

DOI: 10.15514/ISPRAS-2026-38(4)-16



Идентификация финального состояния в автоматах с таймаутами и временными ограничениями

¹ А.С. Твардовский, ORCID: 0000-0001-7705-7214 <tvardal@mail.ru>

² И.Б. Бурдонов, ORCID: 0000-0001-9539-7853 <igor@ispras.ru>

^{2,3} Н.В. Евтушенко, ORCID: 0000-0002-4006-1161 <evtushenko@ispras.ru>

¹ Национальный исследовательский Томский государственный университет,
634050, г. Томск, просп. Ленина, д. 36.

² Институт системного программирования РАН им. В.П. Иванникова,
109004, Россия, г. Москва, ул. А. Солженицына, д. 25.

³ Национальный исследовательский университет «Высшая школа экономики»,
101000, Россия, г. Москва, ул. Мясницкая, д. 20.

Аннотация. В статье вводятся понятия установочной и синхронизирующей последовательностей для конечных автоматов с временными ограничениями и таймаутами. Такие последовательности широко используются для идентификации финального состояния исследуемого автомата как при синтезе тестов на основе формальных моделей, так и при обучении конечно-автоматных моделей. Метод синтеза установочных и синхронизирующих последовательностей основан на использовании конечно-автоматной абстракции временного автомата, поскольку методы синтеза таких идентификационных последовательностей по конечно-автоматной модели хорошо изучены.

Ключевые слова: конечный автомат; временные ограничения; таймауты; установочная последовательность; синхронизирующая последовательность; конечно-автоматная абстракция.

Для цитирования: Твардовский А.С., Бурдонов И.Б., Евтушенко Н.В. Идентификация финального состояния в автоматах с таймаутами и временными ограничениями. Труды ИСП РАН, 2026, том 38, вып. 4, часть 2, стр. 7–22. DOI: 10.15514/ISPRAS-2026-38(4)-16.

Final state identification of Finite State Machines with timeouts and timed guards

¹ A.S. Tvardovskii, ORCID: 0000-0001-7705-7214 <tvardal@mail.ru>

² I.B. Burdonov ORCID: 0000-0001-9539-7853 <igor@ispras.ru>

^{2,3} N.V. Yevtushenko ORCID: 0000-0002-4006-1161 <evtushenko@ispras.ru>

¹ National Research Tomsk State University,
36, Lenin ave., Tomsk, 634050, Russia.

² Institute for System Programming of the Russian Academy of Sciences,
25, Alexander Solzhenitsyn st., Moscow, 109004, Russia.

³ National Research University Higher School of Economics,
20, Myasnitskaya st., Moscow, 101000, Russia.

Abstract. Homing and synchronizing sequences for Finite State Machines (FSM) with timeouts and timed guards are introduced. Such sequences are widely used in model based testing for test derivation as well as for FSM learning. Existence check and derivation of such sequences are based on a corresponding FSM abstraction of a timed machine, since such methods are well studied for classical FSMs.

Keywords: finite state machine (FSM); timed guards; timeouts; homing sequence; synchronizing sequence; FSM abstraction.

For citation: Tvardovskii A.S., Burdonov I.B., Yevtushenko N.V. State identification of Finite State Machines with timeouts and timed guards. Trudy ISP RAN/Proc. ISP RAS, vol. 38, issue 4, part 2, 2026, pp. 7-22 (in Russian). DOI: 10.15514/ISPRAS-2026-38(4)-16.

1. Введение

Фундаментальные исследования проблем идентификации состояний в автоматных моделях ведутся с середины 20-го века [1-7]. Синхронизирующие и установочные последовательности позволяют определить состояние, достигнутое системой после подачи такой последовательности или после наблюдения реакции системы на такую последовательность, что позволит более эффективно проводить дальнейший анализ поведения системы. Полученные результаты используются при тестировании и верификации (компонентов) программного и аппаратного обеспечения (см., например, [8-12], а также при обучении автоматных моделей по заданному множеству трасс [13, 14, 15].

Для различных классов классических конечных автоматов определены необходимые и достаточные условия существования безусловных и условных (адаптивных) установочных и синхронизирующих последовательностей и предложены алгоритмы их синтеза, если такая последовательность существует. Безусловная последовательность представляет собой известную до начала эксперимента последовательность входных воздействий, в то время как для адаптивной последовательности следующее входное воздействие зависит от реакции автомата (системы) на предыдущие входные воздействия. Для некоторых классов автоматов проведение адаптивного эксперимента позволяет существенно сократить верхнюю оценку длины установочной или синхронизирующей последовательности в [7, 16], но может быть осложнено необходимостью организации обратной связи. Также известно, что для недетерминированных конечных автоматов вероятность существования адаптивных последовательностей для идентификации состояний выше, чем вероятность существования безусловных последовательностей.

Для идентификации состояния современных систем, часто приходится учитывать временные аспекты в их поведении, такие как, например, таймауты в телекоммуникационных протоколах, предписывающие серверу разорвать соединение при отсутствии ответа от клиента в течение определенного времени. Существуют различные подходы к внесению

временных аспектов в модель конечного автомата, в зависимости от использования одной временной переменной или множества таковых, использования входных и/или выходных таймаутов, входных и выходных временных ограничений [17-20]. В настоящей работе мы обсуждаем только модели с одной временной переменной, поскольку такое ограничение, с одной стороны, упрощает постановку и решение задачи построения идентификационных последовательностей, а с другой, достаточно часто согласуется со свойствами реальных систем.

Вопросы построения установочных последовательностей для временных автоматов ранее рассматривались авторами для автоматов с временными ограничениями [21] и для автоматов с входными таймаутами [22], где для построения таких последовательностей использовались конечно-автоматные абстракции. Данная работа, в некотором смысле, является обобщением этих результатов и посвящена задаче проверки существования и синтеза безусловных установочных и синхронизирующих последовательностей для автоматов, которые имеют и таймауты, и временные ограничения. Поскольку в автоматах с входными таймаутами автомат может переходить из состояния в состояние без подачи входного воздействия, мы также обсуждаем классификацию установочных и синхронизирующих последовательностей относительно свойств достигнутых финальных состояний.

Содержание статьи следующее. Второй раздел содержит определение автомата с входными таймаутами и временными ограничениями. В разделе 3 представлена предлагаемая в работе классификация установочных и синхронизирующих последовательностей для временных автоматов. Раздел 4 содержит определение конечно-автоматной абстракции, которая ориентирована на синтез идентификационных последовательностей и, вообще говоря, несколько отличается от конечно-автоматных абстракций, рассмотренных в предыдущих работах авторов. В разделе 5 представлены методы синтеза безусловных установочных и синхронизирующих последовательностей.

2. Основные определения и обозначения

Настоящий раздел содержит основные определения и обозначения, используемые в статье.

2.1 Временные конечные автоматы

Автоматом с (входными) таймаутами и временными ограничениями (далее, для краткости *временной автомат* или просто *автомат*, если из контекста понятно, что речь идет о временном автомате) называется пятёрка $S = (S, S_{in}, I, O, \lambda_S, \Delta_S)$, где

- S, I и O конечные непустые множества состояний, входных и выходных символов;
- S_{in} – множество начальных состояний;
- $\lambda_S \subseteq S \times I \times O \times S \times \Pi$ – отношение переходов;
- Π – множество непустых интервалов $\langle a, b \rangle \subseteq [0; \infty)$, где символ « $\langle$ » выбирается из множества $\{[, (, \text{ а символ « $\rangle$ » – из множества $\{],), \}$;$
- Δ_S – функция таймаутов.

Временной автомат содержит вещественную временную переменную, которая отражает время, проведённое автоматом в текущем состоянии, т.е. в переменной постоянно накапливается соответствующее значение, которое «сбрасывается» в 0 при выполнении перехода.

Функция таймаутов $\Delta_S: S \rightarrow S \times (\mathbb{N} \cup \{\infty\})$, где $\mathbb{N}$ – множество целых положительных чисел, отражает допустимое время на ожидание входных символов в текущем состоянии. Если $\Delta_S(s) = (s', T)$, и входной символ не был подан в состоянии s в течение T единиц времени (тактов), то автомат переходит в состояние s' и значение временной переменной «сбрасывается» в 0.

Если $\Delta_S(s) = (s, \infty)$, то автомат может находиться в текущем состоянии сколь угодно долго до получения следующего входного символа.

Входные *временные интервалы*, далее просто временные интервалы, отражают диапазон значений временной переменной, допустимых для выполнения перехода. Переход $(s, i, o, s', g) \in S \times I \times O \times S \times \Pi$ описывает следующую ситуацию: входной символ i принимается автоматом S в состоянии s , когда временная переменная имеет значение $t \in g$; автомат S переходит в состояние s' , производит выходной символ o и значение временной переменной «сбрасывается» в 0. Для каждого состояния $s \in S$ с бесконечным таймаутом вводится неотрицательный целочисленный параметр

$$B_S(s): B_S(s) = \min \{x \mid \forall (s, i, o, s', \langle a, b \rangle) \in \lambda_S (\langle a, b \rangle \cap [x, \infty) = \emptyset) \vee \langle a, b \rangle \cap [x, \infty) = [x, \infty)\}.$$

Иными словами, значение $B_S(s)$ равно наибольшему целому числу, такому что для любого $t > B_S(s)$, t содержится только во временных интервалах переходов в состоянии s , верхней границей которых является ∞ . Если $S_{in} = S$, то автомат называется *неинициальным*, если $|S_{in}| = 1$, то автомат называется *инициальным*, иначе автомат называется *слабо-инициальным*.

Под *временным состоянием* временного автомата S будем понимать пару

$$(s, x) \in S_T = \{(s, x) : (s, x) \in S \times (\mathbb{R}^+ \cup \{0\}) \ \& \ x < T_s \ \& \ \Delta_S(s) = (s', T_s)\},$$

то есть временное состояние – это пара, состоящая из текущего состояния автомата и текущего значения его временной переменной, которое не может превышать таймаут в данном состоянии. Если таймаут в состоянии s равен ∞ , то x в паре (s, x) может быть любым неотрицательным числом.

Аналогично [20, 22], для каждого временного состояния (s, x) и задержки t существует *временной переход* $(s, x) \xrightarrow{t} (s', x')$, где (s', x') – временное состояние, достигнутое автоматом за время t из состояния (s, x) , если при этом на автомат не поступило ни одного входного символа. Для каждого временного состояния (s, x) и вход-выходной пары i/o существует вход-выходной переход $(s, x) \xrightarrow{i/o} (s', 0)$, если $(s, i, o, s', g) \in \lambda_S$ и $x \in g$.

Временной автомат S называется *полностью определённым*, если для каждой пары $((s, x), i) \in S_T \times I$ существует переход $(s, i, o, s', g) \in \lambda_S$, такой что $x \in g$, иначе S называется *частичным*.

Временной автомат S называется *детерминированным*, если для каждой пары переходов $(s, i, o_1, s_1, g_1), (s, i, o_2, s_2, g_2) \in \lambda_S$, где $o_1 \neq o_2$ или $s_1 \neq s_2$, справедливо, что $g_1 \cap g_2 = \emptyset$, иначе S называется *недетерминированным*.

Временной автомат S называется *наблюдаемым*, если для каждой пары переходов $(s, i, o, s_1, g_1), (s, i, o, s_2, g_2) \in \lambda_S$, где $g_1 \cap g_2 \neq \emptyset$, справедливо, что $s_1 = s_2$, иначе S называется *ненаблюдаемым*. По определению, любой детерминированный автомат является наблюдаемым. В настоящей работе мы рассматриваем задачи идентификации состояний для полностью определённых наблюдаемых временных автоматов, т.е. автоматов, для которых в каждом временном состоянии существует хотя бы один переход по каждому входному символу, и, если переходов по одному входному символу несколько, то следующее состояние может быть однозначно определено по соответствующей выходной реакции.

2.2 Временные трассы и временные вход-выходные последовательности

Последовательность временных и вход-выходных переходов, первый из которых начинается во временном состоянии (s_0, x_0) , а каждый последующий переход начинается в финальном состоянии предыдущего, называется *временной трассой* автомата S в состоянии (s_0, x_0) . Таким образом, временная трасса во временном состоянии (s_0, x_0) может быть представлена как $tr = (s_0, x_0) \xrightarrow{t_1} (s'_0, x'_0) \xrightarrow{i_1 o_1} (s_1, 0) \xrightarrow{t_2} (s'_1, x'_1) \xrightarrow{i_2 o_2} \dots \dots \dots \xrightarrow{t_n} (s'_{n-1}, x'_{n-1}) \xrightarrow{i_n o_n} (s_n, 0) \xrightarrow{t_{n+1}} (s'_n, x'_n)$. Аналогично классическому автомату, временной автомат может рассматриваться как преобразователь входных последовательностей в выходные последовательности (transducer), но поскольку реакция автомата зависит от

продолжительности ожидания между подачей предыдущего и следующего входных символов, то вводится понятие временного входного символа.

Временным входным символом назовём пару (i, t) , где $i \in I$ и $t \geq 0$ – вещественное значение момента времени, в который поступил входной символ i ; при этом предполагается, что эксперимент был начат в момент времени $t_0 = 0$, от которого отсчитывается время подачи входных воздействий. Последовательность временных входных символов $\alpha = (i_1, t_1) \dots (i_n, t_n)$, где $t_1 < \dots < t_n$, называется *временной входной последовательностью*.

Пусть временной автомат находится в состоянии (s, x) , где x – время, прошедшее с момента перехода в состояние s . Временной входной символ (i, t) индуцирует во временном состоянии (s, x) трассу $(s, x) \xrightarrow{t} (s', x') \xrightarrow{t} (s'', 0)$, состоящую из временного перехода (ожидания следующего входного символа в течение t тактов) и вход-выходного перехода. Для автомата на рис. 1 входной символ $(x, 1.5)$ индуцирует во временном состоянии $(s_1, 0)$ трассу $(s_1, 0) \xrightarrow{1.5} (s_2, 0.5) \xrightarrow{x_0} (s_1, 0)$, в то время как во временном состоянии $(s_1, 0.5)$ этот входной символ индуцирует трассу $(s_1, 0.5) \xrightarrow{1.5} (s_2, 1) \xrightarrow{x_0} (s_1, 0)$.

Отметим, что формально данная трасса может быть представлена трассами с несколькими временными переходами $(s, x) \xrightarrow{t_1} \dots \xrightarrow{t_k} (s', x') \xrightarrow{t} (s'', 0)$, где $t_1 + \dots + t_k = t$, но далее, мы для простоты считаем, что каждый временной входной символ индуцирует один временной и один вход-выходной переход.

После подачи временной входной последовательности и выполнения временным автоматом S всех индуцированных ею переходов, S может продолжить смену состояний без подачи входных символов с использованием переходов по таймаутам. Поэтому понятие временной входной последовательности $(i_1, t_1) \dots (i_n, t_n)$ расширяется постфиксной временной задержкой $t \geq t_n$, которая определяет ожидание с момента подачи последнего входного символа t_n до окончания эксперимента в момент времени t . Подобная задержка может быть использована, например, для выполнения автоматом переходов по таймауту и достижения устойчивого состояния [20], из которого автомат не может выйти, пока не будет подан входной символ. Таким образом, под временной входной последовательностью также понимается последовательность $\alpha(t) = (i_1, t_1) \dots (i_n, t_n).(t)$. При отсутствии постфиксной задержки, т.е. при $t = t_n$, временная входная последовательность имеет вид $\alpha = (i_1, t_1) \dots (i_n, t_n)$.

Для временного автомата S временная входная последовательность $\alpha(t) = (i_1, t_1) \dots (i_n, t_n).(t)$ индуцирует временную трассу

$$tr = (s_0, x_0) \xrightarrow{t_1} (s'_0, x'_0) \xrightarrow{i_1 o_1} (s_1, 0) \xrightarrow{t_2 - t_1} (s'_1, x'_1) \xrightarrow{i_2 o_2} \dots \xrightarrow{t_n - t_{n-1}} (s'_{n-1}, x'_{n-1}) \xrightarrow{i_n o_n} (s_n, 0) \xrightarrow{t - t_n} (s'_{n+1}, x'_{n+1})$$

во временном состоянии (s_0, x_0) ; при этом состояние (s'_{n+1}, x'_{n+1}) называется *tr-преемником*. Последовательность выходных символов из вход-выходных переходов формирует выходную последовательность $\gamma = o_1 \dots o_n$, т.е. реакцию автомата S на временную входную последовательность $\alpha(t)$, и последовательность $\alpha/\gamma(t) = (i_1, t_1)/o_1 \dots (i_n, t_n)/o_n(t)$ называется *временной вход-выходной проекцией* или *ТЮ-проекцией* трассы tr . Отметим, что постфиксная задержка отражена во временной вход-выходной последовательности, поскольку продолжительность такой задержки влияет на преемника трассы с соответствующей ТЮ-проекцией.

Пример. Рассмотрим временной неинициальный автомат на рис. 1. Временная входная последовательность $\alpha(t) = (x, 1.5)(y, 1.6).(3.1)$ индуцирует во временном состоянии $(s_1, 0)$ трассу $tr = (s_1, 0) \xrightarrow{1.5} (s_2, 0.5) \xrightarrow{x_0} (s_1, 0) \xrightarrow{0.1} (s_1, 0.1) \xrightarrow{y_2} (s_1, 0) \xrightarrow{1.5} (s_2, 0.5)$, соответственно, $\gamma = 02$ является выходной реакцией автомата на $\alpha(t)$, а $(s_2, 0.5)$ – соответствующим *tr-преемником*.

Для недетерминированного временного автомата временная входная последовательность может индуцировать в одном временном состоянии несколько трасс с различными ТЮ-

проекциями и *tr-преемниками*. Тем не менее, для наблюдаемых автоматов справедливо следующее утверждение.

Утверждение 1. Если во временном состоянии (s, x) наблюдаемого временного автомата S входная последовательность $\alpha(t)$ индуцирует трассы tr_1 и tr_2 с одинаковыми ТЮ-проекциями, то tr_1 - и tr_2 -преемники совпадают.

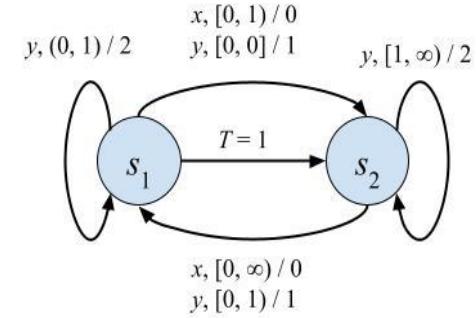


Рис. 1. Временной автомат S .
Fig. 1. Timed FSM S .

3. Установочные и синхронизирующие последовательности для временных автоматов

Мы обсуждаем установочные и синхронизирующие эксперименты с временным автоматом на основе подачи соответствующих идентификационных последовательностей, исходя из предположения, что перед их началом временная переменная исследуемого автомата была «сброшена» в 0, однако начальное состояние автомата не известно. Данное условие может быть легко выполнено подачей в начале эксперимента произвольного входного символа. Тем не менее, необходимо отметить, что при рассмотрении слабо-инициального временного автомата подача такого символа может привести к изменению множества начальных состояний S_{in} на множество S'_{in} . Такое «новое» множество начальных состояний может быть легко вычислено, и далее мы не будем акцентировать внимание на этом шаге.

Аналогично [22], мы выделяем подзадачу синтеза установочных и синхронизирующих последовательностей в стабильные состояния, а также вводим понятие поведенчески-стабильных состояний, поскольку, в отличие от автомата только с таймаутами, на поведение автомата с временными ограничениями влияет не только текущее состояние, но и конкретное значение временной переменной, определяя переход с соответствующим временным интервалом.

Пример. Рассмотрим вновь автомат на рис. 1. Последовательность $(x, 0).(2)$, позволяет определить достигнутое автоматом состояние s_2 , так как из состояния $(s_1, 0)$ автомат перейдет по ней в состояние $(s_2, 0)$, а из $(s_2, 0)$ – в $(s_2, 2)$. Состояние автомата s_2 не изменится до подачи следующего входного символа, но в достигнутом временном состоянии значение временной переменной может стать сколь угодно большим в зависимости от момента подачи следующего входного символа. Тем не менее, с точки зрения дальнейшего анализа поведения автомата, все достигнутые временные состояния, в которых значение временной переменной не меньше максимальной конечной границы временных интервалов $B_s(s_2) = 1$, неотличимы. Таким образом, временные состояния $\{(s_2, t): t \geq 1\}$ образуют класс поведенчески-стабильных (behavior-stable) состояний, и в ходе установочного эксперимента достаточным является определение принадлежности достигнутого (временного) состояния данному классу.

Состояние s временного автомата S называется *стабильным*, если $\Delta_S(s) = (s, \infty)$. Для стабильного состояния s временное состояние (s, x) временного автомата S называется *поведенчески-стабильным* (*b-стабильным* или *b-stable*), если $x > B_S(s)$. Таким образом, достижение автоматом стабильного состояния гарантирует, что автомат не изменит своё состояние под действием переходов по таймаутам, а достижение любого *b-стабильного* временного состояния гарантирует, что дальнейшее поведение автомата не зависит от времени нахождения автомата в текущем состоянии, т.е. дальнейшее поведение автомата одинаково во всех достижимых по временным переходам временных состояниях. Последнее, в частности, важно, если установочный эксперимент проводится с целью достижения автоматом некоторого известного состояния, в котором будут поданы другие, например, тестовые последовательности, и для экспериментатора желательно не быть «привязанным» к моменту достижения этого состояния.

Установочный эксперимент заключается в подаче на автомат входной последовательности, позволяющей определить достигнутое автоматом состояние на основе полученной выходной реакции. Временная вход-выходная последовательность $\alpha/\gamma(t)$ является *установочной* во временном автомате S , если для каждой пары трасс tr_1 и tr_2 во временных состояниях $(s_1, 0)$, $(s_2, 0)$, где $s_1, s_2 \in S_m$, с одинаковой ТПО-проекцией $\alpha/\gamma(t)$, tr_1 -преемник (s_1', x_1') и tr_2 -преемник (s_2', x_2') совпадают, либо являются *b-стабильными* состояниями и $s_1' = s_2'$. Временная входная последовательность $\alpha(t)$ называется *установочной* во временном автомате S , если ТПО-проекция любой трассы, индуцируемой $\alpha(t)$ в любом состоянии $(s, 0)$, $s \in S_m$, является установочной.

Синхронизирующий эксперимент заключается в подаче на автомат входной последовательности, переводящей автомат в известное состояние, независимо от начального состояния и полученной выходной последовательности. Временная входная последовательность $\alpha(t)$ называется *синхронизирующей* во временном автомате S , если для всех трасс $tr_1, \dots, tr_n$, и их tr_i -преемников $(s_1, x_1), \dots, (s_n, x_n)$, индуцируемых $\alpha(t)$ в состояниях из множества $\{(s, 0) : s \in S_m\}$, справедливо, что, либо $(s_i, x_i) = (s_j, x_j)$, либо (s_i, x_i) и (s_j, x_j) являются *b-стабильными* и $s_i = s_j$, $0 < i \leq n$, $0 < j \leq n$.

Отметим также, что определённые выше последовательности позволяют установить достигнутое автоматом временное состояние или класс *b-стабильных* состояний $\{(s, x) : x > B_S(s)\}$, т.е. установить состояние автомата с известным значением временной переменной или множество состояний, временная компонента которых не имеет значения для дальнейшего поведения. Далее, мы используем понятие *класса b-стабильности*, который определяется как множество всех *b-стабильных* временных состояний с одним и тем же состоянием автомата. На практике в ряде случаев достаточно определить достигнутое автоматом состояние и, в таком случае, мы говорим о *state-установочных* и *state-синхронизирующих* последовательностях, сокращенно об *s-установочных* и *s-синхронизирующих* последовательностях.

Временная вход-выходная последовательность $\alpha/\gamma(t)$ называется *s-установочной* в автомате S , если для каждой пары трасс tr_1 и tr_2 во временных состояниях $(s_1, 0)$, $(s_2, 0)$, где $s_1, s_2 \in S_m$, с одинаковой ТПО-проекцией $\alpha/\gamma(t)$, для tr_1 -преемника (s_1', x_1') и tr_2 -преемника (s_2', x_2') , справедливо, что $s_1' = s_2'$. Временная входная последовательность $\alpha(t)$ называется *s-установочной* во временном автомате S , если ТПО-проекция любой трассы, индуцируемой $\alpha(t)$ в любом состоянии $(s, 0)$, $s \in S_m$, является *s-установочной*.

Временная входная последовательность $\alpha(t)$ называется *s-синхронизирующей* в автомате S , если для всех трасс $tr_1, \dots, tr_n$, и их tr_i -преемников $(s_1, x_1), \dots, (s_n, x_n)$, индуцируемых $\alpha(t)$ в состояниях из множества $\{(s, 0) : s \in S_m\}$, справедливо, что $s_i = s_j$, $0 < i \leq n$, $0 < j \leq n$.

Отметим, если α является установочной (синхронизирующей) последовательностью во временном автомате S , то α является *s-установочной* (*s-синхронизирующей*) последовательностью в S . Обратное в общем случае не верно.

Данные выше определения подходят для полностью определенных детерминированных и наблюдаемых недетерминированных временных автоматов; для неинициальных автоматов вместо множества S_m рассматривается множество S .

После проведения установочного или синхронизирующего эксперимента, автомат может сменить своё (временное) состояние, в частности, вследствие переходов по таймаутам. В этом случае его дальнейшее поведение может стать неопределённым без информации о времени, прошедшем с окончания эксперимента.

Установочная (синхронизирующая) последовательность называется *стабильной*, если tr -преемник каждой индуцируемой ею трассы tr является *b-стабильным* временным состоянием. Соответственная *s-установочная* (*s-синхронизирующая*) последовательность называется *стабильной*, если для tr -преемника (s, x) каждой индуцируемой ею трассы tr , состояние s является стабильным.

Если $\alpha(t)$ является стабильной установочной (синхронизирующей) последовательностью во временном автомате S , то $\alpha(t)$ является стабильной *s-установочной* (*s-синхронизирующей*) последовательностью в S . Обратное в общем случае не верно.

Отметим также, что далее, говоря о соответствии между установочными и синхронизирующими последовательностями в различных типах автоматов, а также о проверке их существования и синтезе, мы имеем в виду безыбыточные последовательности, то есть такие, которые перестают быть установочными при сокращении постфиксной задержки или удалении последнего входного символа в случае отсутствия постфиксной задержки. Для последовательностей, которые не являются безыбыточными, сформулированные далее утверждения могут не выполняться, что подробнее обсуждается в последующих разделах.

Пример. Рассмотрим свойства некоторых установочных последовательностей для неинициального временного автомата S на рисунке 1. Последовательность $(\gamma, 0.5)$ является синхронизирующей и установочной последовательностью, но не является стабильной, поскольку завершается в состоянии s_1 с конечным таймаутом, и, соответственно, после одного такта времени автомат перейдет в состояние s_2 . Последовательность $\epsilon.(1)$, где ϵ пустая последовательность, предполагает ожидание одного такта времени без подачи входных символов, и не является установочной последовательностью, поскольку не позволяет идентифицировать достигнутые временные состояния, но является стабильной *s-установочной* последовательностью. Последовательность $\epsilon.(2)$ является стабильной установочной последовательностью, поскольку переводит автомат в известные *b-стабильные* временные состояния.

4. Переход от временного автомата к классическому при построении идентификационных последовательностей

Конечно-автоматная абстракция для автоматов с таймаутами и временными ограничениями [20] позволяет частично описать поведение временного автомата как поведение соответствующего конечного автомата и свести ряд задач идентификации состояний временного автомата к решению соответствующей задачи для его конечно-автоматной абстракции, т.е. к решению задачи в классической теории автоматов. В настоящем разделе мы модифицируем такую конечно-автоматную абстракцию для оптимизации построения установочных и синхронизирующих последовательностей.

4.1 Конечные автоматы

Конечный автомат есть пятерка $P = (P, P_{in}, I, O, h_P)$, где P, I и O – конечные непустые множества: P – множество состояний, $P_{in} \subseteq P$ – множество начальных состояний, I и O – множества входных и выходных символов соответственно, $h_P \subseteq (P \times I \times O \times P)$ – отношение переходов. Кортеж (p, i, o, p') представляет переход под воздействием входного символа i из текущего состояния p в состояние p' с выходным символом o . Автомат P называется полностью определенным и детерминированным, если для каждой пары $(p, i) \in P \times I$ существует ровно одна пара $(o, p') \in O \times P$ такая, что $(p, i, o, p') \in h_P$.

Вход-выходная последовательность $\alpha/\gamma = i_1/o_1, i_2/o_2, \dots, i_l/o_l$ существует в состоянии p_1 автомата P , если существуют переходы $(p_1, i_1, o_1, p_2), (p_2, i_2, o_2, p_3), \dots, (p_l, i_l, o_l, p_{l+1}) \in h_P$. Соответственно, l – длина последовательности α/γ , $\alpha = i_1, i_2, \dots, i_l$ – её входная проекция, и $\gamma = o_1, o_2, \dots, o_l$ – выходная проекция последовательности α/γ . Для полностью определенных автоматов с различными свойствами существуют методы построения установочных и синхронизирующих последовательностей, и подобно [21-22], при построении установочных и синхронизирующих последовательностей для временных автоматов мы используем соответствующие конечно-автоматные абстракции.

4.2 Конечно-автоматная абстракция

Для временного автомата $S = (S, S_{in}, I, O, \lambda_S, \Delta_S)$ конечно-автоматная абстракция представляет собой конечный автомат

$$A_S = (S_A, S_{A_{in}}, I_A, O_A, h_{A_S}), \text{ где } I_A = I \cup I_d \cup \{1\}, I_d = \{i' : i \in I\}, O_A = O \cup \{1\},$$

в котором входной/выходной символ 1/1 отражает увеличение значения временной переменной на один такт. Добавление к входному символу штриха ' отражает наличие дополнительной задержки в интервале (0, 1) перед его подачей, например, 0.5 или 0.99. Мы далее показываем, что таких входных символов достаточно для того, чтобы по конечно-автоматной абстракции можно было проверить наличие или отсутствие установочных и синхронизирующих последовательностей в исходном временном автомате и построить такие кратчайшие безызбыточные последовательности, если соответствующие последовательности существуют. Состояния абстракции содержат две компоненты, состояние временного автомата и число целых тактов времени, проведённых в текущем состоянии, т.е. $S_A = \{(s, d) : d \in \mathbb{N} \cup \{0\} \text{ и } \Delta_S(s) = (s', T_s) \text{ и } d < T_s < \infty\} \cup \{(s, d) : d \in \mathbb{N} \cup \{0\} \text{ и } \Delta_S(s) = (s, \infty) \text{ и } d \leq B_S(s) + 1\}$, $S_{A_{in}} = \{(s, 0) : s \in S_{in}\}$. Для состояния $(s, d) \in S_A$ конечно-автоматной абстракции A_S справедливо, что

$$\begin{aligned} ((s, d), i, o, (s', 0)) &\in h_{A_S}, \text{ если } (s, i, o, s', g) \in \lambda_S \text{ и } d \in g; \\ ((s, d), i', o, (s', 0)) &\in h_{A_S}, \text{ если } (s, i, o, s', g) \in \lambda_S \text{ и } (d; d+1) \subseteq g; \\ ((s, d), 1, 1, (s, d+1)) &\in h_{A_S} \text{ если } \Delta_S(s) = (s', T_s), T_s \neq \infty \text{ и } d < T_s - 1; \\ ((s, d), 1, 1, (s, d+1)) &\in h_{A_S}, \text{ если } \Delta_S(s) = (s, \infty), \text{ и } d \leq B_S(s); \\ ((s, T_s - 1), 1, 1, (s', 0)) &\in h_{A_S}, \text{ если } \Delta_S(s) = (s', T_s), T_s \neq \infty. \end{aligned}$$

Для состояния s временного автомата, такого, что $\Delta_S(s) = (s, \infty)$, если все интервалы переходов в состоянии s с нижней границей $B_S(s)$ закрыты слева, то $((s, B_S(s)), 1, 1, (s, B_S(s))) \in h_{A_S}$ и состояние $(s, B_S(s) + 1)$ удаляется из абстракции со всеми входящими переходами, иначе $((s, B_S(s) + 1), 1, 1, (s, B_S(s) + 1)) \in h_{A_S}$.

Таким образом, в отличие от конечно-автоматной абстракции в [22], в множестве состояний абстракции, рассматриваемой в настоящей работе, отсутствуют состояния с открытыми интервалами вида $(s, (a, a + 1))$, поскольку отражающие подобные задержки метки переместились во входной алфавит; соответственно, рассматривается абстракция с меньшим

числом состояний. С другой стороны, в отличие от [21], множество входных символов абстракции уменьшилось в силу отсутствия входных символов вида $(i, (a, a + 1))$ и $(i, [a, a])$. Таким образом, предлагаемую абстракцию можно рассматривать как компромисс между существующими подходами, и далее мы покажем, что задачи проверки существования и построения установочных и синхронизирующих последовательностей можно решить с использованием такой «сокращенной» конечно-автоматной абстракции.

По определению, подобно [21-22], введенная в этой статье конечно-автоматная абстракция наследует такие свойства временного автомата как полнота (частичность), (не-)детерминизм, (не-)наблюдаемость.

Временная входная последовательность $\alpha(t) = (i_1, t_1) \dots (i_n, t_n)(t)$ временного автомата S может быть переведена во входную последовательность α^{FSM} конечно-автоматной абстракции A_S следующим образом. Пусть $t_0 = 0$ и оператор $[t]$ возвращает целую часть положительного значения t , временной входной символ (i_k, t_k) преобразуется во входную последовательность $\alpha_k = 1^{[t_k - t_{k-1}]} i_k$, если $(t_k - t_{k-1}) - [t_k - t_{k-1}] = 0$, или в $\alpha_k = 1^{[t_k - t_{k-1}]} i'_k$, если $(t_k - t_{k-1}) - [t_k - t_{k-1}] > 0$. Далее, абстрактная входная последовательность формируется как $\alpha^{FSM} = \alpha_1 \alpha_2 \dots \alpha_n 1^{[t - t_n]}$, а соответствующая выходная последовательность имеет вид $\gamma^{FSM} = 1^{[t_1]} o_1 1^{[t_2 - t_1]} o_2 \dots 1^{[t_n - t_{n-1}]} o_n 1^{[t - t_n]}$, где $\gamma = o_1 o_2 \dots o_n$ выходная реакция временного автомата на α . Входная последовательность $\alpha^{FSM} = 1^{T_1} i_1 1^{T_2} i_2 \dots 1^{T_n} i_n 1^T$ конечно-автоматной абстракции A_S может быть преобразована в соответствующую последовательность $\alpha.(t) = (i_1, t_1)(i_2, t_2) \dots (i_n, t_n).(t)$ временного автомата S , построенную следующим образом:

$$\begin{aligned} \text{если } i_1 \in I_d, \text{ то } t_1 &= T_1 + \theta, 0 < \theta < 1, \text{ иначе } t_1 = T_1; \\ \text{если } i_k \in I_d, \text{ то } t_k &= t_{k-1} + T_k + \theta, 0 < \theta < 1, \text{ иначе } t_k = t_{k-1} + T_k, 1 < k \leq n; \\ t &= t_n + T. \end{aligned}$$

Соответствующая выходная последовательность γ получается удалением всех символов 1 из γ^{FSM} .

Отметим, что представленное выше моделирование временных трасс и соответствующих вход-выходных последовательностей конечно-автоматной абстракцией справедливо, если такие трассы начинаются во временных состояниях с нулевым значением времени. Таким образом, описанная выше абстракция временного автомата способна описывать лишь идентификационные эксперименты, которые начинаются после сброса времени, что отражено в следующем утверждении.

Утверждение 2. Временная входная последовательность $\alpha.(t)$ индуцирует временную трассу с ТПО-проекцией $\alpha/\gamma.(t)$ во временном состоянии $(s, 0)$ временного автомата S , если и только если входная последовательность α^{FSM} индуцирует трассу с IO-проекцией $\alpha^{FSM}/\gamma^{FSM}$ в состоянии $(s, 0)$ конечно-автоматной абстракции A_S .

Доказательство данного утверждения можно провести по индукции аналогично таковому для других абстракций временных автоматов [21-22]. Базой индукции являются следующие входные последовательности.

- 1) В S существует временной переход $(s, t)_m \rightarrow (s', t')$, если и только если в состоянии (s, d) конечно-автоматной абстракции A_S существует трасса из $[m]$ переходов $(11 \rightarrow)^{[m]}$ в состояние (s', d') , $d = t$, где, либо $d' = [t']$, либо $d' = B_S(s) + 1$ и (s', t') является b -стабильным.
- 2) В S существует вход-выходной переход $(s, t)_{io} \rightarrow (s', 0)$, если и только если в A_S существует трасса $(s, d)_{io} \rightarrow (s', 0)$, где $d = t$ или в A_S существует трасса $(s, d)_{io} \rightarrow (s', 0)$, где $d = [t]$ и $d \neq t$.

Соответственно, для любого временного состояния $(s, 0)$ автомата S и любого временного входного символа (i, t) существует трасса с ТПО-проекцией $(i, t)/o$, если и только если в состоянии $(s, 0)$ абстракции A_S входная последовательность $1^{[t]} i$ индуцирует трассу с IO-

проекцией $1/1^{i/o}$. Подобные рассуждения распространяются на последовательности произвольной длины по индукции.

Несмотря на то, что рассуждения выше приведены для автомата со сброшенной перед подачей последовательности в 0 временной переменной, аналогичное утверждение может быть доказано для случая, когда в момент начала эксперимента временная переменная имеет целочисленное значение. Последнее не является важным в контексте решаемой задачи, поэтому дальше мы проводим все рассуждения в предположении нулевого значения временной переменной в начале эксперимента.

Заметим, что в контексте задач идентификации состояний особого рассмотрения заслуживает вопрос о соответствии между преемниками временных и конечно-автоматных трасс, чему будут посвящены последующие разделы.

5. Проверка существования и синтез безусловных установочных и синхронизирующих экспериментов

Большинство алгоритмов проверки существования и построения идентификационных последовательностей для конечных автоматов основаны на построении усеченного дерева преемников [см., например, 4, 7]. В силу установленного выше соответствия между вход-выходными последовательностями временного автомата и его абстракции, мы далее предлагаем адаптировать существующие алгоритмы к синтезу установочных и синхронизирующих последовательностей для временных автоматов.

Каждая вершина дерева преемников конечно-автоматной абстракции помечается подмножеством пар различных состояний и синглетонов; рёбра дерева помечаются входными символами. Корню дерева ставится в соответствие множество всех пар различных начальных состояний конечного автомата, в которых может начинаться установочный или синхронизирующий эксперимент. Из вершины, помеченной непустым множеством P , существует ребро с входным символом i в вершину следующего уровня, помеченную множеством Q , где Q содержит пару состояний $\{p'_1, p'_2\}$ или синглетон $\{p'\}$, если в автомате существуют переходы (p_1, i, o, p'_1) , (p_2, i, o, p'_2) или (p_1, i, o, p') , (p_2, i, o, p') , $\{p_1, p_2\} \in P$. Вершина, помеченная множеством Q , является листом, если множество Q содержит только синглетоны или включает непустое множество пар, помечающее вершину меньшего уровня. Установочная последовательность существует, если и только если в дереве есть вершина, помеченная только множеством синглетонов. В качестве установочной последовательности можно выбрать кратчайшую последовательность входных символов, помечающую путь из корня дерева в такую вершину.

Если существует необходимость построить P' -установочную последовательность, переводящую автомат в известное состояние из множества P' , то в дереве должна существовать терминальная вершина, содержащая только синглетоны с состояниями из множества P' [7].

Если существует необходимость построить синхронизирующую (P' -синхронизирующую) последовательность, переводящую автомат в известное состояние (из множества P'), то в дереве преемников должна существовать вершина, помеченная единственным синглетоном (единственным синглетоном с состоянием из P'). Для синтеза таких последовательностей, в корень дерева достаточно поместить только синглетоны для всех состояний автомата.

Основной подход к синтезу установочных и синхронизирующих последовательностей для временных автоматов включает следующие этапы, но может немного модифицироваться в зависимости от типа последовательности.

1. Строится конечно-автоматная абстракция A_s временного автомата S .
2. С использованием классических алгоритмов проверяется, существует ли установочная/синхронизирующая последовательность для конечно-автоматной

абстракции A_s . Если для A_s такой последовательности не существует, то ее не существует и для временного автомата S . В противном случае строится установочная/синхронизирующая последовательность для A_s с использованием классических алгоритмов, которая преобразуется во временную входную последовательность.

Непосредственным использованием описанного выше подхода могут быть построены установочные и синхронизирующие последовательности, которые не являются стабильными. Из определения конечно-автоматной абстракции следует, что временная установочная последовательность, переводящая временной автомат в известное временное состояние, в котором значение временной компоненты не является целочисленным, не может быть отражена в абстракции. Тем не менее, в случае необходимости такие временные установочные последовательности достаточно просто могут быть получены из представленных в абстракции установочных последовательностей, переводящих автомат во временные состояния с целочисленным значением временной переменной. Следующее утверждение устанавливает связь между идентификационными последовательностями временного автомата и его конечно-автоматной абстракции.

Утверждение 3. Временная входная последовательность α без постфиксной задержки, является установочной для временного автомата S , если и только если входная последовательность α^{FSM} является установочной для конечно-автоматной абстракции A_s .

Доказательство. Соответствие между вход-выходными последовательностями временного автомата и его абстракции было установлено в утверждении 2, из которого также явно следует, что для временной входной последовательности без постфиксной задержки справедливо следующее: tr -преемником трассы tr с ТЮ-проекцией α/ψ во временном состоянии $(s, 0)$ автомата S является временное состояние $(s', 0)$, если и только если состояние $(s', 0)$ является tr^{FSM} -преемником трассы tr^{FSM} с Ю-проекцией α^{FSM}/ψ^{FSM} в состоянии $(s, 0)$ автомата A_s . В этом случае, все временные трассы временного автомата с ТЮ-проекцией α/ψ заканчиваются одним и тем же преемником, если и только если все трассы абстракции с Ю-проекцией α^{FSM}/ψ^{FSM} заканчиваются в одном и том же состоянии.

Таким образом, утверждение 3 устанавливает соответствие между существованием установочных последовательностей без постфиксной задержки во временном автомате и его конечно-автоматной абстракции.

Рассмотрим временные входные последовательности с постфиксной задержкой.

Утверждение 4. Временная входная последовательность $\alpha(t)$, где $(t - t_n)$ - целое положительное число и t_n - время подачи последнего входного символа, является установочной для временного автомата S , если и только если входная последовательность α^{FSM} является установочной для конечно-автоматной абстракции A_s .

Доказательство. Если последовательность $\alpha(t)$ имеет целочисленную постфиксную задержку и индуцирует во временном состоянии $(s, 0)$ трассу tr с ТЮ-проекцией $\alpha/\psi(t)$, а α^{FSM} индуцирует в состоянии $(s, 0)$ трассу tr^{FSM} с Ю-проекцией α^{FSM}/ψ^{FSM} , то tr -преемником состояния $(s, 0)$ является временное состояние (s, d) , если и только если tr^{FSM} -преемником является состояние (s, d') , где, либо $d = d'$ (Случай 1), либо $d > d'$ и (s, d) является b -стабильным (Случай 2). В первом случае, целочисленная задержка во временном автомате непосредственно отражена в абстракции последовательностью символов 1. Случай 2 соответствует ситуации, когда абстракция достигла состояния $(s, B_s(s))$ или $(s, B_s(s) + 1)$, в которых переход по символу 1 является «петлёй» (оставляет автомат в текущем состоянии), а временной автомат продолжает наращивать значение временной переменной. В случае 1 рассуждения о преемниках аналогичны таковым для последовательностей без постфиксных задержек. В случае 2, все временные трассы временного автомата с ТЮ-проекцией α/ψ заканчиваются преемником из одного и того же класса b -стабильных временных состояний,

если и только если все трассы абстракции с ИО-проекцией $\alpha^{FSM}/\gamma^{FSM}$ заканчиваются в одном и том же состоянии.

Аналогичное утверждение можно сформулировать для синхронизирующих последовательностей.

Утверждение 5. Временная входная последовательность $\alpha(t)$, где $(t - t_n)$ - целое положительное число и t_n - время подачи последнего входного символа, является синхронизирующей для временного автомата S , если и только если входная последовательность α^{FSM} является синхронизирующей для конечно-автоматной абстракции A_S .

Отметим, что установочные (синхронизирующие) последовательности с постфиксной задержкой, которая не является целочисленной, не могут быть отражены в абстракции. Большинство таких последовательностей не являются безызбыточными, кроме случая, когда последовательность может установить временной автомат в класс b -стабильных временных состояний $\{(s, t) : t > B_S(s)\}$, который не включает состояние $(s, B_S(s))$. Подобное может произойти, если в некоторые переходы автомата помечены интервалом $(B_S(s); \infty)$, т.е. постфиксной задержки $B_S(s)$ недостаточно, чтобы перевести автомат в класс b -стабильности, а дожидаться следующего целого такта необязательно.

Постфиксная задержка в установочной (синхронизирующей) последовательности определяется временем, которое придётся потратить на переход автомата в известное состояние или класс b -стабильности после подачи последнего входного символа. Если во временном автомате существует установочная последовательность $\alpha(t)$ с нецелочисленной постфиксной задержкой, то для $\alpha(t)$ существует трасса, переводящая автомат в класс b -стабильных состояний $\{(s, t) : t > B_S(s)\}$, притом уменьшение задержки до $[t]$ нарушит свойство $\alpha(t)$ быть установочной. Тем не менее, $\alpha(t)$ можно заменить на последовательность $\alpha([t] + 1)$, которая также переведёт временной автомат в класс b -стабильных состояний и будет обнаружена абстракцией. На основе приведённых рассуждений можно доказать следующее утверждение.

Утверждение 6. Для временного автомата S существует установочная (синхронизирующая) последовательность длины l (состоящая из l временных входных символов), если и только если для конечно-автоматной абстракции A_S существует установочная последовательность, в которой l отличных от 1 входных символов.

Таким образом, кратчайшая по временным входным символам временная установочная последовательности всегда может быть найдена по конечно-автоматной абстракции, но, при таком подходе, постфиксная задержка будет округлена до ближайшего целого числа. Иными словами, временной установочной последовательности может быть достаточно ожидания менее одного такта после последнего входного символа, а абстрактной последовательности придётся ожидать целый такт в силу ограничения модели.

Для построения временных установочных и синхронизирующих последовательностей специального вида основной алгоритм немного модифицируется.

Модификация 1. Для синтеза стабильных установочных (синхронизирующих) последовательностей по конечно-автоматной абстракции строится S' -установочная (S' -синхронизирующая) последовательность, где

$$S' = \{(s, B_S(s)) : s \in S \text{ и } ((s, B_S(s)), 1, 1, (s, B_S(s))) \in h_{A_S}\} \cup \{(s, B_S(s) + 1) : s \in S \text{ и } ((s, B_S(s) + 1), 1, 1, (s, B_S(s) + 1)) \in h_{A_S}\}.$$

Модификация 2. Для синтеза s -установочных последовательностей, при построении дерева преемников конечно-автоматной абстракции вносится дополнительное правило усечения: вершина, помеченная множеством Q , является листом, если для любой пары $((s_1, d_1), (s_2, d_2)) \in Q$ справедливо, что $s_1 = s_2$. Последовательность из корня дерева в такую вершину является s -установочной.

Модификация 3. Для синтеза s -синхронизирующих последовательностей, при построении дерева преемников конечно-автоматной абстракции вносится дополнительное правило усечения: вершина, помеченная множеством Q , является листом, если для множества её состояний $(s_1, d_1), (s_2, d_2), \dots, (s_n, d_n) \in Q$ справедливо, что $s_i = s_j, 0 < i \leq n, 0 < j \leq n$. Последовательность из корня дерева в такую вершину является s -синхронизирующей.

6. Заключение

Подводя итог работы, можно выделить следующие основные результаты. Предложены определения и классификация установочных и синхронизирующих последовательностей для автоматов с таймаутами и временными ограничениями.

Предложен алгоритм проверки существования и синтеза установочных и синхронизирующих последовательностей для автоматов с таймаутами и временными ограничениями на основе перехода к соответствующей конечно-автоматной абстракции.

Заметим, что в работе не рассматриваются вопросы сложности построения установочных и синхронизирующих последовательностей для таких временных автоматов. Кроме того, требуют дополнительного изучения вопросы построения таких стабильных последовательностей с наименьшим временем исполнения. Эти и ряд других вопросов по построению идентификационных последовательностей для автоматов с временными ограничениями и таймаутами предполагается включить в будущие исследования.

Список литературы / References

- [1]. Moore E. F. Gedanken-experiments on sequential machines. Automata Studies (Annals of Mathematical Studies), 1956, 1, pp. 129-153.
- [2]. Гилл А. Введение в теорию конечных автоматов. Наука, 1966.
- [3]. Hibbard T.N. Least upper bounds on minimal terminal state experiments of two classes of sequential machines. J. ACM, 8, 1961, pp. 601-612.
- [4]. Kohavi Z. Switching and Finite Automata Theory, McGraw-Hill, 1978.
- [5]. D. Lee, M. Yannakakis. Testing Finite-State Machines: State Identification and Verification. IEEE Transactions on Computers, 1994, 43(3), pp. 306-320.
- [6]. Sandberg S. Homing and Synchronizing Sequences. Lecture Notes in Computer Science, 2005, 3472, pp. 5-33.
- [7]. Н.В. Евтушенко, Н.Г. Кушик. Некоторые задачи идентификации состояний для недетерминированных автоматов. Томск, 2018.
- [8]. Hennie F.C. Fault detecting experiments for sequential circuits. Proc. of the 5th Annual Symposium on Circuit Theory and System Design, 1965, pp. 95-100.
- [9]. Chow T. S. Testing software Design Modelled by Finite State Machines. IEEE Trans. Software Eng., 1978, 4(3), pp. 178-187.
- [10]. Lee, D., Yannakakis, M. Principles and methods of testing finite state machines-a survey. Proceedings of the IEEE, 1996, 8, 4(8), pp.1090-1123.
- [11]. Bochmann G. and Petrenko A. Protocol testing: review of methods and relevance for software testing. Proceedings of the 1994 ACM International Symposium on Software Testing and Analysis (ISSTA'94), pp. 109-124.
- [12]. Kushik, N., López, J., Cavalli, A. and Yevtushenko, N. Improving Protocol Passive Testing through "Gedanken" Experiments with Finite State Machines. Proceedings of Intern. Conf. Quality, Reliability and Security, 2016, pp 315-322.
- [13]. R.L. Rivest and R.E. Schapire. Inference of finite automata using homing sequences. Inf. Comput., 1993, 103(2), pp. 299-347.
- [14]. Roland Groz, Nicolas Br'emon, Adenilso da Silva Simao, and Catherine Oriat Winference: A heuristic approach to retrieve models through black box testing. J. Syst. Softw., 2020, 159.
- [15]. Loes Kruger, Bharat Garhwal, Frits Vaandrager. Lower Bounds for Active Automata Learning. Proceedings of 16th edition of the International Conference on Grammatical Inference, 2023.
- [16]. Yennigun H., Yevtushenko N., Kushik N. and López, J. The effect of partiality and adaptivity on the complexity of FSM state identification problems. Труды ИСП РАН, 2018, том 3, pp. 1-14

- [17]. Alur, R, and Dill. D. L. A Theory of Timed automata. *Theoretical Computer Science*, 1994, 126(2), pp.183-235.
- [18]. Merayo M. G., Nunez, M., Rodriguez I.: Formal Testing from Timed Finite State Machines. *Computer Networks*, 2008, 52 (2), pp. 432–460 (2008)
- [19]. Springintveld J, Vaandrager F, D'Argenio P (2001) Testing timed automata. *Theoretical Computer Science*, 2001, 254(1-2), pp. 225–257.
- [20]. Bresolin D, El-Fakih K, Villa T., Yevtushenko N. Equivalence checking and intersection of deterministic timed finite state machines. *Formal Methods Syst. Des.* 59(1), 2021, pp. 77-102.
- [21]. Tvardovskii A, Yevtushenko N. Deriving homing sequences for Finite State Machines with timed guards. *Automatic Control and Computer Sciences*, 2021, 55 (7), pp. 738–750.
- [22]. Tvardovskii A, Yevtushenko N. Deriving homing sequences for Finite State Machines *with timeouts*. *Comput. Journal*, 2023, 66 (9), pp. 2181-2190.

Информация об авторах / Information about authors

Александр Сергеевич ТВАРДОВСКИЙ – кандидат физико-математических наук, доцент ТГУ. Научные интересы: формальные спецификации, генерация тестов, системы реального времени, теория автоматов.

Alexandr Sergeevich TVARDOVSKII – Cand. Sci. (Phys.-Math.), an associate professor of TSU. Research interests: formal specifications, test generation, real-time systems, automata theory.

Игорь Борисович БУРДОНОВ – доктор физико-математических наук, главный научный сотрудник ИСП РАН. Научные интересы: формальные спецификации, генерация тестов, технология компиляции, системы реального времени, операционные системы, объектно-ориентированное программирование, сетевые протоколы, процессы разработки программного обеспечения.

Igor Borisovich BURDONOV – Dr. Sci. (Phys.-Math.), a Leading Researcher of ISP RAS. Research interests: formal specifications, test generation, compilation technology, real-time systems, operating systems, object-oriented programming, network protocols, software development processes.

Нина Владимировна ЕВТУШЕНКО, доктор технических наук, профессор, главный научный сотрудник ИСП РАН, до 1991 года работала научным сотрудником в Сибирском физико-техническом институте. С 1991 г. работала в ТГУ профессором, зав. кафедрой, зав. лабораторией по компьютерным наукам. Её исследовательские интересы включают формальные методы, теорию автоматов, распределённые системы, протоколы и тестирование программного обеспечения.

Nina Vladimirovna YEVTUSHENKO, Dr. Sci. (Tech.), Prof., a Leading Researcher of ISP RAS, worked at the Siberian Scientific Institute of Physics and Technology as a researcher up to 1991. In 1991, she joined Tomsk State University as a professor and then worked as the chair head and the head of Computer Science laboratory. Her research interests include formal methods, automata theory, distributed systems, protocol and software testing.