

## ПОИСК СОСТЯЗАНИЙ СИГНАЛОВ В НЕЧЕТКИХ АСИНХРОННЫХ АВТОМАТАХ

**СПЕРАНСКИЙ Дмитрий Васильевич**, докт. техн. наук, профессор; e-mail: speranskiy.dv@gmail.com  
**ЛУНЕВ Сергей Александрович**, канд. техн. наук, доцент; e-mail: slunev@mail.ru

Российский университет транспорта (МИИТ), кафедра «Системы управления транспортной инфраструктурой», Москва

Статья посвящена задаче поиска и ликвидации состязаний сигналов в электронных устройствах. Объектами исследования являются дискретные устройства с памятью, работа которых описывается с использованием математической модели нечеткого автомата. Предложена модель нечеткого автомата, которая учитывает специфику рассматриваемой задачи. В этой модели феномен нечеткости проявляется в переходах между состояниями нечеткого автомата. Как и в случае классического (четкого) автомата, задача сводится к противогоночному кодированию состояний нечеткого автомата. В статье предложена концепция построения методов такого кодирования для нечеткого автомата. Идея его состоит в преобразовании заданного нечеткого автомата в виде нечеткого графа в некоторый специальный четкий граф. Дается описание процедуры такого преобразования. Показано, что все возможные состязания, возникающие на нечетком графе, задающем нечеткий автомат, являются подмножеством состязаний, возникающих на преобразованном упомянутым методом четком графе. Поскольку для поиска состязаний сигналов в четких автоматах разработан обширный арсенал методов, он может быть применен к нечетким автоматам.

**Ключевые слова:** надежность электронных устройств, состязания сигналов, модели нечетких устройств, нечеткие автоматы и графы, преобразования нечетких графов в четкие, методы противогоночного кодирования для нечетких автоматов.

DOI: 10.20295/2412-9186-2024-10-02-178-189

### ▼ Введение

Объектами рассмотрения в статье являются дискретные устройства с памятью (ДУ). Одна из важных проблем при синтезе любого устройства состоит в обеспечении надежности его работы. Известно, что синтез электронных схем ДУ происходит с использованием кодирования состояний автомата, являющегося математической моделью ДУ. Кодирование преобразует таблицу переходов автомата в систему соответствующих булевых функций, реализуемых упомянутой схемой. Для обеспечения надежности ДУ необходима ликвидация отказов блока памяти синтезируемого устройства. Она, в принципе, может быть решена с применением методов помехоустойчивого кодирования, разработанных в теории информации. К сожалению, как указывается в [1], не все известные методы такого кодирования гарантируют решение этой проблемы. В частности, этого не происходит, если в синтезируемой схеме ДУ не устранена возможность возникновения состязаний (гонок) сигналов.

Этой проблеме ныне посвящена обширная литература, включающая отдельные разделы и главы в монографиях, например, [2–4], в учебниках и учебных пособиях и большое количество статей. В этих публикациях в качестве математической модели ДУ использовались традиционный конечный синхронный и асинхронный автоматы, которые ныне принято называть четкими.

Такая модель неприменима в ситуациях, когда точное описание объекта исследования и/или его функционирования отсутствует и получить их сложно или невозможно. В этом случае нужны средства, отражающие возникающие неопределенности. В статье Л. Заде [5] такие средства были предложены: им введено понятие нечеткого множества, ставшего основой для создания ныне широко используемой в приложениях теории нечетких множеств.

Предлагаемая статья посвящена одной из таких задач, состоящей в исследовании состязаний (гонок) сигналов и методов противогоночного кодирования состояний нечетких асинхронных автоматов.

Встречающиеся далее термины и понятия из теории нечетких множеств и связанных с ней новых разделов теорий «нечетких» объектов понимаются так, как они трактуются в монографии А. Кофмана [6]. В силу широкого распространения ныне этой терминологии далее она, как правило, не поясняется.

Поиски публикаций по теме предлагаемой статьи для нечетких автоматов показали их отсутствие.

### 1. Некоторые понятия и определения

Вначале напомним некоторые используемые понятия из [6]. Пусть  $A \subset E$ . Принадлежность элемента  $x$  множеству  $A$  будет записываться с помощью характеристической функции  $\mu_A(x)$ , которая может принимать любое неотрицательное значение.

Запись вида

$$A = \{(x_1 | \mu(x_1), (x_2 | \mu(x_2), \dots, (x_n | \mu(x_n))\}, \quad (1)$$

где  $\mu(x_i)$  есть значение характеристической функции, определяющей степень принадлежности элемента  $x_i$  подмножеству  $A$ , задает нечеткое подмножество  $A$  множества  $E$ .

Нечеткий автомат может задаваться на языке матриц [7] либо в виде нечеткого графа. Напомним определение соответствующего нечеткого графа [6]. Пусть  $E_1, E_2$  — два множества и пусть элемент  $x \in E_1$ , а  $y \in E_2$ . Множество упорядоченных пар  $(x, y)$  определяет прямое произведение  $E_1 \cdot E_2$ . Нечеткое подмножество  $G$  такое, что

$$\forall (x, y) \in E_1 \cdot E_2 \mu_G(x, y) \in M, \quad (2)$$

где  $M$  — множество принадлежностей множества  $E_1 \cdot E_2$ , называется нечетким графом. В задачах кодирования состояний далее предполагается, что все элементы  $x_i$  любого подмножества  $A$  состояний нечеткого автомата имеют одинаковые значения характеристической функции  $\mu(x_i) = 1$ . Это интерпретируется просто как факт принадлежности элемента  $x_i$  подмножеству  $A$ . Потому для упрощения записи (1) в ней выражения  $\mu(x_i) = 1$  будут просто опускаться. Дело в том, что в задачах кодирования состояний используется только информация о значениях выходов триггеров, кортежи

которых образуют коды состояний нечеткого автомата.

В выражении (2) дуга  $(x, y)$  графа  $G$  нечеткого автомата принадлежит множеству его дуг, если  $\mu_G(x, y) = 1$ . По аналогии со сказанным выше значение  $\mu_G(x, y) = 1$  имеет место для всех дуг графа из множества  $E_1 \cdot E_2$  и поскольку эти значения явно не используются в задачах кодирования, то они также не указываются.

Условимся, что граф  $G$ , используемый далее в качестве модели нечеткого автомата, задается и изображается как классический четкий граф, но в отличие от него содержит вершины двух типов. Вершины первого типа содержат только одно состояние нечеткого автомата (далее они именуется просто «вершинами»). Вершины второго типа содержат некоторое подмножество состояний нечеткого автомата с числом элементов не менее двух (далее они именуется как  $a$ -вершины (альтернативные)).

В процессе работы нечеткого автомата происходят переходы от одного состояния  $s$  к другому состоянию  $\tilde{s}$ , которые будем обозначать как  $s \rightarrow \tilde{s}$ . В графе нечеткого автомата могут иметь место следующие виды переходов:

- 1)  $v \rightarrow w$ , где  $v$  и  $w$  — его вершины (состояния);
- 2)  $v \rightarrow y$ , где  $v$  — вершина,  $y$  —  $a$ -вершина;
- 3)  $y \rightarrow v$ , обратный переход для указанного в п. 2;
- 4)  $t \rightarrow z$ , где  $t$  и  $z$  — есть  $a$ -вершины.

Исходной информацией для решения рассматриваемой задачи является граф  $G$  нечеткого автомата, множество его состояний и множество разрешенных (допустимых) переходов (вершин-преемников) для каждого состояния нечеткого автомата. Условимся обозначать через  $T(s)$  множество разрешенных состояний-преемников для состояния  $s$ , нечеткого автомата, которые считаются заданными.

Как сказано выше, граф  $G$  содержит два вида вершин и связи (переходы) между ними. Этот граф будет преобразовываться далее в некоторый четкий граф  $\tilde{G}$ . Составная часть графа  $G$ , содержащая все (одноэлементные) вершины и их связи между собой (дуги), включается в синтезируемый поэтапно (пошагово) четкий граф  $\tilde{G}$ . Эту часть  $G$  назовем начальным четким подграфом графа  $G$ . В процессе преобразования этот подграф будет на каждом

очередном этапе пополняться новыми (одноэлементными) вершинами и их связями (дугами). Полученные таким образом новые подграфы мы будем называть текущими четкими подграфами графа  $G$  и обозначать как  $DS(G)$ . Процесс построения четкого графа  $\tilde{G}$  завершится, если после выполнения его очередного шага множество вершин (состояний) текущего подграфа  $DS(G)$  окажется равным множеству вершин (состояний) заданного графа нечеткого автомата.

Выполнение переходов нечеткого автомата последних трех видов, приведенных выше, потребует описания правил выполнения и замещения их в графе  $G$  некоторыми фрагментами из вершин и их связей. Эти правила, в частности, позволят контролировать корректность присутствующих в графе заданного нечеткого автомата переходов вида 2–4.

## 2. Преобразование нечеткого графа нечеткого автомата в четкий граф

В [7] описана модель нечеткого конечного автомата, являющаяся достаточно общей и полезной при решении ряда задач. С другими моделями нечетких автоматов можно ознакомиться, например, в работах [8–10]. Рассматриваемая в статье задача противогоночного кодирования имеет свою специфику, позволяющую использовать вместо упомянутой модели более простую и компактную. Ее легко получить формализованным путем из модели в [7] за счет отказа от применения матричного задания функций переходов и выходов нечеткого автомата и некоторых других упрощений. Изменения эти не принципиальны, но делают изложение материала более ясным и обозримым приводимый пример.

Идея предлагаемого в статье способа решения задачи противогоночного кодирования для нечеткого автомата состоит в преобразовании заданного нечеткого графа  $G$  в четкий граф  $\tilde{G}$ , который используется для нахождения искомого кодирования. Отметим, что для четких автоматов ныне известен целый ряд методов для решения названной задачи и потому весь этот арсенал может быть применен к нечетким автоматам.

Будет показано, что полученное такими методами кодирование состояний для графа  $\tilde{G}$  обеспечивает отсутствие состязаний также и в автомате, заданном нечетким графом  $G$ .

Изложение процесса преобразования заданного автомата (в виде нечеткого графа) описывается ниже и будет иллюстрироваться на примерах. С этой целью рассмотрим граф нечеткого автомата, представленный на рис. 1, имеющий 7 состояний  $s_1, \dots, s_7$ . Для них заданы множества возможных состояний-преемников, но в общем случае не всегда указываются входные сигналы, по которым происходит переход в них. Для нашего примера пусть эти множества таковы:

$$\begin{aligned} T(s_1) &= \{s_1, s_2\}, & T(s_2) &= \{s_2, s_3, s_5\}, \\ T(s_3) &= \{s_3, s_4, s_7\}, & T(s_4) &= \{s_1, s_4\}, \\ T(s_5) &= \{s_3, s_5, s_6\}, & T(s_6) &= \{s_6\}, & T(s_7) &= \{s_7\}. \end{aligned}$$

В рассматриваемом примере упомянутый выше начальный четкий подграф  $DS(G)$  содержит только две вершины ( $s_1, s_2$ ) с инцидентными дугами.

Вначале опишем каким образом при движении по графу  $G$  нечеткого автомата определяются концевые вершины дуг при переходах.

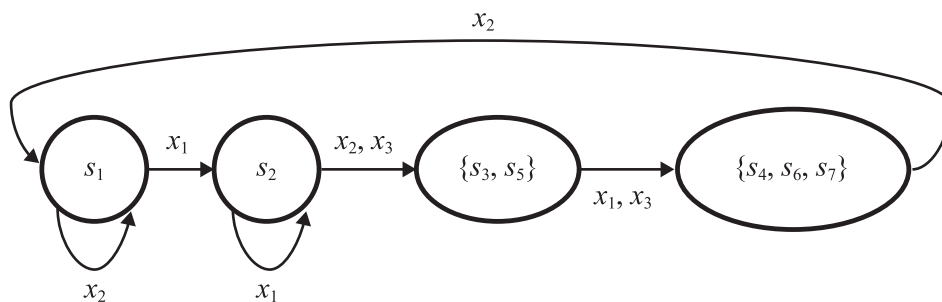


Рис. 1. Нечеткий граф автомата

Общий принцип, по которому они формируются, покажем на примере. Пусть имеется переход между  $a$ -состояниями  $\{a_1, a_2, a_3\} \rightarrow \{b_1, b_2\}$ . Эта запись означает, каждое состояние из  $\{a_1, a_2, a_3\}$  переходит в каждое состояние в  $\{b_1, b_2\}$ , что эквивалентно следующим переходам:  $a_1 \rightarrow b_1, a_1 \rightarrow b_2, a_2 \rightarrow b_1, a_2 \rightarrow b_2, a_3 \rightarrow b_1, a_3 \rightarrow b_2$ . Заметим, что некоторые переходы в действительности могут отсутствовать из-за ограничений, порождаемых задаваемыми множествами  $T(a_1), T(a_2), T(a_3)$ .

Переходы 1-го вида — это обычные стандартные переходы между состояниями в четких автоматах, изображаемые в виде одной дуги, связывающей их концевые вершины.

В случае переходов вида 2–4, содержащих  $a$ -состояния, ситуация меняется. При синтезе четкого графа в качестве модели для нечеткого автомата вместо одной дуги могут порождаться одновременно несколько дуг, замещающих один такой переход.

Рассмотрим переход 2-го вида. Пусть  $\tilde{S} = \{\tilde{s}_1, \dots, \tilde{s}_k\}$  есть некоторое подмножество состояний (вершин) нечеткого автомата, тогда такой переход изображается в графе как  $s \rightarrow \{\tilde{s}_1, \dots, \tilde{s}_k\}$ . Здесь  $\{\tilde{s}_1, \dots, \tilde{s}_k\}$  есть  $a$ -состояние. При синтезе модели четкого графа  $\tilde{G}$  этот переход замещается фрагментом, изображенным на рис. 2.

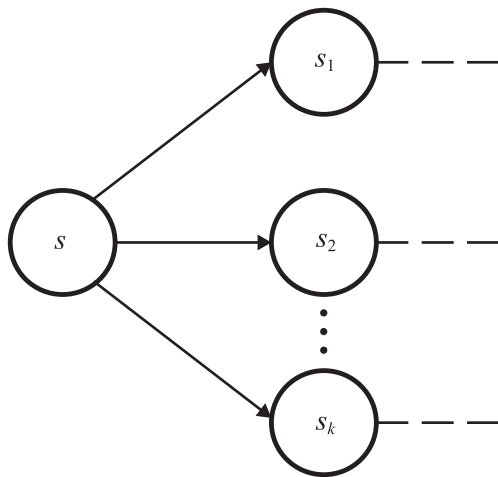


Рис. 2. Фрагмент графа для замещения переходов 2-го вида

Таким образом, один переход 2-го вида в графе заменяется  $k$  переходами 1-го вида  $s \rightarrow \tilde{s}_i$ , т. е. состояние  $s$  получает  $k$  состояний-преемников в графе  $\tilde{G}$ . Так, в нашем при-

мере переход  $s_2 \rightarrow \{s_3, s_5\}$  заменяется двумя переходами  $s_2 \rightarrow s_3, s_2 \rightarrow s_5$ . Перейдем к переходам 3-го вида, изображаемым в графе  $G$  как  $\{\tilde{s}_1, \dots, \tilde{s}_k\} \rightarrow s$ .

Используя множества  $T(s_i)$ , найдем множества состояний  $\tilde{E}_1 = \{s\} \cap T(s_1), \dots, \tilde{E}_k = \{s\} \cap T(s_k)$ . Если все они окажутся пустыми, то это означает невозможность перехода в состояние  $s$ , то есть он некорректен.

Пусть среди них имеются непустые пересечения  $\tilde{E}_{i_1}, \dots, \tilde{E}_{i_m}$ . Рассмотрим возможные варианты при наличии непустых пересечений:

а) если  $\tilde{E}_{i_1} \cup \dots \cup \tilde{E}_{i_m} = \{s\}$ , то исследуемый переход в качестве преемника имеет единственное состояние  $s$ . В этом случае исследуемый переход замещается фрагментом графа, изображенным на рис. 3.

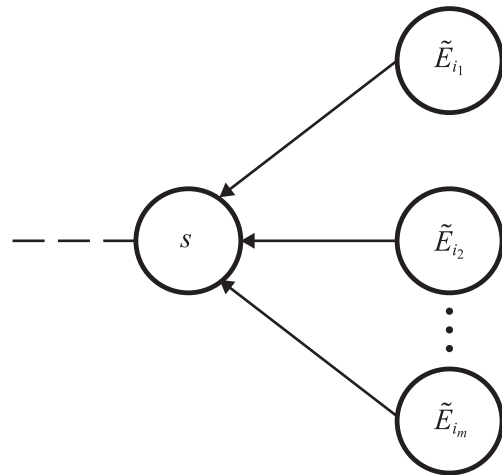
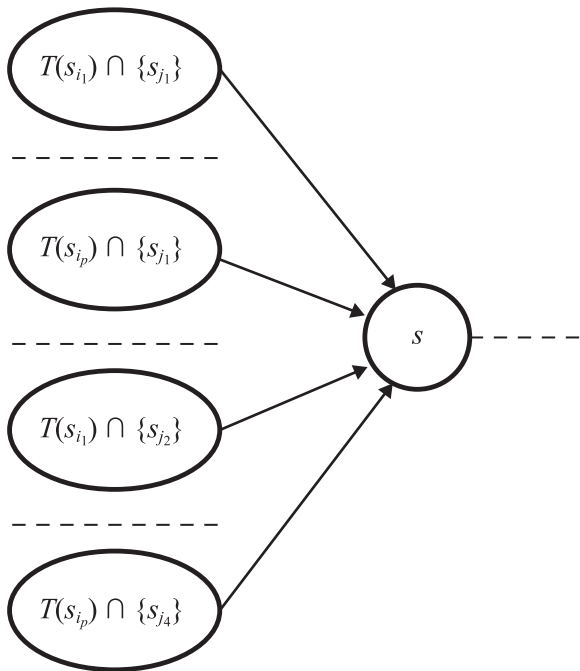


Рис. 3. Фрагмент графа для замещения переходов 3-го вида (выполнено условие а)

б) если равенство из пункта а) не выполняется, но выполняется условие  $s \in (\{T(\tilde{s}_1) \cup \dots \cup T(\tilde{s}_k)\})$ , то состоянием-преемником рассматриваемого перехода также является состояние  $s$ .

В этом случае исследуемый переход замещается фрагментом графа, изображенным на рис. 4.

Так, в нашем примере имеется переход  $\{s_4, s_6, s_7\} \rightarrow s_1$ . Легко проверить, что в этом случае выполняется сразу оба пункта а) и б), приведенные выше, так что указанный переход корректен и состоянием-преемником для  $\{s_4, s_6, s_7\}$  является  $s_1$ . Этот переход замещается в  $DS(G)$  только одной дугой  $s_4 \rightarrow s_1$ , поскольку  $s_1$  не является в нашем примере преемником для состояний  $s_6, s_7$ .



**Рис. 4.** Фрагмент графа для замещения переходов 3-го вида (выполнено условие б)

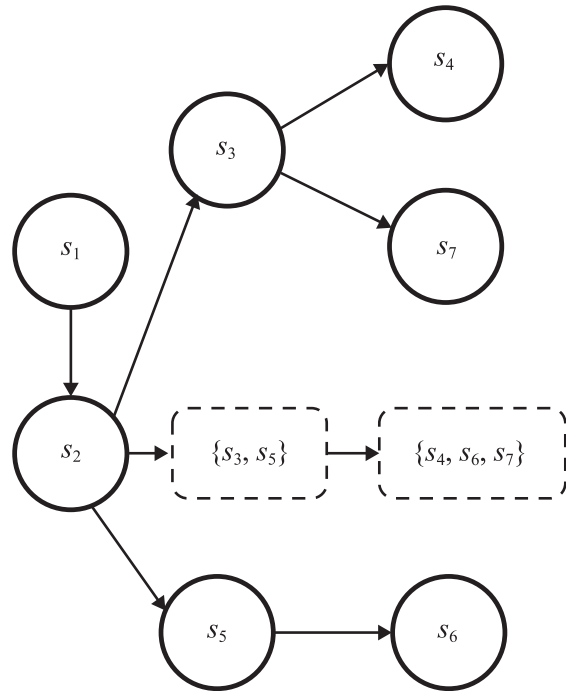
Рассмотрим теперь переход 4-го вида  $\{s_{i_1}, \dots, s_{i_p}\} \rightarrow \{s_{j_1}, \dots, s_{j_q}\}$ . Проверим корректность этого перехода. С этой целью найдем множества  $T(s_{i_v}) \cap \{s_{j_w}\}$  для всех пар состояний  $(s_{i_v}, s_{j_w})$ , порожденных  $a$ -состояниями рассматриваемого перехода. Если объединение всех указанных множеств содержит множество  $\{s_{j_1}, \dots, s_{j_q}\}$ , то этот переход корректен. При преобразовании графа  $G$  в четкий граф все (одноэлементные) вершины  $T(s_{i_v}) \cap \{s_{j_w}\}$  (они являются концами дуг в текущем четком подграфе  $DS(G)$ ) замещают рассматриваемый нами переход. Это означает, что подграф  $DS(G)$  пополняется новыми одноэлементными вершинами.

Так, в нашем примере имеется переход между  $a$ -состояниями  $\{s_3, s_5\} \rightarrow \{s_4, s_6, s_7\}$ . Найдем следующие множества:

$$\begin{aligned} T(s_3) \cap \{s_4\} &= \{s_4\}, & T(s_3) \cap \{s_6\} &= \emptyset, \\ T(s_3) \cap \{s_7\} &= \{s_7\}, & T(s_5) \cap \{s_4\} &= \emptyset, \\ T(s_5) \cap \{s_6\} &= \{s_6\}, & T(s_3) \cap \{s_7\} &= \{s_7\}. \end{aligned}$$

Объединение всех этих множеств содержит состояния  $\{s_4, s_6, s_7\}$ . Итак, этот переход корректен и при преобразовании  $G$  в четкий

граф он замещается переходами  $s_3 \rightarrow s_4, s_3 \rightarrow s_7, s_5 \rightarrow s_6$ . Последнее означает, что состояния  $s_4, s_6, s_7$  появятся в подграфе  $DS(G)$  в качестве одноэлементных вершин. В рассматриваемом примере этот переход замещается фрагментом графа, изображенным на рис. 5.



**Рис. 5.** Фрагмент графа для замещения переходов 4-го вида (частный пример)

Опишем теперь в целом поэтапный процесс преобразования нечеткого графа  $G$ , который задает нечеткий автомат. Первый этап состоит в произвольном выборе некоторой вершины  $s$  в начальном четком подграфе  $G$ . Из нее в общем случае исходит  $t$  дуг  $s \rightarrow s_1, \dots, s_t$ . В зависимости от концевых вершин этих дуг такие переходы относятся к одному из четырех видов. В соответствии с изложенным выше происходит «обработка» каждого из перечисленных переходов, которая в результате приводит к пополнению  $DS(G)$  новыми вершинами, а также инцидентными им дугами. Затем происходит проверка равенства множества состояний  $DS(G)$  множеству состояний графа  $G$ . Если равенство не выполняется, осуществляем «обработку» тех переходов в графе, которые еще содержат  $a$ -состояния. Далее процесс продолжается аналогичным образом до получения очередного подграфа  $DS(G)$ , в котором все вершины являются обычными одноэлементными.



В модели нечеткого автомата, предложенной в [7], при подаче некоторой входной последовательности, выполняются переходы от состояния к состоянию, начиная с заданного начального. Последовательность состояний, через которые он проходит в результате подачи на него некоторой входной последовательности сигналов, назовем траекторией движения автомата. Выбор очередного звена выполняется по различным правилам в зависимости от рассматриваемой задачи. Отметим, что выбор звена в общем случае приходится выполнять из нескольких возможных вариантов продолжения траектории. В [7] это делается с учетом степеней принадлежности нескольких состояний и дуг строящейся траектории, но результатом выбора всегда является единственный вариант. При этом все остальные потенциально возможные варианты исключаются из рассмотрения. Очевидно, что специфика рассматриваемой нами задачи кодирования состояний при поиске опасных состязаний в нечетком автомате требует знания всех возможных вариантов траекторий движения нечеткого автомата, среди которых находятся и опасные траектории, движение по которым приводят к опасным состязаниям сигналов. Понятие опасных состязаний сигналов трактуется нами так, как это определяется в [2] (разделы 59.1 и 59.2 главы 59).

Понятно, что предлагаемый способ сведения задачи поиска опасных траекторий на нечетком графе  $G$  нечеткого автомата к решению той же задачи на синтезируемом нами четком графе  $\tilde{G}$  будет обоснован, если справедливо утверждение: множество всех возможных траекторий на графе  $G$ , получаемых по описанным выше правилам реализации переходов всех 4 видов, является подмножеством множества всех возможных траекторий (в частном случае совпадает) четкого графа  $\tilde{G}$ .

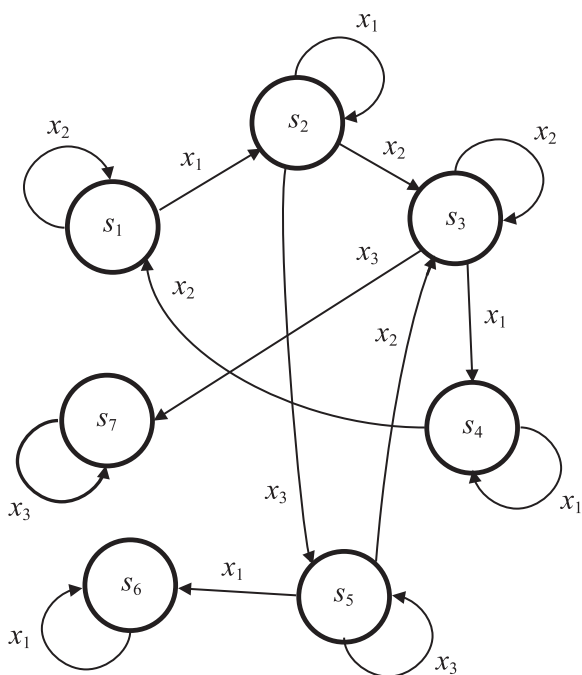
Из изложенного выше (правил выполнения всех 4 видов переходов, определения для них состояний-преемников, замещения таких переходов фрагментами схемы из одноэлементных вершин) справедливость сформулированного утверждения достаточно очевидна. Однако ее можно подтвердить и формальным

(довольно громоздким, но строгим в математическом плане) доказательством, которое вряд ли уместно помещать в техническом журнале.

Проиллюстрируем теперь поэтапное преобразование нечеткого графа на рис. 1 в четкий граф. Множества возможных состояний-преемников  $T(s_i)$  для них заданы и представлены выше.

Начальный четкий подграф для нечеткого автомата содержит только две вершины  $s_1, s_2$ . Из них в качестве начальной для процесса преобразования нечеткого автомата выберем, например, вершину  $s_1$ . Из нее переходим по графу в вершину  $s_2$ , затем в  $a$ -вершину  $\{s_3, s_5\}$ . В соответствии с правилом для переходов 2-го вида в нашем примере переход  $s_2 \rightarrow \{s_3, s_5\}$  заменяется двумя переходами  $s_2 \rightarrow s_3, s_2 \rightarrow s_5$ . Таким образом, подграф  $DS(G)$  пополняется двумя новыми вершинами  $s_3, s_5$ , представленными на рис. 5. Следующий переход в нашем графе выполняется между  $a$ -состояниями  $\{s_3, s_5\} \rightarrow \{s_4, s_6, s_7\}$  и относится к 4-му виду. Выше он был детально пояснен и показано, что этот переход корректен и при преобразовании  $G$  в четкий граф он замещается переходами между одноэлементными состояниями  $s_3 \rightarrow s_4, s_3 \rightarrow s_7, s_5 \rightarrow s_6$ .

Теперь осталось выполнить по графу  $G$  последний переход  $\{s_4, s_6, s_7\} \rightarrow s_1$ , относящийся к 3-му виду. Выше он также подробно пояснен и показано, что в этом случае выполняются оба пункта а) и б) в правилах «обработки» таких переходов, так что указанный переход корректен и состоянием-преемником для  $\{s_4, s_6, s_7\}$  действительно является  $s_1$ . Этот переход замещается в  $DS(G)$  только одной дугой  $s_4 \rightarrow s_1$ , поскольку  $s_1$  не является преемником для состояний  $s_6, s_7$ . Проверка показывает, что в этом подграфе  $DS(G)$  множество его состояний совпадает со множеством состояний исходного графа нечеткого автомата, т. е. процесс преобразования завершен. В общем случае иногда требуется провести недостающие дуги 1-го вида, связывающие вершины последнего подграфа  $DS(G)$ , на основе заданных подмножеств  $T(s_i)$ . Полученный четкий граф  $\tilde{G}$  для рассматриваемого примера представлен на рис. 6.



**Рис. 6.** Четкий граф, полученный после преобразования нечеткого автомата на рис. 1

### 3. Поиск состязаний сигналов в нечетком автомате

Известно [2–4], что синтез ДУ с памятью начинается с описания его функционирования с использованием соответствующей модели конечного автомата.

Для построения по этой модели реальной схемы  $N$  выполняется структурный синтез ДУ. На этом этапе каждое состояние автомата заменяется булевым вектором, размерность которого  $k$  должна удовлетворять неравенству  $2^k \geq |S|$ . Это означает, что каждому состоянию автомата  $A$  (модели) ставится в соответствие взаимно-однозначное отображение  $\rho$  в вектор размерности  $k$ , то есть некоторое состояние схемы  $N$  с памятью. Каждый элемент вектора представляет собой двоичный кортеж длины  $k$ -значения выходов триггеров автомата  $A$  (код состояния), а сам автомат является асинхронным и в общем случае частично заданным (не полностью определенным).

Далее в статье в качестве математической модели нечеткого автомата  $G$  используется детерминированный (четкий) граф  $\tilde{G}$ , полученный из  $G$  путем описанного выше преобразования.

Понятно, что граф  $\tilde{G}$ , представляющий модель упомянутого конечного асинхронного автомата  $A$ , полностью моделируется [4] реальной схемой, построенной в результате описанного упомянутого преобразования. Поэтому в нем множество траекторий движения автомата по состояниям четкого графа содержит все варианты траекторий, имеющих в исходном графе нечеткого автомата.

Пусть для автомата  $A$  используются следующие обозначения:  $S$  — множество состояний,  $X$  — входной алфавит,  $\delta : S \cdot X \rightarrow S$  — функция переходов. В [2, гл. 59] введены понятия устойчивого состояния, состязаний (гонок) элементов памяти, опасных и неопасных, которые далее используются. Поскольку задача противогоночного кодирования оперирует только с кодами состояний, то функция выходов автомата не приводится.

Следуя [2, гл. 59], обозначим через  $U(s_i, s_j)$  множество всех возможных промежуточных состояний, в которые автомат может попасть при переходе  $s_i \rightarrow s_j$ .

Два перехода  $s_i \rightarrow s_j$  и  $s_k \rightarrow s_l$ , выполняемые при одном и том же сигнале в различные состояния  $(s_j \neq s_l)$ , называются парой переходов. В [2, гл. 59] приведены условия отсутствия опасных состязаний. Формулировка этих условий приводится ниже дословно в виде утверждения из упомянутой монографии.

**Утверждение 59.1.** При одновременном возбуждении элементов памяти в процессе перехода опасные состязания отсутствуют тогда и только тогда, когда для каждой пары переходов  $s_i \rightarrow s_j$  и  $s_k \rightarrow s_l$  имеет место  $U(s_i, s_j) \cap U(s_k, s_l) = \emptyset$ .

### 4. Пример поиска состязаний сигналов в нечетких автоматах

Выше было показано, что для нечеткого автомата, полученного после преобразования его в соответствующий четкий автомат, могут использоваться методы противогоночного кодирования, разработанные для четких автоматов. В качестве иллюстрации рассмотрим задачу поиска упомянутого кодирования нечеткого автомата, четкая модель которого приведена на рис. 6.

Поскольку у автомата  $|S| = 7$ , начнем кодирование, используя кортежи длины  $k = 3$ . Пусть

множество входных сигналов  $X$  содержит 3 входных сигнала ( $x_1, x_2, x_3$ ) и обычно они расположены над дугами графа автомата, например, как это сделано на рис. 6.

Для поиска противогоночного кодирования может быть использован, например, универсальный метод кодирования [3], позволяющий путем введения дополнительных состояний решать задачу кодирования для любого асинхронного автомата. При его использовании для кодирования состояний автомата с  $m$  состояниями требуется  $m$  элементов памяти. Поэтому на практике часто используются другие, более эффективные по требуемой памяти методы. Один из них — эвристический метод развязывания пар [3]. Для рассматриваемого примера воспользуемся им и проиллюстрируем его применение.

Введем два определения. Пусть  $(\alpha, \beta)$  и  $(\gamma, \varepsilon)$  — две пары переходов между состояниями, закодированные двоичными кодами длины  $k$ , пары назовем развязанными, если при некотором  $i$  из отрезка  $[1, k]$   $i$ -й разряд кода принимает значение 1 на паре  $(\alpha, \beta)$  и значение 0 на паре  $(\gamma, \varepsilon)$ . Если это не выполняется, то эти пары назовем связанными.

Алгоритм противогоночного кодирования базируется на Утверждении 59.1 [2]. Идея алгоритма проста: находятся все пары переходов, имеющие хотя бы один общий входной сигнал, который инициирует эти переходы. На найденных парах выполняется присваивание разрядам их кодов таких значений, чтобы они стали развязанными.

Пусть пары переходов обозначены как  $\alpha \rightarrow \beta$  и  $\gamma \rightarrow \varepsilon$ , а значения  $\alpha\beta\gamma\varepsilon$  соответственно четверка их состояний. Литеры в четверке кодов могут принимать значения 0, 1, \* (символ неопределенности).

Опишем кратко алгоритм развязывания пар. Он последовательно проверяет все пары переходов. Если при некотором  $r \in [1, k]$  значения  $r$ -го разряда кодов образует наборы 0011 или 1100, пара считается развязанной, после чего переходим к следующей паре. При появлении четверки  $\alpha\beta\gamma\varepsilon$ , содержащей символы \* (от одного до четырех), доопределяются эти неопределенные значения так, чтобы получился один из наборов 0011 или 1100. Затем осуществляется просмотр очередной пары.

В результате развязывания пар переходов длина кода может оказаться не минимальной. При введении нового разряда могут появиться развязанные пары, которые уже были развязаны раньше. Поэтому этот алгоритм кодирования требует минимизации длины получаемых кодов состояний. Она может быть реализована, например, следующим образом: исключается один из разрядов кодов, после чего некоторые пары могут оказаться связанными. Затем применим алгоритм развязывания и исключаем еще один разряд. Такие действия производятся до тех пор, пока длина кода не перестанет уменьшаться. Если в результате работы алгоритма некоторые значения разрядов окажутся неопределенными, то их можно доопределить произвольным образом.

Для поиска состязаний сигналов в четком графе  $\tilde{G}$  нужно знать таблицу переходов соответствующего автомата. В графе  $\tilde{G}$  указываются переходы между состояниями автомата, но в общем случае в нем может быть задано только некоторое их подмножество. В этом случае для формирования таблицы переходов необходимо определить недостающие переходы (некоторые элементы таблицы).

Известно, что для асинхронного автомата выход есть функция устойчивого состояния, т. е. такой автомат является частичным. Исследуемая здесь задача предполагает устойчивость таблицы переходов [4]. Последнее означает, что для любого элемента таблицы (он задает некоторый переход) должна существовать цепочка переходов, заканчивающаяся устойчивым состоянием. Если автомат задан в виде графа, то для любого входного символа  $x$  и любой вершины в графе должен существовать путь, в котором все дуги помечены символом  $x$ , в вершину с петлей, помеченной тем же символом  $x$ . Такая таблица называется нормальной. Далее потребуем, чтобы при любом входном сигнале автомат сразу попадал в устойчивое состояние, что обеспечивает нормальность таблицы переходов.

Предполагается, что задаются входные сигналы, дающие петли, для всех состояний:  $x_1 - s_4, x_3 - s_5$  (такие сигналы для  $s_1, s_2, s_3$  приведены на рис. 1. Поскольку прочерк в клетке таблицы означает отсутствие перехода, тогда как следствие получаем  $x_1 - s_6, x_3 - s_7$ . Таким



образом, таблица переходов является нормальной.

Отметим, что некоторые сигналы явно указываются при задании графа  $G$  (например, упорядоченные сигналы  $x_2, x_3$  в переходе  $s_2 \rightarrow \{s_3, s_5\}$  на рис. 1, что делает однозначным заполнение столбца таблицы для  $s_2$ . Информация о части сигналов может задаваться в качестве дополнительной. В рассматриваемом примере в множествах  $T(s_2), T(s_3), T(s_5)$  будем считать явно указанными только входные сигналы  $x_2, x_3, x_1$  для переходов к состояниям  $s_3, s_7, s_6$  (но не все множество преемников). Приведенная информация позволяет сформировать таблицу переходов (табл. 1) для рассматриваемого примера.

**Таблица 1. Таблица переходов автомата  $\tilde{G}$**

$x$	$s$						
	$s_1$	$s_2$	$s_3$	$s_4$	$s_5$	$s_6$	$s_7$
$x_1$	$s_2$	$s_2$	$s_4$	$s_4$	$s_6$	$s_6$	–
$x_2$	$s_1$	$s_3$	$s_3$	$s_1$	$s_3$	–	–
$x_3$	–	$s_5$	$s_7$	–	$s_5$	–	$s_7$

Очевидно, что описать единый алгоритм определения части неизвестных переходов в таблице в общем случае невозможно. Способ определения таких сигналов связан с особенностями каждого конкретного автомата (его графа). Понятно также, что разрешимость этой задачи может потребовать некоторой дополнительной информации.

Используя табл. 1, построим три совокупности пар переходов состояний  $W(x_1), W(x_2), W(x_3)$ , появляющихся при подаче соответствующих входных сигналов:

$$W(x_1) = ((s_1, s_2), (s_2, s_2), (s_3, s_4), (s_4, s_4), (s_5, s_6), (s_6, s_6)),$$

$$W(x_2) = ((s_1, s_1), (s_2, s_3), (s_3, s_3), (s_4, s_1), (s_5, s_3)),$$

$$W(x_3) = ((s_2, s_5), (s_3, s_7), (s_5, s_5), (s_7, s_7)).$$

Начнем последовательный просмотр различных пар переходов в  $W(x_1)$  и будем их развязывать. Переходы  $(s_1, s_2)$  и  $(s_2, s_2)$  развязаны, поскольку состояния  $s_2$  у них совпадают. Пары переходов  $(s_1, s_2)$  и  $(s_3, s_4)$  развязываем путем

введения переменной  $d_1$  (табл. 2). Пары переходов  $(s_1, s_2)$  и  $(s_4, s_4)$  при этом окажутся тоже развязанными, поскольку им соответствует четверка 0011. Паре переходов  $(s_1, s_2)$  и  $(s_5, s_6)$  теперь соответствует четверка 00\*\* и если положить  $d_1 = 1$ , то эта пара также становится развязанной (табл. 3). Легко проверить, что с учетом значений  $d_1$  в табл. 3 из всех возможных пар переходов, образованных из элементов  $W(x_1)$ , окажется не развязанной только одна пара переходов  $(s_3, s_4)$  и  $(s_5, s_6)$ . Чтобы развязать ее, введем еще одну переменную  $d_2$  со значениями, приведенными в табл. 4.

По аналогии с изложенным получим табл. 5 и табл. 6 для совокупностей состояний в  $W(x_2), W(x_3)$ .

**Таблица 2. Развязывание пар состояний  $(s_1, s_2), (s_3, s_4)$**

$s$	$d_1$
$s_1$	0
$s_2$	0
$s_3$	1
$s_4$	1
$s_5$	*
$s_6$	*
$s_7$	*

**Таблица 3. Развязывание пар состояний  $(s_1, s_2), (s_5, s_6)$**

$s$	$d_1$
$s_1$	0
$s_2$	0
$s_3$	1
$s_4$	1
$s_5$	1
$s_6$	1
$s_7$	*

**Таблица 4. Развязывание пар состояний  $(s_3, s_4), (s_5, s_6)$**

$s$	$d_1$	$d_2$
$s_1$	0	*
$s_2$	0	*
$s_3$	1	0
$s_4$	1	0
$s_5$	1	1
$s_6$	1	1
$s_7$	*	*

Табл. 6 не является минимальной, и ее можно минимизировать способом, который был упомянут выше. Будем последовательно исключать столбцы из нее, что может привести к появлению неразвязанных пар. В этом случае потребуется произвести развязывания и делается попытка исключить еще один столбец из таблицы. Это делается до тех пор, пока длина кода не уменьшается.

Здесь не приводятся промежуточные результаты минимизации. Заметим только, что в табл. 6 удалось исключить два столбца  $d_1$  и  $d_2$ , в результате была получена табл. 7, содержащая только 3 столбца для переменных  $d_3, d_4, d_5$ . В ней еще остались символы неопределенных значений в кодах некоторых состояний, которые после соответствующего доопределения дают табл. 8 противогоночного кодирования.

**Заключение**

В последние годы нечеткие автоматы используются в качестве моделей дискретных устройств различного назначения. Укажем лишь несколько публикаций в области контроля и диагностики устройств, например,

[11–15], близких к научным интересам авторов статьи, где эти модели нашли приложения. Указанные статьи посвящены разработке метода синтеза тестов для нечетких автоматов с конечной памятью, тестированию систем с нечеткими компонентами, функциональному контролю линейных динамических систем методом обращения, тестированию нечетких линейных автоматов, разработке метода обращения выходов в нечетких дискретных системах.

В статье предложена концепция получения методов противогоночного кодирования нечетких автоматов. Она состоит в преобразовании нечеткого графа, задающего нечеткий автомат, в некоторый четкий граф. Дано описание способа такого преобразования. Показано, что все состязания, возникающие на полученном четком графе, содержат в качестве подмножества все возможные состязания, которые имеют место в нечетком графе нечеткого автомата. Поскольку для четких автоматов разработан ряд эффективных методов поиска состязаний сигналов, все они применимы для решения упомянутой задачи противогоночного кодирования для нечетких автоматов.

**Таблица 5. Развязывание пар состояний в  $W(x_2)$**

$s$	$d_1$	$d_2$	$d_3$
$s_1$	0	1	1
$s_2$	0	0	0
$s_3$	1	0	0
$s_4$	1	0	1
$s_5$	1	1	0
$s_6$	1	1	*
$s_7$	*	*	*

**Таблица 6. Развязывание пар состояний в  $W(x_3)$**

$s$	$d_1$	$d_2$	$d_3$	$d_4$
$s_1$	0	1	1	*
$s_2$	0	0	0	0
$s_3$	1	0	0	1
$s_4$	1	0	1	*
$s_5$	1	1	0	0
$s_6$	1	1	*	*
$s_7$	*	*	*	1

**Таблица 7. Развязывание противогоночного кодирования**

$s$	$d_3$	$d_4$	$d_5$
$s_1$	1	*	0
$s_2$	0	0	0
$s_3$	0	1	*
$s_4$	1	*	*
$s_5$	0	0	1
$s_6$	*	*	1
$s_7$	*	1	1

**Таблица 8. Таблица кодирования после доопределения символов \***

$s$	$d_3$	$d_4$	$d_5$
$s_1$	1	1	0
$s_2$	0	0	0
$s_3$	0	1	1
$s_4$	1	1	1
$s_5$	0	0	1
$s_6$	1	0	1
$s_7$	0	1	1

Заметим, что платой за сведение задачи для нечетких автоматов к той же задаче для четкого автомата являются затраты на преобразование нечеткого графа нечеткого автомата в соответствующий четкий, на формирование таблицы переходов четкого автомата. Кроме того, мощность множества траекторий в четком графе, в котором ведется поиск составных, заведомо больше мощности множества возможных траекторий в нечетком графе нечеткого автомата, что делает поиск более затратным. ▲

### Библиографический список

1. Сагалович Ю. Л. Метод повышения надежности конечного автомата / Ю. Л. Сагалович // Проблемы передачи информации. 1965. Т. 1, вып. 1. С. 27–35.
2. Закревский А. Д., Поттосин Ю. В., Черемисинова Л. Д. Логические основы проектирования дискретных устройств / А. Д. Закревский, Ю. В. Поттосин, Л. Д. Черемисинова. М.: ФИЗМАТЛИТ, 2007. 592 с.
3. Фомичев В. С. Формальные языки, грамматики и автоматы: / Курс лекций [Сайт] / СПб. 2006. URL: [https://old.etu.ru/misc/LGA\\_2007\\_FINAL/Index/html](https://old.etu.ru/misc/LGA_2007_FINAL/Index/html).
4. Буркатовская Л. И., Буркатовская Ю. Б. Логическое проектирование дискретных устройств / Л. И. Буркатовская, Ю. Б. Буркатовская. Томск: Томский госуниверситет. 2011. 172 с.
5. Zadeh L. A. (1965) Fuzzy sets. *Information and Control*. Vol. 8, iss. 3. P. 338–353.
6. Кофман А. Введение в теорию нечетких множеств / А. Кофман. М.: Радио и связь. 1982. 432 с.
7. Сперанский Д. В. Эксперименты с нечеткими автоматами / Д. В. Сперанский // Автоматика и телемеханика. 2015. № 2. С. 107–124. EDN: TOBFFF.
8. Santos E. Maximin automata / E. Santos // *Information and Control*. 1968. Vol. 13. P. 363–377.
9. Topencharov V., Stoeva S. Fuzzy-topological automata / V. Topencharov, S. Stoeva // *Fuzzy Sets and Systems*. 1985. Vol. 16, iss. 1. P. 65–74.
10. Reyneri L. M. An Introduction to Fuzzy State Automata. *Biological and Artificial Computation: From Neuroscience to Technology* / L. M. Reyneri // *Lecture Notes in Computer Science*. 1997. Vol. 1240. P. 273–283.
11. Сперанский Д. В. Синтез обнаруживающих тестов для нечетких автоматов с конечной памятью / Д. В. Сперанский // Вестник Томского государственного университета. Управление, вычислительная техника и информатика. 2024. № 66. С. 120–127. DOI: 10.17223/19988605/66/12. EDN: PLAEMB.
12. Сперанский Д. В. О задаче обращения выходов нечетких дискретных систем / Д. В. Сперанский // Известия Саратовского университета. Новая серия. Серия: Математика. Механика. Информатика. 2022. Т. 22, № 1. С. 112–122. DOI: 10.18500/1816-9791-2022-22-1-112-122. EDN: VMOPOL.
13. Горелик А. В., Журавлев И. А., Орлов А. В. и др. Тестирование систем с нечеткими дискретными компонентами / А. В. Горелик, И. А. Журавлев, А. В. Орлов, Д. В. Сперанский // Автоматика на транспорте. 2020. Т. 6, № 4. С. 518–531. DOI: 10.20295/2412-9186-2020-6-4-518-531. EDN: ZXTJIM.
14. Сперанский Д. В. Тестирование нечетких линейных автоматов / Д. В. Сперанский // Известия Саратовского университета. Новая серия. Серия: Математика. Механика. Информатика. 2019. Т. 19, № 2. С. 233–240. DOI: 10.18500/1816-9791-2019-19-2-233-240. EDN: JPNAKG
15. Сперанский Д. В. Синтез тестов для нечетких линейных автоматов // Танаевские чтения: Доклады Восьмой Международной научной конференции — НАН Беларуси, Объединенный институт проблем информатики, Институт математики, Белорусский госуниверситет. 2018. Минск. С. 161–165.

*TRANSPORT AUTOMATION RESEARCH. 2024. Vol. 10, no. 2. P. 178–189*  
 DOI: 10.20295/2412-9186-2024-10-02-178-189

### Search for signaling races in fuzzy asynchronous automata

#### Information about authors

**Speranskiy D. V.**, Doctor in Engineering, Professor. E-mail: speranskiy.dv@gmail.com

**Lunev S. A.**, PhD in Engineering, Associate Professor. E-mail: slunev@mail.ru

Russian University of Transport (MIIT), Department of Transportation Infrastructure Management Systems, Moscow

**Abstract.** The object of study is discrete devices with memory, which describes the operation of such devices by using the mathematical model of fuzzy automaton. The fuzzy automaton model is proposed, which takes into account the specifics of the problem under consideration. In this model, the phenomenon of fuzziness is manifested in the transitions between the states of the fuzzy automaton. As in the case of a classical automaton, the problem is reduced to the counterhonking coding of fuzzy automaton states. The paper proposes a principle of construction of methods of such coding for fuzzy automaton. Its idea consists in transformation of a given fuzzy automaton in the form of a fuzzy graph into some special classical graph. A description of the procedure of such transformation is given. It is proved that all possible races on the fuzzy graph given fuzzy automaton, are a subset of the races occurring on the classical graph transformed by the mentioned procedure. Since an extensive arsenal of methods has been developed for finding signal races in classical automata, it can be applied to fuzzy automaton.

**Keywords:** reliability of electronic devices, signal races, models of fuzzy device, fuzzy automata and graphs, transformations of fuzzy graphs into classical ones, methods of counterhonking coding for fuzzy automaton.

#### References

1. Sagalovich Yu. L. Metod povysheniya nadezhnosti konechnogo avtomata / Yu. L. Sagalovich // Problemy peredachi informacii. 1965. T. 1, vyp. 1. S. 27–35. (In Russian)
2. Zakrevskij A. D., Pottosin Yu. V., Cheremisinova L. D. Logicheskie osnovy proektirovaniya diskretnyh ustrojstv / A. D. Zakrevskij, Yu. V. Pottosin, L. D. Cheremisinova. M.: FIZMATLIT. 2007. 592 s. (In Russian)
3. Fomichev V. S. Formal'nye yazyki, grammatiki i avtomaty / Kurs lekcij [Sajt] / SPb. 2006. URL: [https://old.etu.ru/misc/LGA\\_2007\\_FINAL/Index/html](https://old.etu.ru/misc/LGA_2007_FINAL/Index/html). (In Russian)
4. Burkatovskaya L. I., Burkatovskaya Yu. B. Logicheskoe proektirovanie diskretnyh ustrojstv / L. I. Burkatovskaya, Yu. B. Burkatovskaya. Tomsk, Tomskij gosuniversitet. 2011. 172 s. (In Russian)
5. Zadeh L. A. (1965) Fuzzy sets. Information and Control. Vol. 8, iss. 3. P. 338–353.
6. Kofman A. Vvedenie v teoriyu nechetkih mnozhestv / A. Kofman. M.: Radio i svyaz'. 1982. 432 s. (In Russian)
7. Speranskiy D. V. Eksperimenty s nechetkimi avtomatami / D. V. Speranskiy // Avtomatika i telemekhanika. 2015. № 2. S. 107–124. EDN: TOBFFF. (In Russian)
8. Santos E. Maximin automata / E. Santos // Information and Control. 1968. Vol. 13. P. 363–377.
9. Topencharov V., Stoeva S. Fuzzy-topological automata / V. Topencharov, S. Stoeva // Fuzzy Sets and Systems. 1985. Vol. 16, iss. 1. P. 65–74.
10. Reyneri L. M. An Introduction to Fuzzy State Automata. Biological and Artificial Computation: From Neuroscience to Technology / L. M. Reyneri // Lecture Notes in Computer Science. 1997. Vol. 1240. P. 273–283.
11. Speranskiy D. V. Sintez obnaruzhivayushchih testov dlya nechetkih avtomatov s konechnoj pamyat'yu / D. V. Speranskiy // Vestnik Tomskogo gosudarstvennogo universiteta. Upravlenie, vychislitel'naya tekhnika i informatika. 2024. № 66. S.120–127. DOI: 10.17223/19988605/66/12. EDN: PLAEMB. (In Russian)
12. Speranskiy D. V. O zadache obrashcheniya vyhodov nechetkih diskretnyh sistem / D. V. Speranskiy // Izvestiya Saratovskogo universiteta. Novaya seriya. Seriya: Matematika. Mekhanika. Informatika. 2022. T. 22, № 1. S. 112–122. DOI: 10.18500/1816-9791-2022-22-1-112-122. EDN: VMOPOL. (In Russian)
13. Gorelik A. V., Zhuravlev I. A., Orlov A. V. i dr. Testirovanie sistem s nechetkimi diskretnymi komponentami / A. V. Gorelik, I. A. Zhuravlev, A. V. Orlov, D. V. Speranskiy // Avtomatika na transporte. 2020. T. 6, № 4. S. 518–531. DOI: 10.20295/2412-9186-2020-6-4-518-531. EDN: ZXTJIM. (In Russian)
14. Speranskiy D. V. Testirovanie nechetkih linejnyh avtomatov / D. V. Speranskiy // Izvestiya Saratovskogo universiteta. Novaya seriya. Seriya: Matematika. Mekhanika. Informatika. 2019. T. 19, № 2. S. 233–240. DOI: 10.18500/1816-9791-2019-19-2-233-240. EDN: JPNACK (In Russian)
15. Speranskiy D. V. Sintez testov dlya nechetkih linejnyh avtomatov // Tanaevskie chteniya: Doklady Vos'moj Mezhdunarodnoj nauchnoj konferencii — NAN Belarusi, Ob"edinennyj institut problem informatiki, Institut matematiki, Belorusskij gosuniversitet. 2018. Minsk. S. 161–165. (In Russian)