Моделирование и анализ информационных систем. Т. 24, №6 (2017), с. 730–742
Modeling and Analysis of Information Systems. Vol. 24, No 6 (2017), pp. 730–742

DOI: 10.18255/1818-1015-2017-6-730-742

УДК 519.713

К синтезу синхронизирующих и установочных последовательностей для входо-выходных полуавтоматов

Кушик Н. Г., Евтушенко Н. В., Бурдонов И. Б., Косачев А. С.

получена 3 сентября 2017

Аннотация. В работе рассматриваются задачи проверки существования и синтеза синхронизирующих и установочных последовательностей для конечных входо-выходных полуавтоматов. Соответствующие последовательности могут быть использованы при идентификации состояния проверяемой системы после подачи подходящей входной последовательности. В модели, исследуемой в работе, действия разделены на входные и выходные, однако отсутствуют выделенные явно семейства начальных и финальных состояний. В статье определяются понятия синхронизирующей и установочной последовательностей и предлагаются методы их синтеза для специального класса входо-выходных полуавтоматов, у которых в каждом состоянии определены переходы или только по входным, или только по выходным действиям; кроме того, в соответствующем графе переходов отсутствуют циклы по выходным символам. Для описанного класса входо-выходных полуавтоматов устанавливаются необходимые и достаточные условия существования синхронизирующих и установочных последовательностей и оценивается длина таких последовательностей. Выделяются подклассы полуавтоматов, для которых худшие (в основном экспоненциальные) оценки сложности не являются достижимыми.

Ключевые слова: входо-выходные полуавтоматы, синхронизирующая последовательность, установочная последовательность

Для цитирования: Кушик Н. Г., Евтушенко Н. В., Бурдонов И. Б., Косачев А. С., "К синтезу синхронизирующих и установочных последовательностей для входо-выходных полуавтоматов", *Моделирование и анализ информационных систем*, **24**:6 (2017), 730–742.

Об авторах:
Кушик Наталья Геннадьевна, д-р физ.-мат. наук,
Университет Телеком Южный Париж / Париж Сакле,
ул. Ш. Фурье, 9, г. Еври, 91000 Франция, e-mail: natalia.kushik@telecom-sudparis.eu

Евтушенко Нина Владимировна, orcid.org/0000-0002-4006-1161, д-р техн. наук, профессор,
Институт системного программирования РАН, ул. Солженицына, 25, г. Москва, 109004 Россия
Томский государственный университет, ул. Ленина, 36, г. Томск, 634050 Россия, e-mail: nyevtush@gmail.com

Бурдонов Игорь Борисович, orcid.org/0000-0001-9539-7853, д-р физ.-мат. наук, старший научный сотрудник,
Институт системного программирования РАН, ул. Солженицына, 25, г. Москва, 109004 Россия, e-mail: igor@ispras.ru

Косачев Александр Сергеевич, orcid.org/0000-0001-5316-3813, канд. физ.-мат. наук, старший научный сотрудник,
Институт системного программирования РАН, ул. Солженицына, 25, г. Москва, 109004 Россия, e-mail: kos@ispras.ru

Благодарности:
Работа частично поддержана проектами РНФ № 16-49-03012 и РФФИ № 17-07-00682.

Введение

Проблема идентификации состояний в конечных автоматах (Finite State Machines или FSMs в англоязычной литературе) активно исследуется, начиная с первой работы Мура, посвященной синтезу так называемых «умозрительных» экспериментов для конечных автоматов [12]. Результаты Мура были в дальнейшем развиты исследователями для различных классов конечных автоматов (см., например, [2, 6, 11]). При идентификации текущего/финального состояния на исследуемую систему подается заранее сформированная синхронизирующая или установочная последовательность. Если последовательность синхронизирующая, то из любого состояния исследуемая система переходит в известное состояние. Если последовательность установочная, то по наблюдаемой реакции системы на установочную последовательность делается вывод о состоянии, достигнутом системой после подачи последовательности. В настоящей работе мы не рассматриваем адаптивные входные последовательности, т.е. входная синхронизирующая (или установочная) последовательность сформирована заранее, до проведения эксперимента с системой.

Идентификация состояний используется в различных приложениях. Одним из таких приложений являются автоматные методы синтеза проверяющих тестов для дискретных управляющих систем [4, 7, 13], и существует достаточно много публикаций, каким образом можно построить такие последовательности для детерминированных и недетерминированных (в том числе ненаблюдаемых), частичных и полностью определенных автоматов [1, 5, 7, 13]. Помимо так называемого «активного» тестирования, идентификация текущего состояния проверяемой системы используется также в задачах мониторинга (пассивного тестирования) при наблюдении поведения проверяемой реализации. В частности в работе [8] авторы исследуют возможность ускорения/оптимизации мониторинга при наличии в спецификации системы синхронизирующих и/или установочных последовательностей. Если текущее состояние системы известно, то можно существенно сократить множество проверяемых свойств при “пассивной” верификации системы.

Тем не менее, при описании поведения дискретных управляющих систем с целью проверки их надежности и корректности функционирования достаточно часто используются не только модель конечного автомата, но также и входо-выходные полуавтоматы. Причиной является тот факт, что выходная реакция системы не обязательно появляется непосредственно после каждого входного символа, и, вообще говоря, на один входной символ возможна реакция в виде последовательности выходных символов. Входо-выходные полуавтоматы позволяют описать такие ситуации подходящей математической моделью, однако методы анализа таких моделей с точки зрения идентификации текущего/финального состояния авторам работы не известны. Соответственно, предлагаемые методы построения установочных и синхронизирующих последовательностей для такой модели являются новыми.

Следует отметить, тем не менее, что синхронизирующие эксперименты для полуавтоматов рассматривались и ранее, однако в известных случаях действия, помечающие переходы, не разделены на входные и выходные [1, 13, 15]. Иными словами, при синтезе эксперимента во внимание принимается тот факт, что в каждом состоянии полуавтомата может быть подан любой из определенных в данном состоянии символов, которые можно понимать как входные символы; выходных символов среди действий нет. В данной работе мы рассматриваем более широкий класс полуавтоматов, а именно полуавтоматы, в которых в каждом состоянии может быть или принят входной, или произведен выходной символ. Таким образом, мы распространяем полученные результаты по синтезу синхронизирующих экспериментов на исследуемый класс полуавтоматов.

Мы отмечаем, что представленные в работе результаты частично вошли в публикацию [16] по результатам Восьмого симпозиума по спецификации и верификации программ "Program Semantics, Specification and Verification: Theory and Applications" (2017 г.). В настоящей статье мы "наследуем" развитые в предыдущей статье алгоритмы. Научная новизна данной работы заключается в 1) доказательстве корректности предложенных алгоритмов, 2) установлении оценок длин синхронизирующей и установочной последовательностей для рассматриваемого класса входо-выходных полуавтоматов и 3) выделении специальных подклассов входо-выходных полуавтоматов, для которых худшие (в основном экспоненциальные) оценки сложности не являются достижимыми.

Структура работы следующая. В первом разделе приводятся основные определения и обозначения. Методы синтеза установочных и синхронизирующих последовательностей для входо-выходных полуавтоматов приведены во втором разделе. Третий раздел содержит заключение, в котором, в частности, освещаются перспективы дальнейших научных исследований.

1. Основные определения и обозначения

Под *входо-выходным полуавтоматом* (или просто *полуавтоматом*) в этой статье понимается четверка $\boldsymbol{S} = (S, I, O, T_S)$, где S есть конечное непустое множество состояний; I и O суть конечные непересекающиеся алфавиты входных и выходных символов; $T_S \subseteq S \times I \times S \cup S \times O \times S$ есть *отношение переходов*, где тройки $(s, i, s') \in T_S$ и $(s, o, s') \in T_S$ суть *переходы* [1].

В этой статье мы рассматриваем специальный класс полуавтоматов, для которых выполняются следующие условия.

1. В каждом состоянии полуавтомата определены только входные символы или только выходные символы, т.е. $S = S_1 \cup S_2$, $S_1 \cap S_2 = \emptyset$ и $T_S \subseteq S_1 \times I \times S \cup S_2 \times O \times S$, причем в полуавтомате отсутствуют состояния, из которых нет переходов.

2. Диаграмма переходов полуавтомата не содержит циклов, помеченных только выходными символами, т.е. язык полуавтомата не содержит последовательностей с бесконечным суффиксом из выходных символов.

3. В полуавтомате есть специальный выходной символ $\delta \notin O$, представляющий так называемое "молчание" (англ. – "quiescence [14]") в состояниях, где определены переходы по входным символам: соответственно в каждом состоянии $s \in S_1$ определен переход-петля, помеченный символом δ, т.е. $(s, \delta, s) \in T_S$.

[1] Таким образом, в данной работе рассматриваются полуавтоматы без переходов по ненаблюдаемым действиям и без выделения множеств начальных и финальных состояний.

Пример входо-выходного полуавтомата $\boldsymbol{S}$ показан на рис. 1. Полуавтомат имеет 5 состояний, т.е. $S = \{s_1, \ldots, s_5\}$, для которых $S_1 = \{s_1, s_2, s_5\}$ и $S_2 = \{s_3, s_4\}$. В каждом состоянии из S_1 полуавтомат "принимает" входные символы i_1 и i_2. В состояниях s_3 и s_4 полуавтомат не принимает входные символы, а выдает выходные реакции o_1 или o_2.

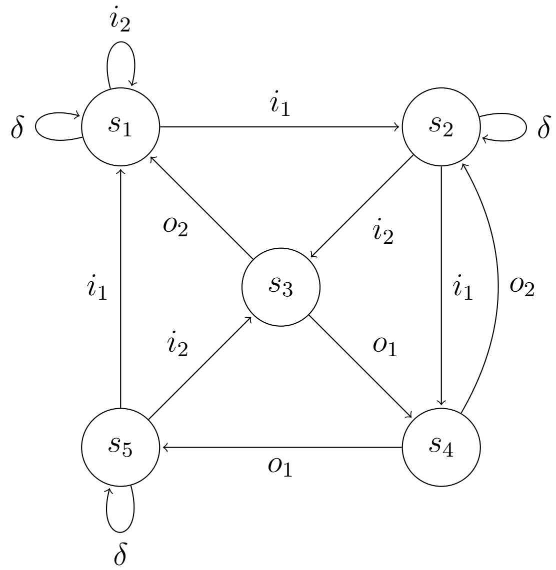

Рис. 1. Входо-выходной полуавтомат $\boldsymbol{S}$
Fig. 1. An Input/Output automaton $\boldsymbol{S}$

Как обычно, синхронизирующие и установочные входные последовательности используются для идентификации состояния полуавтомата после подачи такой последовательности, при условии, что начальное состояние полуавтомата не известно. Основываясь на [16], мы далее вводим определения таких последовательностей для входо-выходных полуавтоматов. Отметим, что подача входной последовательности и идентификация состояния после подачи последовательности осуществляются согласно следующему **предположению**.

Перед подачей первого входного символа тестер ожидает поступления выходного символа в течение определенного выходного таймаута t. Если полуавтомат выдает выходной символ, то таймер обнуляется, и тестер ждет очередной выходной символ в течение последующих t единиц времени. Если в течение t единиц времени выходной символ полуавтоматом не производится, то на тестер подается следующий входной символ или тестирование заканчивается, и таймер обнуляется. Такой порядок подачи входных символов объясняет наличие специального выходного символа $\delta \notin O$; когда выходной символ не наблюдается в течение t единиц времени, мы полагаем, что производится выходной символ δ. Выходной таймаут обычно определяется, исходя из самой длинной возможной выходной последовательности на поданный входной символ.

Как обычно, трассой в полуавтомате называется допустимая последовательность действий, и γ-преемник состояния $s \in S$ есть множество состояний, которые достижимы из состояния по трассе γ, т.е. γ-преемник множества S содержит каждое состояние из S, достижимое из некоторого состояния $s \in S$ по трассе γ. Синхронизирующей последовательностью называется входная последовательность, после подачи которой полуавтомат переходит в одно и то же состояние из любого начального состояния. Другими словами, последовательность $\alpha = i_1 i_2 \ldots i_k$ называется *синхронизирующей* для полуавтомата $\boldsymbol{S}$, если существует состояние $s \in S$, такое что для любой последовательности действий $\beta_1 i_1 \beta_2 i_2 \ldots \beta_k i_k \beta_{k+1}$ в полуавтомате, где p есть длина самой длинной последовательности только из выходных символов и $\beta_j \in (O \cup \{\delta\})^p$, $j = 1, \ldots, k+1$, справедливо, что $\beta_1 i_1 \beta_2 i_2 \ldots \beta_k i_k \beta_{k+1}$-преемник множества S (множество состояний $\beta_1 i_1 \beta_2 i_2 \ldots \beta_k i_k \beta_{k+1}$-state-after-$S$) есть пустое множество или $\{s\}$. Таким образом, синхронизирующая последовательность позволяет единственным образом идентифицировать состояние полуавтомата после ее подачи без наблюдения производимых полуавтоматом выходных реакций.

Установочная последовательность позволяет единственным образом идентифицировать состояние полуавтомата после ее подачи и наблюдения производимых полуавтоматом выходных реакций. Таким образом, последовательность $\alpha = i_1 i_2 \ldots i_k$ называется *установочной* для полуавтомата $\boldsymbol{S}$, если для каждой трассы $\beta_1 i_1 \beta_2 i_2 \ldots \beta_k i_k \beta_{k+1}$, $\beta_j \in (O \cup \{\delta\})^p$, $j = 1, \ldots, k+1$, справедливо, что $\beta_1 i_1 \beta_2 i_2 \ldots \beta_k i_k \beta_{k+1}$-преемник множества S есть пустое множество или синглетон. Непосредственной проверкой можно убедиться, что для автомата $\boldsymbol{S}$ на рис. 1 $\alpha = i_1 i_1$ является установочной последовательностью.

2. Методы синтеза синхронизирующих и установочных последовательностей для входо-выходных полуавтоматов

В данном разделе мы предлагаем алгоритмы построения синхронизирующих и установочных последовательностей для входо-выходных полуавтоматов из описанного выше класса. Кроме того, мы уточняем оценки сложности построения таких последовательностей и рассматриваем классы полуавтоматов с пониженными оценками длин синхронизирующих и установочных последовательностей.

2.1. Построение синхронизирующих последовательностей

Как отмечалось ранее, синхронизирующие последовательности и методы их синтеза хорошо изучены для “классических” полуавтоматов, в которых алфавит действий не разделен на подмножества входных (стимулов) и выходных (реакций) действий соответственно. Если в полуавтомате $\boldsymbol{S}$ определены переходы только по входным действиям, то последовательность действий α называется *синхронизирующей*, если существует состояние полуавтомата, в которое α переводит полуавтомат из любого начального состояния. Алгоритмы проверки существования и построения синхронизирующих последовательностей (если существуют) для таких, в том числе недетерминированных и, возможно, частичных, полуавтоматов опубликованы в [1,5,13,15]. При построении синхронизирующей последовательности для входо-выходного полуавтомата $\boldsymbol{S}$, который удовлетворяет условиям 1–3, описанным выше, мы предлагаем переход к "классическому" полуавтомату без выходных действий, который обладает тем же самым множеством синхронизирующих последовательностей.

Алгоритм 1: синтез полуавтомата $\boldsymbol{A}$

Вход : Входо-выходной полуавтомат $\boldsymbol{S} = (S, I, O, T_S)$

Выход: Полуавтомат $\boldsymbol{A}$

Шаг 1 Строим полуавтомат $\boldsymbol{A} = (S_1, I, T_A)$ с пустым множеством переходов, т.е. $T_A = \emptyset$.

Шаг 2. Для каждого перехода $(s, i, s'') \in T_S$, где $s, s'' \in S_1$, добавляем к T_A переход (s, i, s''); для каждого перехода (s, i, s''), где $s \in S_1$ и $s'' \in S_2$, добавляем к T_A переход (s, i, s'''), где $s''' \in S_1$ и s''' является β-преемником состояния s'' в полуавтомате $\boldsymbol{S}$, $\beta \in O^*$.

Утверждение 1. *Множества синхронизирующих последовательностей для полуавтоматов* $\boldsymbol{A}$ *(построен по алгоритму 1) и* $\boldsymbol{S}$ *совпадают.*

Доказательство. По определению, последовательность $\alpha = i_1 \dots i_k$ переводит полуавтомат $\boldsymbol{A}$ из состояния $s' \in S_1$ в состояние $s \in S_1$, если и только если существуют $\beta_j \in (O \cup \{\delta\})^p$, $j = 1, \dots, k+1$, такие, что $\beta_1 i_1 \beta_2 i_2 \dots \beta_k i_k \beta_{k+1}$-преемник состояния s' во входо-выходном полуавтомате $\boldsymbol{S}$ есть состояние s. Последовательность α является синхронизирующей в полуавтомате $\boldsymbol{A}$, если и только если существует состояние $s \in S_1$, такое, что α-преемник любого состояния полуавтомата $\boldsymbol{A}$ равен $\{s\}$. Последнее означает, что $\beta_1 i_1 \beta_2 i_2 \dots \beta_k i_k \beta_{k+1}$-преемник любого состояния $s' \in S_1$ входо-выходного полуавтомата $\boldsymbol{S}$, где p есть длина самой длинной последовательности только из выходных символов и $\beta_j \in (O \cup \{\delta\})^p$, $j = 1, \dots, k+1$, есть пустое множество или $\{s\}$. Обратно, если α является синхронизирующей последовательностью во входо-выходном полуавтомате $\boldsymbol{S}$, то существует состояние $s \in S_1$, такое что $\beta_1 i_1 \beta_2 i_2 \dots \beta_k i_k \beta_{k+1}$-преемник любого состояния входо-выходного полуавтомата $\boldsymbol{S}$ есть пустое множество или $\{s\}$, т.е. α есть синхронизирующая последовательность в полуавтомате $\boldsymbol{A}$. □

Следствие 1. *Входо-выходной полуавтомат* $\boldsymbol{S}$ *обладает синхронизирующей последовательностью, если и только если полуавтомат* $\boldsymbol{A}$*, построенный по алгоритму 1, обладает синхронизирующей последовательностью.*

Отметим, что определенный выше класс входо-выходных полуавтоматов не требует полной определенности функции поведения в состояниях из множества S_1 по входным символам. Поэтому полуавтомат $\boldsymbol{A}$, синтезируемый по алгоритму 1, может оказаться частично определенным, если в исходном полуавтомате в некотором состоянии из множества S_1 определены переходы не по всем входным символам. С другой стороны, множество классических полуавтоматов образует подкласс исследуемого класса входо-выходных автоматов, и, вместе с тем, для частичных детерминированных полуавтоматов (в англоязычной литературе Partial Deterministic Automaton или PDA) задача проверки существования синхронизирующей последовательности является PSPACE-полной [3]. Этот факт обосновывает PSPACE-трудность задачи проверки существования синхронизирующей последовательности для входо-выходных полуавтоматов, исследуемых в данной статье.

Высокие оценки сложности задач проверки существования последовательностей, идентифицирующих текущее состояние полуавтомата, как известно, тесно связаны с максимальными оценками минимальных длин этих последовательностей. Следует отметить, что для входо-выходных полуавтоматов из исследуемого класса эта оценка является экспоненциальной даже в случае, когда поведение полуавтомата определено по всем входным символам в состояниях из множества S_1.

Утверждение 2. *Для синхронизируемого входо-выходного полуавтомата* $\boldsymbol{S}$*, поведение которого определено по всем входным символам в каждом из состояний из множества* S_1*, длина кратчайшей синхронизирующей последовательности не превосходит* $O(2^{|S_1|})$.

Доказательство. Полуавтомат $\boldsymbol{A}$, синтезируемый по алгоритму 1, имеет $|S_1|$ состояний и $|I|$ действий. Этот полуавтомат является полностью определенным, однако он может быть недетерминированным, поскольку для одного и того же входного символа в некотором состоянии возможно существование различных выходных последовательностей, переводящих исходный входо-выходной полуавтомат $\boldsymbol{S}$ в различные состояния из одного и того же состояния из множества S_1. Длина кратчайшей синхронизирующей последовательности полуавтомата $\boldsymbol{A}$ в худшем случае экспоненциально зависит от числа состояний, в частности, для "классического" полуавтомата с множеством состояний S_1 может иметь длину порядка $2^{|S_1|}$ [5], что доказывает утверждение. □

В качестве примера рассмотрим проверку существования синхронизирующей последовательности для полуавтомата на рис. 1. Соответствующий "классический" полуавтомат $\boldsymbol{A}$ (без выходных действий) показан на рис. 2. Непосредственной проверкой можно убедиться, что для полуавтомата $\boldsymbol{A}$ на рис. 2 не существует синхронизирующей последовательности. Таким образом, и для входо-выходного полуавтомата на рис. 1 не существует синхронизирующей последовательности.

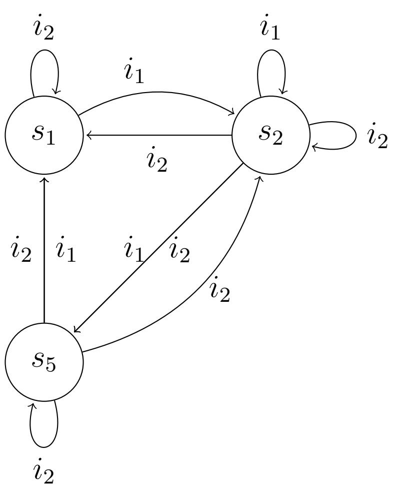

Рис. 2. Полуавтомат $\boldsymbol{A}$, полученный после применения алгоритма 1
Fig. 2. Automaton $\boldsymbol{A}$ returned by Algorithm 1

2.2. Построение установочных последовательностей

В настоящей работе задача проверки существования и синтеза установочных последовательностей для входо-выходных полуавтоматов сводится к задаче синтеза установочных последовательностей для классических, возможно, частичных и недетерминированных автоматов, для которых такая задача хорошо исследована [6–8].

Рассмотрим конечный автомат $\boldsymbol{M} = (M, I, O, h_M)$ [2], в котором M – конечное непустое множество состояний, I и O - конечные непересекающиеся входной и выходной алфавиты. В отличие от рассмотренной выше модели входо-выходного полуавтомата переходами в автомате являются четверки (s, i, o, s'), т.е. в любой трассе автомата за каждым входным символом следует выходной символ, и каждая трасса заканчивается выходным символом. Автомат называется *полностью определенным*, если в каждом состоянии определен переход по каждому входному символу, иначе, автомат называется частично определенным. Автомат называется *детерминированным*, если в каждом состоянии по каждому входному символу определено не более одного перехода, иначе, автомат называется *недетерминированным*. Обычным образом отношение переходов автомата распространяется на входные и выходные последовательности. Автомат называется *сильно связным*, если для любых двух состояний m_1 и m_2 существует входная последовательность, переводящая автомат из состояния m_1 в состояние m_2. Детерминированный автомат называется *приведенным*, если для любых двух состояний существует входная последовательность, выходные реакции на которую в этих состояниях различны.

Входная последовательность α называется *установочной* в автомате, если для любой трассы γ, проекция которой на входной алфавит равна α, γ-преемник каждого подмножества состояний имеет мощность не более единицы, причем одинаковые выходные реакции на последовательность α гарантируют достижимость одного и того же состояния. Методы синтеза установочных последовательностей для конечных автоматов хорошо исследованы, в частности, для полностью определенных детерминированных автоматов – в [6, 13]; для недетерминированных автоматов – в [8]; для частичных автоматов – в [17]. Подобно методу построения синхронизирующих последовательностей, мы предлагаем достаточно простую процедуру построения конечного автомата по заданному входо-выходному полуавтомату, для которых множества установочных последовательностей совпадают.

Алгоритм 2: синтез автомата $\boldsymbol{M}$

Вход : Входо-выходной полуавтомат $\boldsymbol{S} = (S, I, O, T_S)$

Выход: Автомат $\boldsymbol{M}$

Шаг 1 Строим автомат $\boldsymbol{M} = (S_1, I, O \cup O^2 \cup \dots \cup O^p \cup \{\delta\}, T_M)$ с пустым множеством переходов, т.е. $T_M = \emptyset$, где p есть длина самой длинной последовательности из выходных символов в полуавтомате $\boldsymbol{S}$.

Шаг 2. Для каждого состояния $s \in S_1$, такого что $(s, i, s') \in T_S$, $s' \in S_1$, добавляем к T_M переход (s, i, δ, s').

Шаг 3. Для каждого состояния $s \in S_1$, такого что $(s, i, s') \in T_S$, $s' \in S_2$, добавляем к T_M переход $(s, i, o_1 o_2 \dots o_k, s'')$, $k \le p$, где $s'' \in S_1$ есть $o_1 o_2 \dots o_k$-преемник состояния s'.

Утверждение 3. *Множества установочных последовательностей автомата* $\boldsymbol{M}$ *(построен по алгоритму 2) и полуавтомата* $\boldsymbol{S}$ *совпадают.*

Доказательство. По определению, последовательность $\alpha = i_1 \dots i_k$ является установочной в автомате $\boldsymbol{M}$, если для любой трассы γ, проекция которой на входной алфавит равна α, γ-преемник каждого подмножества состояний имеет мощность не более единицы, причем одинаковые выходные реакции на последовательность α гарантируют достижимость одного и того же состояния. По построению автомата $\boldsymbol{M}$, последнее означает, что $\beta_1 i_1 \beta_2 i_2 \dots \beta_k i_k \beta_{k+1}$-преемник любого состояния входо-выходного полуавтомата $\boldsymbol{S}$, где p есть длина самой длинной последовательности только из выходных символов и $\beta_j \in (O \cup \{\delta\})^p$, $j = 1, \dots, k+1$, есть пустое множество или $\{s\}$ для подходящего $s \in S$, причем одинаковые трассы $\beta_1 i_1 \beta_2 i_2 \dots \beta_k i_k \beta_{k+1}$ гарантируют достижимость одного и того же состояния из различных состояний. Обратно, если α является установочной последовательностью во входо-выходном полуавтомате $\boldsymbol{S}$, то $\beta_1 i_1 \beta_2 i_2 \dots \beta_k i_k \beta_{k+1}$-преемник любого состояния входо-выходного полуавтомата $\boldsymbol{S}$ не более чем одноэлементен, т.е. α есть установочная последовательность в автомате $\boldsymbol{M}$. □

Следствие 2. *Входо-выходной полуавтомат* $\boldsymbol{S}$ *обладает установочной последовательностью, если и только если автомат* $\boldsymbol{M}$*, построенный по алгоритму 2, обладает установочной последовательностью.*

Как отмечалось ранее, для проверки существования и построения установочной последовательности в автомате $\boldsymbol{M}$ можно воспользоваться алгоритмами из [6, 8, 9, 13], которые "обрабатывают" детерминированные и недетерминированные, полностью и частично определенные автоматы. Отметим также, что в случае ненаблюдаемости недетерминизма полученного конечного автомата $\boldsymbol{M}$ в силу высокой сложности задачи идентификации финального состояния автомата удобно воспользоваться различными эвристиками усечения так называемого установочного дерева (получается из дерева преемников, представляющего "развертку" поведения автомата). Например, в работе [8] предлагается использовать поиск установочной последовательности для частичного, возможно, ненаблюдаемого автомата с ограничением на максимальную длину требуемой последовательности. Этот факт обосновывается оценкой длины кратчайшей установочной последовательности для недетерминированных автоматов даже в случае, когда исходный полуавтомат является определенным по всем входным символам в состояниях из множества S_1.

Известные результаты по достижимости оценки длины кратчайшей установочной последовательности для наблюдаемого полностью определенного автомата обосновывают достижимость экспоненциальной оценки длины установочной последовательности для входо-выходного полуавтомата. В этом случае автомат из работы [10] может быть естественным образом преобразован в полуавтомат "разверткой" каждого перехода i/o в последовательность io длины два. Таким образом, имеет место следующее утверждение.

Утверждение 4. *Для любого* $n > 3$ *существует входо-выходной полуавтомат* $\boldsymbol{S}$*, для которого* $|S_1| = n$ *и длина кратчайшей установочной последовательности составляет* $2^{n-1} - 1$.

Возможность описанного выше преобразования конечного автомата во входо-выходной полуавтомат также обосновывает и PSPACE-трудность задачи проверки существования установочной последовательности, поскольку задача проверки существования установочной последовательности является PSPACE-полной даже для частично определенного детерминированного автомата. PSPACE-полноту задачи проверки существования установочной последовательности для входо-выходных полуавтоматов нужно исследовать дополнительно, поскольку в результате перехода (полиномиальной сложности) к классическому автомату можно в худшем случае получить частичный ненаблюдаемый автомат, для которого принадлежность подходящей задачи классу PSPACE остается под вопросом.

В качестве примера проверим наличие установочной последовательности в полуавтомате $\boldsymbol{S}$ на рис. 1. В этом полуавтомате нет синхронизирующей последовательности, однако непосредственной проверкой с использованием методов из [6, 7, 13] можно убедиться, что соответствующий автомат $\boldsymbol{M}$ (рис. 3) обладает установочной последовательностью $\alpha = i_1 i_1$.

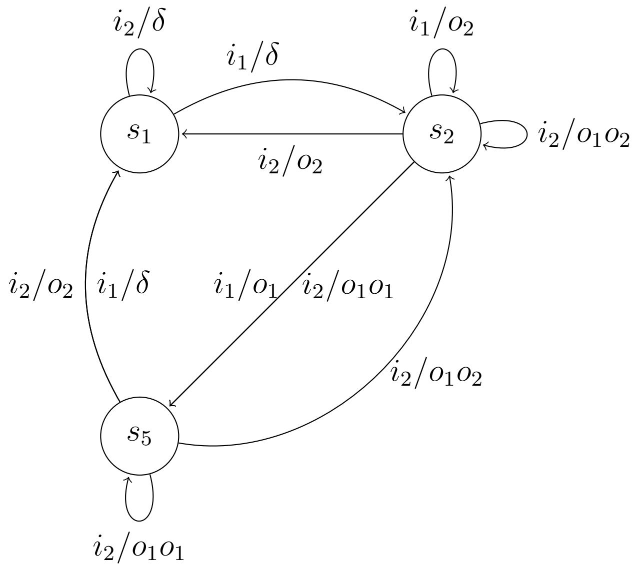

Рис. 3. Автомат $\boldsymbol{M}$, полученный из $\boldsymbol{S}$ после применения алгоритма 2
Fig. 3. Automaton $\boldsymbol{M}$ returned for $\boldsymbol{S}$ by Algorithm 2

Согласно приведенным выше рассуждениям и утверждениям, задачи проверки существования и построения синхронизирующих и установочных последовательностей для рассматриваемого класса входо-выходных полуавтоматов имеют достаточно высокую сложность. Однако в ряде случаев эта сложность может быть понижена за счет наложения дополнительных ограничений на исследуемый полуавтомат. Например, вновь обратимся к полуавтоматам, в которых в каждом допустимом состоянии определен переход по каждому входному символу. Договоримся также, что в каждом состоянии из множества S_2 определен единственный выходной символ, т.е. после каждого входного символа следует единственная последовательность выходных символов.

В этом случае подходящий автомат $\boldsymbol{M}$ и полуавтомат $\boldsymbol{A}$, синтезируемые по алгоритмам 1 и 2, представляют собой “хорошие” детерминированные, всюду определенные системы переходов. Задачи проверки существования и синтеза установочных и синхронизирующих последовательностей для полностью определенных сильно связных приведенных детерминированных автоматов (и "классических" полуавтоматов соответственно) могут быть решены за полиномиальное время, поэтому проблема идентификации финального/текущего состояния такого входо-выходного полуавтомата может быть решена за полиномиальное время. Более того, длины соответствующих входных последовательностей ограничиваются полиномом второй и третьей степеней относительно мощности подмножества состояний полуавтомата, в которых определены переходы по входным символам. Примером "хорошего" класса входо-выходных полуавтоматов может служить подмножество детерминированных по выходам полуавтоматов, у которых в состояниях из множества S_1 определен переход по каждому входному символу $i \in I$, а в состояниях из множества S_2 – переход только по одному выходному символу. Если для каждого входо-выходного полуавтомата из данного подкласса подходящий автомат $\boldsymbol{M}$ является приведенным и сильно связным, то всякий исследуемый входо-выходной полуавтомат обязательно обладает установочной последовательностью, причем длина этой последовательности не превосходит $|S_1|(|S_1|-1)/2$. Исследование расширения множества таких "хороших" подклассов полуавтоматов является частью дальнейшей работы.

3. Заключение

В настоящей работе исследуется задача синтеза синхронизирующих и установочных последовательностей для специального класса входо-выходных полуавтоматов, в которых в каждом состоянии определены только входные или только выходные действия, и в диаграмме переходов отсутствуют циклы, помеченные только выходными действиями. Мы показываем, каким образом задачу синтеза синхронизирующей и установочной последовательностей для полуавтоматов из этого класса можно свести к решению подобных задач для классических автоматов и полуавтоматов, для которых такие алгоритмы существуют. Кроме того, в статье получены оценки сложности проверки существования и построения таких последовательностей. Описан один из классов входо-выходных полуавтоматов, для которых полученные оценки сложности могут быть понижены. В дальнейшем мы планируем продолжить исследование классов входо-выходных полуавтоматов с пониженными оценками сложности для синхронизирующих и установочных последовательностей, в частности, за счет использования адаптивных "умозрительных" экспериментов. Мы также планируем провести исследования по расширению класса рассматриваемых входо-выходных полуавтоматов, добавив ненаблюдаемое действие τ, а также допущение о существовании композитных состояний, в которых допустимы как входные, так и выходные действия.

Список литературы / References

[1] Мартюгин П. В., "Нижние оценки длины кратчайших бережно синхронизирующих слов для двух- и трёхбуквенных частичных автоматов", *Дискретн. анализ и исслед. опер.*, **15**:4 (2008), 44–56; [Martjugin P. V., "Nizhnie ocenki dliny kratchajshih berezhno

sinhronizirujushhih slov dlja dvuh- i trjohbukvennyh chastichnyh avtomatov", *Diskretn. analiz i issled. oper.*, **15**:4 (2008), 44–56, (in Russian).]

[2] Gill A., "State-identification experiments in finite automata", *Information and Control*, **4** (1961), 132–154.

[3] Martyugin P., "Computational Complexity of Certain Problems Related to Carefully Synchronizing Words for Partial Automata and Directing Words for Nondeterministic Automata", *Theory Comput. Syst.*, **54**:2 (2014), 293–304.

[4] Hierons R. M., Jourdan G. V., Ural H., Yenigün H., "Using adaptive distinguishing sequences in checking sequence constructions", *Proc. of the 2008 ACM symposium on Applied computing (SAC)*, ACM, 2008, 682–687.

[5] Ito M., Shikishima-Tsuji K., "Some Results on Directable Automata", *Theory Is Forever: Essays Dedicated to Arto Salomaa on the Occasion of His 70th Birthday*, Lecture Notes in Computer Science, **3113**, Springer, Berlin, Heidelberg, 2004, 125–133.

[6] Kohavi Z., *Switching and Finite Automata Theory*, McGraw-Hill, New York, 1978.

[7] Kushik N., El-Fakih K., Yevtushenko N., Cavalli A.R., "On adaptive experiments for nondeterministic finite state machines", *International Journal on Software Tools for Technology Transfer*, **18**:3 (2016), 251–264.

[8] Kushik N., López J., Cavalli A.R., Yevtushenko N., "Improving Protocol Passive Testing through 'Gedanken' Experiments with Finite State Machines", *Proc. 2016 IEEE International Conference on Software Quality, Reliability and Security (QRS)*, IEEE, 2016, 315–322.

[9] Kushik N., El-Fakih K., Yevtushenko N., "Preset and adaptive homing experiments for nondeterministic finite state machines", *Implementation and Application of Automata. CIAA 2011*, Lecture Notes in Computer Science, **6807**, 2011, 215–224.

[10] Kushik N., Yevtushenko N., "On the Length of Homing Sequences for Nondeterministic Finite State Machines", *Implementation and Application of Automata. CIAA 2013*, Lecture Notes in Computer Science, **7982**, 2013, 220–231.

[11] Lee D., Yannakakis M., "Testing finite-state machines: state identification and verification", *IEEE Trans. on Computers.*, **43**:3 (1994), 306–320.

[12] Moore E.F., "Gedanken-experiments on sequential machines", *Automata Studies (Annals of Mathematical Studies)*, Princeton University Press, 1956, 129–153.

[13] Sandberg S., "Homing and Synchronizing Sequences", *Model-Based Testing of Reactive Systems*, Lecture Notes in Computer Science, **3472**, 2005, 5–33.

[14] Tretmans J., "Test Generation with Inputs, Outputs and Repetitive Quiescence", *Software – Concepts and Tools*, **17**:3 (1996), 103–120.

[15] Volkov M.V., "Synchronizing Automata and the Černý Conjecture", *Language and Automata Theory and Applications. LATA 2008*, Lecture Notes in Computer Science, **5196**, 11–27.

[16] Kushik N.G, Yevtushenko N.V, Burdonov I.B., Kossatchev A.S, "Synchronizing and Homing Experiments for Input/output Automata", *System Informatics*, 2017, № 10, 1–9.

[17] Yenigün H., Yevtushenko N., Kushik N., "The complexity of checking the existence and derivation of adaptive synchronizing experiments for deterministic FSMs", *Information Processing Letters*, **127** (2017), 49–53.

Kushik N. G., Yevtushenko N. V., Burdonov I. B., Kossatchev A. S., "Deriving Synchronizing and Homing Sequences for Input/Output Automata", *Modeling and Analysis of Information Systems*, **24**:6 (2017), 730–742.

DOI: 10.18255/1818-1015-2017-6-730-742

Abstract. In this paper, we study the problem of existence check and derivation of synchronizing and homing sequences for finite input/output automata. Corresponding sequences can be effectively used for the current state identification of a system under test / verification, after the input sequence is applied. In the model considered in the paper, the alphabet of actions is divided into disjoint sets of inputs and outputs; however, no sets of possible initial or final states are defined. We introduce the notions of homing and synchronizing sequences for a specific class of such machines for which at each state the transitions only under inputs or under outputs are defined, and the machine transition diagram does not contain cycles labeled by outputs, i.e. the language of the machine does not contain traces with infinite postfix of outputs. For such a class of input/output automata, we establish necessary and sufficient conditions for the existence of synchronizing and homing sequences and discuss the length of such sequences. We also define some subclasses of automata for which the worst-case upper bounds (normally, exponential) are not reachable.

Keywords: input/output automata, synchronizing sequence, homing sequence

On the authors:
Natalia G. Kushik, PhD, Assistant Professor
Télécom SudParis/Université Paris-Saclay,
9 Charles Fourier str., Évry 91000, France, e-mail: natalia.kushik@telecom-sudparis.eu

Nina V. Yevtushenko, orcid.org/0000-0002-4006-1161, PhD, Professor,
Institute for System Programming RAS, 25 Solzhenitsyn str., Moscow 109004, Russia,
Tomsk State Univeristy, 36 Lenin str., Tomsk 634050, Russia, nyevtush@gmail.com

Igor B. Burdonov, orcid.org/0000-0001-9539-7853, PhD, Senior Researcher,
Institute for System Programming RAS,
25 Solzhenitsyn str., Moscow 109004, Russia, e-mail: igor@ispras.ru

Alexander S. Kossatchev, orcid.org/0000-0001-5316-3813, PhD, Senior Researcher,
Institute for System Programming RAS,
25 Solzhenitsyn str., Moscow 109004, Russia, e-mail: kos@ispras.ru

Acknowledgments:
The work was partially supported by the Russian Science Foundation (RSF), project № 16-49-03012 and № 17-07-00682 of Russian Foundation for Basic Research.