцидентности $\mathbf{A}$ уравнения (1): $a_{ij} = -1 \rightarrow a_{ij} = -\alpha$ и $a_{ij} = 1 \rightarrow a_{ij} = \alpha$, где $\alpha \neq 0$, эквивалентно отношениям (4) и (5). В случае неявного определения ингибиторных дуг (равными нулю) в матрице инцидентности, то есть при $\alpha = 0$, ингибиторная СП может быть представлена в виде фундаментального уравнения СП (1). Для идентификации ингибиторных дуг в матрице инцидентности выполняется программирование логики срабатывания переходов путем построения (генерации) минимального порождающего множества решений уравнения состояний (1). В общем случае программирование логики срабатывания каждого перехода заключается в определении активности смежных переходов путем частичного определения вектора $\boldsymbol{\tau}$. Вектор покрытия сети $\boldsymbol{\tau}$ в данном случае определяет не только состав, но и логику запуска переходов. Покрытие сети S определяет логику функционирования сети.

Вектор покрытия переходов сети $\boldsymbol{\tau}$ интерпретируется как цикл срабатывания (переключения) структурного автомата, или дискретный интервал времени, в течение которого входные сигналы с учетом текущего внутреннего состояния автомата достигают выходов и переводят автомат в новое состояние. Множество векторов покрытия переходов $\mathbf{S}$ интерпретируется как множество циклов переключения автоматов.

Вектор начальной разметки сети $\boldsymbol{\mu}_0$ интерпретируется как вектор начального состояния КЦА, вектор конечной разметки сети $\boldsymbol{\mu}$ интерпретируется как вектор конечного состояния КЦА. Входные и выходные воздействия в сетевой модели задаются на множестве позиций $\mathbf{P} - \mathbf{P}^0 = \mathbf{P}^- + \mathbf{P}^+$, где $\mathbf{P}$ – множество всех позиций сети, $\mathbf{P}^0$ – множество внутренних позиций сети, $\mathbf{P}^-$ – множество входных позиций сети, $\mathbf{P}^+$ – множество выходных позиций сети. Входная комбинация сигналов на входах автомата интерпретируется как вектор начальной разметки входных позиций сети $\boldsymbol{\mu}_0(\mathbf{P}^-)$. Выходная комбинация сигналов на выходах автомата интерпретируется как вектор конечной разметки выходных позиций сети $\boldsymbol{\mu}(\mathbf{P}^+)$. Начальное и конечное внутренние состояния автомата определяются вектором начальной разметки $\boldsymbol{\mu}_0(\mathbf{P}^q)$ и вектором конечной разметки $\boldsymbol{\mu}(\mathbf{P}^q)$, где $\mathbf{P}^q$ – множество внутренних позиций сети, входящих в состав обратных связей.

Моделирование условного разрыва обратных связей для последовательностных схем осуществляется путем исключения соответствующих позиций $\mathbf{P}^q$ из состава внутренних позиций сети $\mathbf{P}^0$. Состояние сети определяется как достижимое, если для него существует решение, удовлетворяющее условию: $\Delta\boldsymbol{\mu}(\mathbf{P}^q) \neq 0$, и как устойчивое, если для него существует решение, удовлетворяющее условию: $\Delta\boldsymbol{\mu}(\mathbf{P}^q) = 0$.

Множество $\Delta\mathbf{M}(\mathbf{P} - \mathbf{P}^0)$, заданное на множестве входных и выходных позиций сети, интерпретируется как исходная таблица истинности (табл.) или таблица переходов (переключений) состояний КЦА. В качестве множества обучающих наборов могут использоваться все наборы из таблицы истинности либо только минимальное множество наборов.

На множестве обучающих наборов сетевая модель КЦА может быть представлена в виде системы матричных уравнений состояний СП:

$$\Delta\mathbf{M} = \mathbf{A} \cdot \mathbf{S}. \tag{6}$$

Для двухразрядного сумматора, приведенного на рисунке 1, на минимальном множестве обучающих наборов система матричных уравнений (6) может быть приведена к виду:

$$
\begin{array}{c}
\begin{matrix} \Delta\boldsymbol{\mu}_2 & \Delta\boldsymbol{\mu}_3 & \Delta\boldsymbol{\mu}_4 & \Delta\boldsymbol{\mu}_5 & \Delta\boldsymbol{\mu}_9 \end{matrix} \\
\begin{bmatrix}
-1 & 0 & -1 & 0 & 0 \\
0 & -1 & -1 & 0 & 0 \\
0 & 0 & 0 & -1 & 0 \\
0 & 0 & 0 & 0 & -1 \\
x & x & x & x & x \\
x & x & x & x & x \\
x & x & x & x & x \\
x & x & x & x & x \\
x & x & x & x & x \\
x & x & x & x & x \\
x & x & x & x & x \\
x & x & x & x & x \\
x & x & x & x & x \\
x & x & x & x & x \\
x & x & x & x & x \\
x & x & x & x & x \\
x & x & x & x & x \\
x & x & x & x & x \\
1 & 1 & 0 & 0 & 0 \\
0 & 0 & 1 & 1 & 1
\end{bmatrix}
\end{array}
=
\begin{array}{cc}
 & \begin{matrix} t_1 & t_2 & t_3 & t_4 & T_5 & T_6 & T_7 & T_8 \end{matrix} \\
\begin{matrix} p_1 \\ p_2 \\ p_3 \\ p_4 \\ p_5 \\ p_6 \\ p_7 \\ p_8 \\ p_9 \\ p_{10} \\ p_{11} \\ p_{12} \\ p_{13} \\ p_{14} \\ p_{15} \\ p_{16} \\ p_{17} \\ p_{18} \\ p_{19} \\ p_{20} \end{matrix} &
\begin{bmatrix}
-1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 \\
1 & 0 & 0 & 0 & -1 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 & -1 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & -1 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & -1 & 0 & 0 & 0 \\
1 & 0 & 0 & 0 & 0 & -1 & 0 & 0 \\
0 & 1 & 0 & 0 & 0 & -1 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 & -1 & 0 & 0 \\
0 & 0 & 0 & 1 & 0 & -1 & 0 & 0 \\
1 & 0 & 0 & 0 & 0 & 0 & -1 & 0 \\
0 & 1 & 0 & 0 & 0 & 0 & -1 & 0 \\
0 & 0 & 1 & 0 & 0 & 0 & -1 & 0 \\
0 & 0 & 0 & 1 & 0 & 0 & -1 & 0 \\
0 & 0 & 0 & 0 & 0 & 1 & 0 & -1 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & -1 \\
0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1
\end{bmatrix}
\end{array}
\cdot
\begin{array}{c}
\begin{matrix} \boldsymbol{\tau}_2 & \boldsymbol{\tau}_3 & \boldsymbol{\tau}_4 & \boldsymbol{\tau}_5 & \boldsymbol{\tau}_9 \end{matrix} \\
\begin{bmatrix}
x & x & x & x & x \\
x & x & x & x & x \\
x & x & x & x & x \\
x & x & x & x & x \\
x & x & x & x & x \\
x & x & x & x & x \\
x & x & x & x & x \\
x & x & x & x & x
\end{bmatrix}
\end{array}.
$$

Разметка внутренних позиций множества $\Delta\mathbf{M}$ не определена. Состав переходов покрытия $\mathbf{S}$ также не определен. Полностью определена только исходная матрица инцидентности $\mathbf{A}$.

2 Синтез логики математической модели КЦА

Для решения задачи синтеза логики математической модели КЦА (обучения) используются методы исчисления инвариантов уравнения состояний маркированного графа структурной схемы автомата. Инварианты маркированного графа являются мощным инструментом исследования структурных свойств сетей и представляют собой решения однородных систем уравнений.

Как известно, для маркированных графов вектор $\Delta\boldsymbol{\mu}$ в уравнении состояний (1) равен нулю на множестве всех внутренних позиций сети:

$$\Delta\boldsymbol{\mu}(\mathbf{P}^0)=0, \qquad (7)$$

где $\mathbf{P}^0$ – множество внутренних позиций сети.

Соответственно, уравнение (1) для маркированных графов имеет решение $\mathbf{x}$ тогда и только тогда, когда вектор $\Delta\boldsymbol{\mu}$ ортогонален любому решению $\mathbf{y}$ соответствующего однородного уравнения:

$$\mathbf{A}^{\mathrm{T}}(\mathbf{P},\mathbf{T}^0)\cdot\mathbf{y}=0, \qquad (8)$$

где $\mathbf{A}^{\mathrm{T}}$ – транспонированная матрица инцидентности сети,

$\mathbf{y} \in \mathbf{R}$ – вектор покрытия позиций сети при $y_j = \{0,1\}$,

$\mathbf{R}$ – покрытие позиций сети для заданного вектора $\Delta\boldsymbol{\mu}$,

$\mathbf{T}^0$ – множество внутренних переходов сети.

Свойство ортогональности для маркированных графов, соответственно, может быть представлено в виде произведения вектора $\Delta\boldsymbol{\mu}$ и покрытия $\mathbf{R}$:

$$\Delta\boldsymbol{\mu}\cdot\mathbf{R}=0. \qquad (9)$$

Следует отметить, что уравнение (8) справедливо только для подкласса однородных СП. Проблема заключается в том, что при моделировании логики свойство однородности маркированных графов утрачивается. Возможность исчисления инвариантов уравнения состояний маркированного графа достигается в результате неявного определения логики маркированного графа. Свойство однородности маркированных графов (7) сохраняется в сетевой модели с неявным определением логики, но не может быть использовано для определения вектора $\Delta\boldsymbol{\mu}$ равным нулю на множестве внутренних позиций сетевой модели при решении уравнения (1).

Тем не менее, свойство ортогональности (9) составляет основу метода синтеза логики КЦА и позволяет определить каждый вектор $\Delta\boldsymbol{\mu}$ в системе уравнений (1) равным нулю на множестве внутренних позиций сети $\mathbf{P}^0_{\mathbf{R}}$, входящих в состав соответствующего покрытия $\mathbf{R}$:

$$\Delta\boldsymbol{\mu}(\mathbf{P}^0_{\mathbf{R}}) = 0. \qquad (10)$$

Для решения однородного на множестве позиций $\mathbf{P}^0_{\mathbf{R}}$ уравнения (1) могут быть использованы стандартные методы исчисления инвариантов с учетом целочисленности. При этом условие (10) служит в качестве критерия достижимости при решении уравнения (1) с неявно определяемой логикой и формируется в процессе решения уравнения (8). Для заданного вектора

$\boldsymbol{\Delta\mu}$ уравнение (1) может иметь множество решений. Поэтому отбор решений выполняется исходя из принципа минимального пересечения множества инвариантов – $\mathbf{S}$ и $\mathbf{R}$ для системы уравнений (6).

В результате исчисления инвариантов система матричных уравнений для двухразрядного сумматора примет вид:

$$
\begin{array}{c}
\begin{array}{ccccc} \Delta\boldsymbol{\mu}_2 & \Delta\boldsymbol{\mu}_3 & \Delta\boldsymbol{\mu}_4 & \Delta\boldsymbol{\mu}_5 & \Delta\boldsymbol{\mu}_9 \end{array} \\
\begin{bmatrix}
-1 & 0 & -1 & 0 & 0 \\
0 & -1 & -1 & 0 & 0 \\
0 & 0 & 0 & -1 & 0 \\
0 & 0 & 0 & 0 & -1 \\
0 & x & x & x & x \\
x & 0 & x & x & x \\
x & x & x & x & x \\
x & x & x & x & x \\
x & x & 0 & x & x \\
x & x & 0 & x & x \\
x & x & x & x & x \\
x & x & x & x & x \\
x & x & x & x & x \\
x & x & x & x & x \\
x & x & x & 0 & x \\
x & x & x & x & 0 \\
x & x & 0 & x & x \\
x & x & x & 0 & 0 \\
1 & 1 & 0 & 0 & 0 \\
0 & 0 & 1 & 1 & 1
\end{bmatrix}
\end{array}
=
\begin{array}{c}
\begin{array}{cccccccc} t_1 & t_2 & t_3 & t_4 & T_5 & T_6 & T_7 & T_8 \end{array} \\
\begin{array}{c} p_1\\p_2\\p_3\\p_4\\p_5\\p_6\\p_7\\p_8\\p_9\\p_{10}\\p_{11}\\p_{12}\\p_{13}\\p_{14}\\p_{15}\\p_{16}\\p_{17}\\p_{18}\\p_{19}\\p_{20} \end{array}
\begin{bmatrix}
-1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 \\
1 & 0 & 0 & 0 & -1 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 & -1 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & -1 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & -1 & 0 & 0 & 0 \\
1 & 0 & 0 & 0 & 0 & -1 & 0 & 0 \\
0 & 1 & 0 & 0 & 0 & -1 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 & -1 & 0 & 0 \\
0 & 0 & 0 & 1 & 0 & -1 & 0 & 0 \\
1 & 0 & 0 & 0 & 0 & 0 & -1 & 0 \\
0 & 1 & 0 & 0 & 0 & 0 & -1 & 0 \\
0 & 0 & 1 & 0 & 0 & 0 & -1 & 0 \\
0 & 0 & 0 & 1 & 0 & 0 & -1 & 0 \\
0 & 0 & 0 & 0 & 0 & 1 & 0 & -1 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & -1 \\
0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1
\end{bmatrix}
\end{array}
\cdot
\begin{array}{c}
\begin{array}{ccccc} \boldsymbol{\tau}_2 & \boldsymbol{\tau}_3 & \boldsymbol{\tau}_4 & \boldsymbol{\tau}_5 & \boldsymbol{\tau}_9 \end{array} \\
\begin{bmatrix}
1 & 0 & 1 & 0 & 0 \\
0 & 1 & 1 & 0 & 0 \\
0 & 0 & 0 & 1 & 0 \\
0 & 0 & 0 & 0 & 1 \\
1 & 1 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 1 & 1 \\
0 & 0 & 1 & 1 & 1
\end{bmatrix}
\end{array}.
$$

при

$$
\begin{bmatrix}
-1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & -1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & -1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & -1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & -1 & -1 & -1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & -1 & -1 & -1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & -1 & -1 & -1 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & -1 & 0 & 1
\end{bmatrix}
\cdot
\begin{array}{c}
\begin{array}{cccccc} \mathbf{R}_2 & \mathbf{R}_3 & & \mathbf{R}_4 & \mathbf{R}_5 & \mathbf{R}_9 \\ \mathbf{y}_1 & \mathbf{y}_2 & \mathbf{y}_3 & \mathbf{y}_4 & \mathbf{y}_5 & \mathbf{y}_6 \end{array} \\
\begin{array}{c} p_1\\p_2\\p_3\\p_4\\p_5\\p_6\\p_7\\p_8\\p_9\\p_{10}\\p_{11}\\p_{12}\\p_{13}\\p_{14}\\p_{15}\\p_{16}\\p_{17}\\p_{18}\\p_{19}\\p_{20} \end{array}
\begin{bmatrix}
1 & 0 & 1 & 0 & 0 & 0 \\
0 & 1 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 0 \\
0 & 0 & 0 & 0 & 0 & 1 \\
1 & 0 & 0 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 0 \\
0 & 0 & 0 & 0 & 0 & 1 \\
0 & 0 & 1 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 1 \\
1 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & 1 & 1 & 1
\end{bmatrix}
\end{array}
=
\begin{bmatrix}
0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0
\end{bmatrix}.
$$

Проекция неявно определенной входной логики составных переходов компонентов и выходной логики простых переходов линий соединений на исходную структурную схему автомата сводится к решению системы однородных уравнений (1) и (8) с неопределенной матрицей инцидентности:

$$\Delta\boldsymbol{\mu} = \mathbf{A}^{\alpha}(\mathbf{P}, \mathbf{T}) \cdot \mathbf{x} \quad \text{при } \Delta\boldsymbol{\mu}\,(\mathbf{P}^0) = 0$$

и (11)

$$\mathbf{A}^{\mathbf{T}\alpha}(\mathbf{P}, \mathbf{T}^0) \cdot \mathbf{y} = 0 \quad \text{при } \Delta\boldsymbol{\mu} \cdot \mathbf{R} = 0,$$

где $\mathbf{A}^{\alpha} = \mathbf{A}^{+\alpha} - \mathbf{A}^{-\alpha}$ – неопределенная матрица инцидентности, для которой $a_{ij} = \{0, 1, -1, \alpha, -\alpha\}$.

Введение неизвестных в матрицу инцидентности $\mathbf{A}$ уравнения (1) осуществляется для входных дуг составных переходов в соответствии с выражением:

$$a_{ij} = -1 \rightarrow a_{ij} = -\alpha,$$

а для выходных дуг простых переходов – в соответствии с выражением:

$$a_{ij} = 1 \rightarrow a_{ij} = \alpha, \text{ где } \alpha = \{0, 1\}.$$

После введения неизвестных в матрицу инцидентности система матричных уравнений для двухразрядного сумматора может быть представлена как однородная на множестве внутренних позиций сети:

$$
\begin{array}{c}
\begin{array}{ccccc} \Delta\boldsymbol{\mu}_2 & \Delta\boldsymbol{\mu}_3 & \Delta\boldsymbol{\mu}_4 & \Delta\boldsymbol{\mu}_5 & \Delta\boldsymbol{\mu}_9 \end{array} \\
\begin{bmatrix}
-1 & 0 & -1 & 0 & 0 \\
0 & -1 & -1 & 0 & 0 \\
0 & 0 & 0 & -1 & 0 \\
0 & 0 & 0 & 0 & -1 \\
0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 \\
1 & 1 & 0 & 0 & 0 \\
0 & 0 & 1 & 1 & 1
\end{bmatrix}
\end{array}
=
\begin{array}{c}
\begin{array}{cccccccc} t_1 & t_2 & t_3 & t_4 & T_5 & T_6 & T_7 & T_8 \end{array} \\
\begin{array}{c} p_1 \\ p_2 \\ p_3 \\ p_4 \\ p_5 \\ p_6 \\ p_7 \\ p_8 \\ p_9 \\ p_{10} \\ p_{11} \\ p_{12} \\ p_{13} \\ p_{14} \\ p_{15} \\ p_{16} \\ p_{17} \\ p_{18} \\ p_{19} \\ p_{20} \end{array}
\begin{bmatrix}
-1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 \\
\alpha_1 & 0 & 0 & 0 & -\alpha_1 & 0 & 0 & 0 \\
0 & \alpha_2 & 0 & 0 & -\alpha_2 & 0 & 0 & 0 \\
0 & 0 & \alpha_3 & 0 & -\alpha_3 & 0 & 0 & 0 \\
0 & 0 & 0 & \alpha_4 & -\alpha_4 & 0 & 0 & 0 \\
\alpha_5 & 0 & 0 & 0 & 0 & -\alpha_5 & 0 & 0 \\
0 & \alpha_6 & 0 & 0 & 0 & -\alpha_6 & 0 & 0 \\
0 & 0 & \alpha_7 & 0 & 0 & -\alpha_7 & 0 & 0 \\
0 & 0 & 0 & \alpha_8 & 0 & -\alpha_8 & 0 & 0 \\
\alpha_9 & 0 & 0 & 0 & 0 & 0 & -\alpha_9 & 0 \\
0 & \alpha_{10} & 0 & 0 & 0 & 0 & -\alpha_{10} & 0 \\
0 & 0 & \alpha_{11} & 0 & 0 & 0 & -\alpha_{11} & 0 \\
0 & 0 & 0 & \alpha_{12} & 0 & 0 & -\alpha_{12} & 0 \\
0 & 0 & 0 & 0 & 0 & \alpha_{13} & 0 & -\alpha_{13} \\
0 & 0 & 0 & 0 & 0 & 0 & \alpha_{14} & -\alpha_{14} \\
0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1
\end{bmatrix}
\end{array}
\cdot
\begin{array}{c}
\begin{array}{ccccc} \boldsymbol{\tau}_2 & \boldsymbol{\tau}_3 & \boldsymbol{\tau}_4 & \boldsymbol{\tau}_5 & \boldsymbol{\tau}_9 \end{array} \\
\begin{bmatrix}
1 & 0 & 1 & 0 & 0 \\
0 & 1 & 1 & 0 & 0 \\
0 & 0 & 0 & 1 & 0 \\
0 & 0 & 0 & 0 & 1 \\
1 & 1 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 1 & 1 \\
0 & 0 & 1 & 1 & 1
\end{bmatrix}
\end{array}.
$$

Для каждого вектора $\boldsymbol{\tau}$ вычисляется соответствующая матрица $\mathbf{A}^{\boldsymbol{\tau}}$. Значения неизвестных для ингибиторных дуг определяются неявно и равны нулю, что обеспечивает решение проблемы матричного представления ингибиторных СП. При этом для каждого составного перехода, входящего в состав вектора , определяются соответствующий простой переход и его структурные связи, характерные только для данного вектора $\boldsymbol{\tau}$. В результате объединения матриц $\mathbf{A}^{\boldsymbol{\tau}}$ формируется ингибиторная матрица инцидентности: $\mathbf{A}^{\mathbf{I}} = \cup\, \mathbf{A}^{\boldsymbol{\tau}}$.

Практически проекция неявно определенной логики на исходную структурную схему автомата сводится к обнулению строк исходной матрицы инцидентности для каждого вектора $\boldsymbol{\tau}$ на множестве позиций сети, не входящих в состав соответствующего покрытия $\mathbf{R}$ с последующим объединением матриц для каждого $\mathbf{R}$. Нулевые строки в матрице инцидентности $\mathbf{A}^{\mathbf{I}}$ удаляются.

Для двухразрядного сумматора матрица инцидентности $\mathbf{A^I}$ примет вид:

$$\mathbf{A^I} = \begin{array}{c} \\ p_1 \\ p_2 \\ p_3 \\ p_4 \\ p_5 \\ p_6 \\ p_9 \\ p_{10} \\ p_{15} \\ p_{16} \\ p_{17} \\ p_{18} \\ p_{19} \\ p_{20} \end{array} \begin{array}{c} \begin{array}{ccccccccccc} \mathrm{t}_1 & \mathrm{t}_2 & \mathrm{t}_3 & \mathrm{t}_4 & \mathrm{t}_{5.1} & \mathrm{t}_{5.2} & \mathrm{t}_6 & \mathrm{t}_{7.1} & \mathrm{t}_{7.2} & \mathrm{t}_{8.1} & \mathrm{t}_{8.2} \end{array} \\ \begin{bmatrix} -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & -1 \\ 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \end{bmatrix} \end{array}.$$

Матрица $\mathbf{A^I}$, в свою очередь, может быть представлена в виде ингибиторного графа СП (рис. 2.):

Для формирования формулы (сетевого алгоритма) исходной функции, заданной на множестве обучающих наборов, могут быть использованы операции реляционной алгебры и более сложные реляционные исчисления над множествами $\mathbf{A^\tau}$. В итоге множество полученных матриц $\mathbf{A^\tau}$ составляют объединенную матрицу неоднородной ингибиторной СП или сетевой алгоритм (модель логической схемы) КЦА (рис. 2). Построение модели синаптической памяти нейронов происходит в процессе формирования объединенной матрицы $\mathbf{A^I}$. В дальнейшем полученный сетевой алгоритм в виде матрицы инцидентности $\mathbf{A^I}$ используется в качестве исходной информации для решения задач анализа достижимости и генерации достижимых устойчивых состояний КЦА [6].

3 Генерация решений математической модели КЦА

Математическая модель КЦА позволяет воспроизвести как заданное в процессе обучения множество решений, так и множество решений, которые не были заданы в процессе обучения. Задача генерации решений математической модели КЦА сводится к решению задачи достижимости ингибиторных СП.

Анализ достижимости ингибиторных СП, в свою очередь, сводится к решению системы уравнений с ингибиторной матрицей инцидентности $\mathbf{A^I}$:

$$\Delta\boldsymbol{\mu} = \mathbf{A^I}(\mathbf{P}, \mathbf{T}) \cdot \boldsymbol{\tau} \quad \text{при } \Delta\boldsymbol{\mu}\,(\mathbf{P}^0_{\boldsymbol{\tau}}) = 0,\ \Delta\boldsymbol{\mu}\,(\mathbf{P}^0_{\mathbf{R}}) = 0$$

$$\text{и } \mathbf{A^I}^{\mathrm{T}}(\mathbf{P}, \mathbf{T}^0_{\boldsymbol{\tau}}) \cdot \mathbf{y} = 0 \quad \text{при } \Delta\boldsymbol{\mu} \cdot \mathbf{R} = 0, \tag{12}$$

где $\boldsymbol{\tau}$ – вектор покрытия переходов матрицы $\mathbf{A^I}$,

$\mathbf{T}^0_{\boldsymbol{\tau}}$ – множество внутренних переходов, входящих в состав вектора $\boldsymbol{\tau}$,

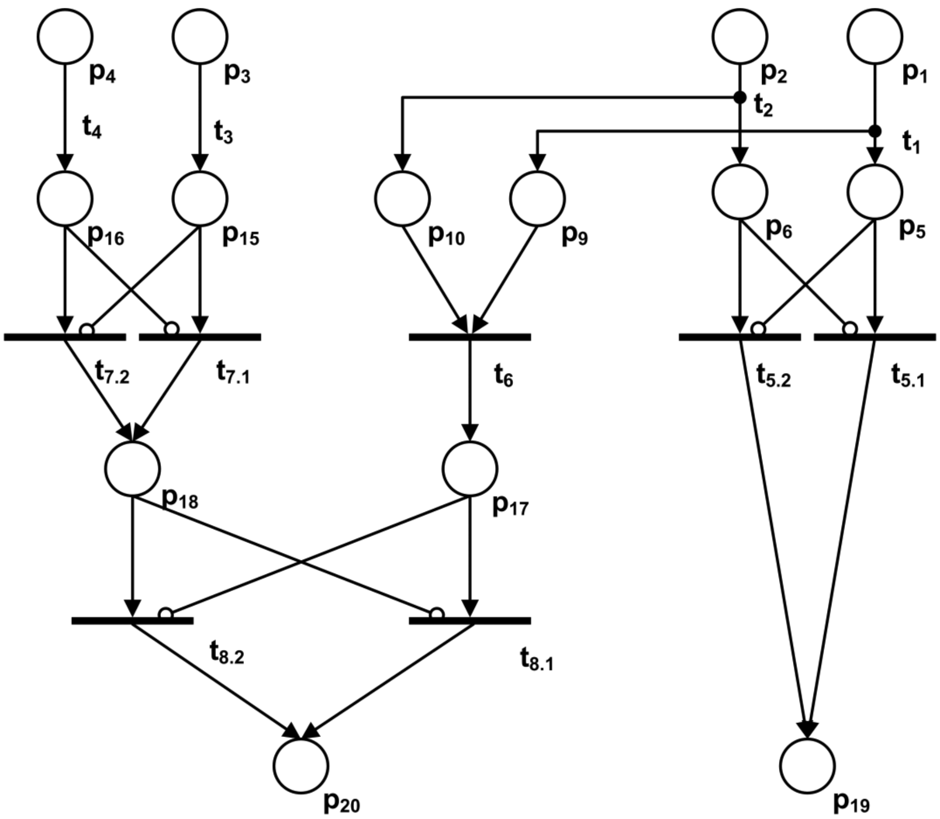

Рис. 2. Ингибиторный граф двухразрядного сумматора

$\mathbf{P}_{\boldsymbol{\tau}}^{0}$ – множество входных позиций вектора $\boldsymbol{\tau}$ на множестве внутренних позиций сети.

Условие $\Delta\boldsymbol{\mu}\ (\mathbf{P}_{\mathbf{R}}^{0})=0$ определяет критерий достижимости входных и выходных позиций сети, условие $\Delta\boldsymbol{\mu}\ (\mathbf{P}_{\boldsymbol{\tau}}^{0})=0$ – критерий (логику) срабатывания переходов сети. При этом критерий достижимости и критерий срабатывания переходов формируются в процессе исчисления инвариантов системы уравнений (12).

В случае если вектор $\Delta\boldsymbol{\mu}$ полностью определен на множестве входных и выходных позиций сети, выполняется анализ достижимости устойчивых состояний автомата (верификация структурной схемы автомата). Система уравнений (12) может иметь только одно решение для каждого вектора $\Delta\boldsymbol{\mu}$.

Генерация достижимых устойчивых состояний КЦА выполняется в случае, если вектор $\Delta\boldsymbol{\mu}$ полностью не определен на множестве входных и выходных позиций сети. В случае неопределенности или неполного определения вектора $\Delta\boldsymbol{\mu}$ система уравнений (12) имеет множество решений. Множество решений системы уравнений (12), в свою очередь, ограничено и зависит от степени определенности вектора $\Delta\boldsymbol{\mu}$. Доопределение вектора $\Delta\boldsymbol{\mu}$ для каждого полученного вектора $\boldsymbol{\tau}$ выполняется путем простого умножения вектора $\boldsymbol{\tau}$ на матрицу инцидентности $\mathbf{A}^{\mathbf{I}}$. Все множество решений может быть получено даже в случае полной неопределенности вектора $\Delta\boldsymbol{\mu}$. Количество решений соответствует количеству возможных переключений автомата или количеству наборов таблицы истинности (таблицы переключений).

Проблема заключается в том, что известные методы генерации решений линейных систем уравнений в целых неотрицательных числах имеют асимптотически экспоненциальную вычислительную сложность, что затрудняет их применение для анализа реальных систем. Критическим с точки зрения эффективности является время генерации (построения) минимального порождающего множества решений (МПМР) на множестве невыраженных переменных. Решение проблемы может быть получено в результате учета специфики логических схем КЦА.

Генерация МПМР уравнения состояний КЦА выполняется исходя из принципа активности компонентов (составных переходов сетевой модели) для каждого состояния автомата. В составе сетевой модели каждого компонента одновременно может быть активизирован только один простой переход (набор из таблицы истинности компонента). Соответственно количество единиц в комбинации равно количеству активных переходов компонентов схемы. Данное ограничение определяется спецификой срабатывания КЦА и необходимо для минимизации перебора комбинаций переходов, а также для исключения возможных недействительных решений. Практически множество решений, полученных в процессе обучения КЦА для минимального множества обучающих наборов, может быть использовано в качестве МПМР.

Все множество возможных решений для двухразрядного сумматора приведено ниже:

$$
\begin{array}{c}
\\ \\ \\ \\ \\ \\ S = \\ \\ \\ \\ \\ \\
\end{array}
\begin{array}{r}
\\ t_1 \\ t_2 \\ t_3 \\ t_4 \\ t_{5.1} \\ t_{5.2} \\ t_6 \\ t_{7.1} \\ t_{7.2} \\ t_{8.1} \\ t_{8.2}
\end{array}
\begin{array}{c}
\begin{array}{cccccccccccccccc}
\boldsymbol{\tau}_1 & \boldsymbol{\tau}_2 & \boldsymbol{\tau}_3 & \boldsymbol{\tau}_4 & \boldsymbol{\tau}_5 & \boldsymbol{\tau}_6 & \boldsymbol{\tau}_7 & \boldsymbol{\tau}_8 & \boldsymbol{\tau}_9 & \boldsymbol{\tau}_{10} & \boldsymbol{\tau}_{11} & \boldsymbol{\tau}_{12} & \boldsymbol{\tau}_{13} & \boldsymbol{\tau}_{14} & \boldsymbol{\tau}_{15} & \boldsymbol{\tau}_{16}
\end{array} \\
\begin{bmatrix}
0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 \\
0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 \\
0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 \\
0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 \\
0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\
0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0
\end{bmatrix}
\end{array}
$$

и

$$
\begin{array}{c}
\\ \\ \\ \\ \\ \\ \\ \Delta\mathbf{M} = \\ \\ \\ \\ \\ \\ \\
\end{array}
\begin{array}{r}
\\ p_1 \\ p_2 \\ p_3 \\ p_4 \\ p_5 \\ p_6 \\ p_9 \\ p_{10} \\ p_{15} \\ p_{16} \\ p_{17} \\ p_{18} \\ p_{19} \\ p_{20}
\end{array}
\begin{array}{c}
\begin{array}{cccccccccccccccc}
\Delta\boldsymbol{\mu}_1 & \Delta\boldsymbol{\mu}_2 & \Delta\boldsymbol{\mu}_3 & \Delta\boldsymbol{\mu}_4 & \Delta\boldsymbol{\mu}_5 & \Delta\boldsymbol{\mu}_6 & \Delta\boldsymbol{\mu}_7 & \Delta\boldsymbol{\mu}_8 & \Delta\boldsymbol{\mu}_9 & \Delta\boldsymbol{\mu}_{10} & \Delta\boldsymbol{\mu}_{11} & \Delta\boldsymbol{\mu}_{12} & \Delta\boldsymbol{\mu}_{13} & \Delta\boldsymbol{\mu}_{14} & \Delta\boldsymbol{\mu}_{15} & \Delta\boldsymbol{\mu}_{16}
\end{array} \\
\begin{bmatrix}
0 & -1 & 0 & -1 & 0 & -1 & 0 & -1 & 0 & -1 & 0 & -1 & 0 & -1 & 0 & -1 \\
0 & 0 & -1 & -1 & 0 & 0 & -1 & -1 & 0 & 0 & -1 & -1 & 0 & 0 & -1 & -1 \\
0 & 0 & 0 & 0 & -1 & -1 & -1 & -1 & 0 & 0 & 0 & 0 & -1 & -1 & -1 & -1 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 \\
0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 \\
0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 \\
0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\
0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 \\
0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 1
\end{bmatrix}
\end{array}.
$$

Для построения протоколов достижимости ингибиторных СП могут быть использованы стандартные методы построения протоколов достижимости обычных СП. В общем случае процедура построения протоколов достижимости сводится к вычислению последовательности векторов запуска переходов и текущих разметок сети для каждого вектора $\boldsymbol{\tau}$, начиная с вектора начальной разметки $\boldsymbol{\mu}_0$ и до тех пор, пока не будет достигнута разметка $\boldsymbol{\mu}$. Последовательность векторов запуска переходов $\mathbf{u}_k$ и векторов текущей разметки $\boldsymbol{\mu}_k$ может быть получена путём итеративного решения уравнения (2). На каждом шаге итерации $k = \overline{1, n}$ все переходы, входящие в состав вектора $\boldsymbol{\tau}$, проверяются на выполнение условия (3). Переходы, удовлетворяющие условию (3), каждый в отдельности и все одновременно, составляют вектор $\mathbf{u}_k$. На каждом шаге итерации $\boldsymbol{\tau} = \boldsymbol{\tau} - \mathbf{u}_k$. В отличие от процедуры построения протоколов достижимости обычных СП наличие исходного вектора $\boldsymbol{\tau}$ является обязательным.

В случае если вектор $\boldsymbol{\tau}$ не определен, обязательным является наличие вектора $\boldsymbol{\mu}_0$, полностью определенного на множестве входных позиций сети. Система уравнений (12) в данном случае имеет только одно решение, которое может быть получено путем вычисления последовательности векторов запуска переходов и текущей разметки сети, начиная с заданной начальной разметки сети $\boldsymbol{\mu}_0$ и до тех пор, пока сеть не перейдет в устойчивое или тупиковое состояние. При этом начальная разметка $\boldsymbol{\mu}_0$ может изменяться в процессе построения протоколов достижимости. Возможность достижимости устойчивых или тупиковых состояний зависит от длительности и последовательности начальных разметок сети.

Соответствующая последовательность векторов запуска переходов и векторов текущей разметки сети вычисляется путём итеративного решения уравнения состояний:

$$\boldsymbol{\mu}_k = \boldsymbol{\mu}_0 + \mathbf{A}^{+\mathbf{I}} \cdot \mathbf{u}_k \tag{13}$$

при условии, что

$$\boldsymbol{\mu}_{k-1}(\mathbf{P}_{\mathbf{u}_k}) - \mathbf{A}^{-\mathbf{I}} \cdot \mathbf{u}_k = 0, \tag{14}$$

где $\boldsymbol{\mu}_k$ – вектор текущей разметки сети,

$\boldsymbol{\mu}_0$ – вектор начальной разметки сети,

$\mathbf{u}_k$ – вектор запуска переходов в сети,

$\mathbf{P}_{\mathbf{u}_k}$ – множество входных позиций переходов, входящих в состав вектора $\mathbf{u}_k$.

Уравнение (13) определяет правило смены разметки ингибиторных СП для моделирования потока данных. Условие (14) служит в качестве правила запуска переходов ингибиторных СП. Длительность сигналов моделируется как сохранение разметки соответствующих позиций сети в течение одной или более итераций. При одинаковом времени задержки всех компонентов время срабатывания переходов принимается равным дискретному интервалу времени. Соответственно, время достижимости интерпретируется как количество итераций.

Заключение

Математическая модель КЦА апробирована на примере моделирования n-разрядного параллельного сумматора. В результате обучения была сформирована математическая модель n-разрядного параллельного сумматора, которая позволила с заданной точностью воспроизводить все множество наборов таблицы истинности n-разрядного параллельного сумматора.

Особое преимущество предлагаемой математической модели КЦА заключается в том, что процедура обучения может выполняться на ограниченном (минимальном) множестве обучающих наборов. С ростом разрядности сумматора при экспоненциальном росте общего количества наборов достигается практически линейная зависимость роста обучающих наборов. Соответственно, достигается приемлемое время обучения. При этом разрядность сумматора ограничена только емкостью памяти и быстродействием компьютера.

Кроме того, математическая модель КЦА может быть использована не только для решения арифметических задач, но и задач распознавания и управления. Для функции распознавания при экспоненциальном росте проверяющих наборов количество обучающих наборов соизмеримо с количеством типов объектов распознавания. Для функции управления основным критерием качества результата считается возможность формирования управленческих решений (управляющих кодов) и их последовательностей, не предусмотренных в процессе обучения.

Логика компонентов (составных переходов) исходной структурной схемы КЦА в процессе синтеза (обучения) на каждом шаге обучения может изменяться, т. е. таблицы истинности компонентов сети не имеют фиксированного размера и, соответственно, фиксированной логической функции. Процесс формирования логики компонентов ограничен только количеством входов компонентов или полным перебором возможных комбинаций сигналов на входах каждого компонента. При условии достаточно большого количества входов каждого компонента процесс формирования логики компонентов и сети в целом практически бесконечен. Непрерывность подачи последовательности входных воздействий определяет бесконечность процесса генерации решений, что в сочетании с бесконечностью процесса формирования логики компонентов обеспечивает возможность преодоления тупиковых состояний (мышления).

Для решения задач построения математической модели КЦА, синтеза логики (обучения) и генерации решений могут быть использованы стандартные методы и средства решения матричных уравнений или СЛАУ. Моделирование n-разрядного параллельного сумматора выполнено в среде Tensor Flow. Tensor Flow представляет вычисления в виде графов потоков данных, отслеживающих свое состояние (stateful dataflow graph), и позволяет перенести выполнение ресурсоемких вычислительных задач из среды с одним CPU (Central Processing Unit) в гетерогенную быструю среду с несколькими GPU (Graphics Processing Unit). Перспективной представляется возможность моделирования на базе TPU (Tensor Processing Unit). Принимая во внимание возможности математической модели КЦА, Tensor Flow призвана обеспечить массовый параллелизм и высокую масштабируемость машинного обучения и генерации решений. Не исключается также возможность реализации математической модели на базе FPGA (Field-Programmable Gate Array).

СПИСОК ЛИТЕРАТУРЫ

1. Principal directions of developing the design methods for intelligent systems to control robots / V.V. Kozhevnikov, V.V. Prikhodko, M.Yu. Leontyev et al. // Journal of Numerical Analysis, Industrial and Applied Mathematics (JNAIAM). 2018. Vol. 12, no. 1−2, pp. 29−49.

2. The concept of designing an intellectual robot control system based on the mathematical model of cognitive digital automata / V.V. Kozhevnikov, V.V. Prikhodko, M.Yu. Leontyev et al. // Journal of Numerical Analysis, Industrial and Applied Mathematics (JNAIAM). 2018. Vol. 12, no. 3−4, pp. 63−79.

3. Станкевич Л.А., Юревич Е.И. Искусственный интеллект и искусственный разум в робототехнике. − СПб. : Изд-во Политехнического ун-та, 2012. − 167 с.

4. Кожевников В.В. Концепция математического моделирования когнитивных цифровых автоматов // Ученые записки УлГУ. Сер. Математика и информационные технологии. − 2015. − Вып. 1 (7). − С. 62−69.

5. Кожевников В.В. Метод математического моделирования логических схем цифровых автоматов // Автоматизация процессов управления. − 2012. − № 4 (30). – С. 97–101.

6. Кожевников В.В. Метод анализа достижимости устойчивых состояний логических схем цифровых автоматов // Автоматизация процессов управления. − 2014.− № 1 (35). − С. 99–108.

7. Питерсон Д. Теория сетей Петри и моделирование систем. − М. : Мир, 1984. − 264 с.

8. Кожевников В.В. Метод анализа достижимости ингибиторных сетей Петри // Автоматизация процессов управления. −2013. − № 3 (33) − С. 29–34.

9. Кожевников В.В. Концепция сетей Петри с программируемой логикой // Ученые записки УлГУ. Сер. Математика и информационные технологии. − 2014. − Вып. 1 (6). − С. 50–55.

10. Мурата Т. Сети Петри. Свойства, анализ, приложения // ТИИЭР. − 1989. − № 44. − С. 41–85.

REFERENCES

1. Kozhevnikov V.V., Prikhodko V.V., Leontev M.Iu. et al. Principal Directions of Developing the Design Methods for Intelligent Systems to Control Robots. *Journal of Numerical Analysis, Industrial and Applied Mathematics (JNAIAM),* 2018, vol. 12, no. 1–2, pp. 29–49.

2. Kozhevnikov V.V., Prikhodko V.V., Leonteev M.Iu. et al. The Concept of Designing an Intellectual Robot Control System Based on the Mathematical Model of Cognitive Digital Automata. *Journal of Numerical Analysis, Industrial and Applied Mathematics (JNAIAM),* 2018, vol. 12, no. 3-4, pp. 63–79.

3. Stankevich L.A., Iurevich E.I. I*skusstvennyi intellekt i iskusstvennyi razum v robototekhnike* [Artificial Intelligence and Artificial Brain in Robotics]. St. Petersburg, Polytechnic University, 2012. 167 p.

4. Kozhevnikov V.V. Kontseptsiia matematicheskogo modelirovaniia kognitivnykh tsifrovykh avtomatov [The Concept of Cognitive Digital Automata Mathematical Modeling]. *Uchenye zapiski UlGU. Ser. Matematika i informatsionnye tekhnologii* [Proc. of UlSU. Ser. Mathematics and Information Technologies], 2015, iss. 1 (7), pp. 62–69.

5. Kozhevnikov V.V. Metod matematicheskogo modelirovaniia logicheskikh skhem tsifrovykh avtomatov [Method of Mathematical Modeling of Logic Circuits of Digital Automata]. *Avtomatizatsiia protsessov upravleniia* [Automation of Control Processes], 2012, no. 4 (30), pp. 97–101.

6. Kozhevnikov V.V. Metod analiza dostizhimosti ustoichivykh sostoianiy logicheskikh skhem tsifrovykh avtomatov [Reachability Analysis Method of Digital Automata Logic Circuits Stable States]. *Avtomatizatsiia protsessov upravleniia* [Automation of Control Processes], 2014, no. 1 (35), pp. 99–108.

7. Peterson D. *Teoriia setei Petri i modelirovanie system* [Petri Nets Theory and the Modeling of Systems]. Moscow, Mir Publ., 1984. 264 p.

8. Kozhevnikov V.V. Metod analiza dostizhimosti ingibitornykh setey Petri [The Accessibility-Analysis Method of Inhibitory Petri Nets]. *Avtomatizatsiia protsessov upravleniia* [Automation of Control Processes], 2013, no. 3 (33), pp. 29–34.

9. Kozhevnikov V.V. Kontseptsiia setei Petri s programmiruemoi logikoi [Petri Nets Concept with Programmable Logic]. *Uchenye zapiski UlGU. Ser. Matematika i informatsionnye tekhnologii* [Proc. of UlSU. Ser. Mathematics and Information Technologies], 2014, iss. 1 (6), pp. 50–55.

10. Murata T. Seti Petri. Svoistva, analiz, prilozheniia [Petri Nets. Properties, Analysis, Applications]. *TIIER* [Institute of Electrical and Electronics Engineers, Journal], 1989, no. 44, pp. 41–85.