MODELING AND ANALYSIS OF INFORMATION SYSTEMS, VOL. 27, NO. 4, 2020
journal homepage: www.mais-journal.ru

THEORY OF COMPUTING

Deriving Homing Sequences for Finite State Machines with Timed Guards

A. S. Tvardovskii[1], N. V. Yevtushenko[2,3]

DOI: 10.18255/1818-1015-2020-4-376-395

[1]National Research Tomsk State University, 36 Lenin Ave, Tomsk 634050, Russia.
[2]Ivannikov Institute for System Programming of the RAS, 25 Alexander Solzhenitsyn St., Moscow 109004, Russia.
[3]National Research University Higher School of Economics, 20 Myasnitskaya St., Moscow 101000, Russia.

MSC2020: 68Q45
Research article
Full text in Russian

Received November 9, 2020
After revision November 30, 2020
Accepted December 16, 2020

State identification is the well-known problem in the theory of Finite State Machines (FSM) where homing sequences (HS) are used for the identification of a current FSM state, and this fact is widely used in the area of software and hardware testing and verification. For various kinds of FSMs, such as partial, complete, deterministic, non-deterministic, there exist sufficient and necessary conditions for the existence of preset and adaptive HS and algorithms for their derivation. Nowadays timed aspects become very important for hardware and software systems and for this reason classical FSMs are extended by clock variables. In this work, we address the problem of checking the existence and derivation of homing sequences for FSMs with timed guards and show that the length estimation for timed homing sequence coincides with that for untimed FSM. The investigation is based on the FSM abstraction of a Timed FSM, i.e. on a classical FSM which describes behavior of corresponding TFSM and inherits some of its properties. When solving state identification problems for timed FSMs, the existing FSM abstraction is properly optimized.

Keywords: Finite State Machine; timed guards; FSM abstraction; homing sequence

INFORMATION ABOUT THE AUTHORS

Aleksandr Sergeevich Tvardovskii correspondence author	orcid.org/0000-0001-7705-7214. E-mail: tvardal@mail.ru PhD.
Nina Vladimirovna Yevtushenko	orcid.org/0000-0002-4006-1161. E-mail: evtushenko@ispras.ru Doctor of technical sciences, professor.

Funding: The reported study was funded by RFBR, project number 19-07-00327.

For citation: A. S. Tvardovskii and N. V. Yevtushenko, “Deriving Homing Sequences for Finite State Machines with Timed Guards”, *Modeling and analysis of information systems*, vol. 27, no. 4, pp. 376-395, 2020.

МОДЕЛИРОВАНИЕ И АНАЛИЗ ИНФОРМАЦИОННЫХ СИСТЕМ, ТОМ 27, № 4, 2020
сайт журнала: www.mais-journal.ru

THEORY OF COMPUTING

Синтез установочных последовательностей для автоматов с временными ограничениями

А. С. Твардовский[1], Н. В. Евтушенко[2,3]

DOI: 10.18255/1818-1015-2020-4-376-395

[1]Национальный исследовательский Томский Государственный университет, пр. Ленина, д. 36, г. Томск, 634050 Россия.
[2]Институт системного программирования РАН, ул. А. Солженицына, д. 25, г. Москва, 109004 Россия.
[3]Национальный исследовательский университет «Высшая школа экономики», ул. Мясницкая, д. 20, г. Москва, 101000 Россия.

УДК 519.7
Научная статья
Полный текст на русском языке

Получена 9 ноября 2020 г.
После доработки 30 ноября 2020 г.
Принята к публикации 16 декабря 2020 г.

Идентификация состояний является хорошо известной задачей теории конечных автоматов, и установочные последовательности, которые позволяют идентифицировать текущее состояние конечного автомата, широко используются в областях тестирования и верификации программного и аппаратного обеспечения. Для автоматов различных классов, полностью определенных и частичных, детерминированных и недетерминированных, установлены необходимые и достаточные условия существования безусловных и адаптивных установочных последовательностей и предложены алгоритмы их синтеза, если такая последовательность существует. В настоящее время при верификации и тестировании программного и аппаратного обеспечения необходимо учитывать временные аспекты, что приводит к расширению автоматных моделей временными переменными. В настоящей работе мы исследуем задачи проверки существования и синтеза безусловных и адаптивных установочных последовательностей для автоматов с временными ограничениями и показываем, что оценки на длину таких последовательностей совпадают с оценками для классических конечных автоматов. Предлагаемый подход основан на использовании конечно-автоматной абстракции временного автомата, то есть описании временного автомата соответствующим конечным автоматом, который сохраняет свойства временного автомата относительно установочных последовательностей.

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

ИНФОРМАЦИЯ ОБ АВТОРАХ

Александр Сергеевич Твардовский автор для корреспонденции	orcid.org/0000-0001-7705-7214. E-mail: tvardal@mail.ru кандидат физико-математических наук.
Нина Владимировна Евтушенко	orcid.org/0000-0002-4006-1161. E-mail: evtushenko@ispras.ru доктор технических наук, профессор.

Финансирование: Исследование выполнено при финансовой поддержке РФФИ в рамках научного проекта № 19-07-00327.

Для цитирования: A. S. Tvardovskii and N. V. Yevtushenko, “Deriving Homing Sequences for Finite State Machines with Timed Guards”, *Modeling and analysis of information systems*, vol. 27, no. 4, pp. 376-395, 2020.

Введение

Проблемы идентификации состояний в автоматных моделях исследуются с середины 20-го века [1—5], в первую очередь, для классических конечных автоматов [1, 2]. Установочные, различающие и синхронизирующие последовательности, позволяющие идентифицировать состояния в конечно-автоматных моделях, широко используются в областях тестирования и верификации программного и аппаратного обеспечения [2, 4, 6—10]. Установочные и синхронизирующие последовательности позволяют определить состояние, достигнутое автоматом после подачи такой последовательности [3, 11—14]. Данный факт используется как для установки исследуемой системы в известное состояние при активном тестировании, так и для определения текущего состояния системы в ходе мониторинга при пассивном тестировании [8, 15]. В обоих случаях знание достигнутого автоматом состояния позволяет более эффективно провести дальнейший анализ исследуемой системы.

Задача синтеза установочных последовательностей формулируется для конечных автоматов, в которых вывод о достигнутом состоянии может быть сделан на основе соответствующей выходной реакции автомата после подачи установочной последовательности [8, 14, 16]. Понятие синхронизирующей последовательности формулируется для конечных полуавтоматов и отражает последовательность действий, которая переводит полуавтомат в известное состояние независимо от начального состояния полуавтомата [4, 14]. Задача синтеза установочных последовательностей хорошо изучена для классических конечных автоматов, однако недостаточно исследована для различных расширений таких автоматов.

Для различных классов классических конечных автоматов определены необходимые и достаточные условия существования безусловных [8, 11, 12, 14] и условных (адаптивных) [17—19] установочных последовательностей и предложены алгоритмы их синтеза, если такие последовательности существуют. Безусловная установочная последовательность представляет собой известную до начала эксперимента последовательность входных воздействий, по реакции на которую можно определить достигнутое автоматом состояние. Для адаптивной последовательности следующее входное воздействие зависит от реакции автомата на предыдущие входные воздействия, что усложняет проведение эксперимента, но позволяет сократить длину установочной последовательности в ряде случаев [17]. Также известно, что для недетерминированных конечных автоматов вероятность существования адаптивной установочной последовательности значительно выше, чем вероятность существования безусловной установочной последовательности.

Функционирование современного программного и аппаратного обеспечения часто зависит от временных аспектов, что мотивирует исследования в области временных автоматов. Существуют различные подходы к внесению временных аспектов в модель конечного автомата, такие как входные таймауты, выходные таймауты (задержки на переходах), входные и выходные временные ограничения [20—24]. Все эти подходы объединяет внесение в модель автомата временной переменной, от значения которой зависит поведение автомата. В настоящей работе мы исследуем задачу проверки существования и синтеза безусловных и адаптивных установочных последовательностей для автоматов с одной временной переменной и (входными) временными ограничениями [21, 23]. А именно, мы адаптируем классические конечно-автоматные алгоритмы для построения установочных последовательностей для полностью определенных временных автоматов.

Предлагаемый в работе подход основан на использовании конечно-автоматной абстракции [23], т.е. конечного автомата, в ряде случаев адекватно описывающего поведение временного автомата. Как показано в работе, такая абстракция позволяет проверить существование временной установочной последовательности с использованием классических алгоритмов и оценить длину кратчайшей установочной последовательности, если таковая существует. Известный подход к построению конечно-автоматной абстракции сопровождается существенным увеличением входного алфавита или множества состояний относительно исходного временного автомата [23]. Поскольку такая абстракция является избыточной при решении ряда задач [21, 25, 26], в том числе задач, связанных с установочными и тестовыми последовательностями, мы предлагаем подход к оптимизации конечно-автоматной абстракции.

Результаты данной работы были частично представлены на международном семинаре PSSV-2020 (XI Workshop Program Semantics, Specification and Verification: Theory and Applications) и опубликованы в [26]. В настоящей статье мы расширяем полученные результаты для адаптивных установочных последовательностей, а также формулируем более общий алгоритм оптимизации конечно-автоматной абстракции, используемой для проверки существования и построения установочных последовательностей.

Структура работы следующая. Раздел 1 содержит определения классического и временного автомата, а также описание существующего подхода к синтезу конечно-автоматной абстракции. В разделе 2 исследуется проблема существования и синтеза безусловных и адаптивных установочных последовательностей для автоматов с временными ограничениями. Раздел 3 описывает предлагаемый в работе подход к оптимизации конечно-автоматной абстракции при синтезе установочных последовательностей.

1. Основные определения и обозначения

1.1. Конечный автомат

Под конечным автоматом [1] понимается четвёрка $\mathsf{P} = (P, I, O, h_{\mathsf{P}})$, где P — конечное непустое множество состояний, I — конечное непустое множество входных символов (входной алфавит), O — конечное непустое множество выходных символов (выходной алфавит), $I \cap O = \varnothing$, $h_{\mathsf{P}} \subseteq (P \times I \times O \times P)$ — отношение переходов. Кортеж (p, i, o, p') описывает переход из состояния p в состояние p' под действием входного символа i с выдачей выходного символа o. В соответствии с введённым определением, мы рассматриваем так называемые неинициальные автоматы, в которых каждое состояние может быть начальным, если явно не указано обратное.

Если в автомате P для любой пары $(p, i) \in P \times I$ существует переход $(p, i, o, p') \in h_{\mathsf{P}}$, то автомат называется полностью определённым, иначе – частичным. Если в автомате P для любой пары $(p, i) \in P \times I$ существует не более одного перехода $(p, i, o, p') \in h_{\mathsf{P}}$, то автомат называется детерминированным, иначе – недетерминированным. Полностью определённый недетерминированный автомат называется наблюдаемым, если для каждой тройки $(p, i, o) \in P \times I \times O$ существует не более одного перехода $(p, i, o, p') \in h_{\mathsf{P}}$, т.е. следующее состояние автомата для пары (входной символ / выходной символ) определяется единственным образом.

В настоящей работе мы рассматриваем полностью определённые наблюдаемые, возможно, недетерминированные конечные автоматы. Исключением являются тестовые примеры, которые используются для представления адаптивных входных последовательностей. Тестовые примеры, которые представляют собой инициальные наблюдаемые автоматы специального вида, являются частичными автоматами при $|I| > 1$ и подробно рассматриваются в разделе 2.2.

Для автомата P допустимая последовательность пар (входной символ / выходной символ) $i_1/o_1, i_2/o_2, \ldots, i_n/o_n$ в состоянии p_1 называется вход-выходной последовательностью в состоянии p_1, если в P существует соответствующая последовательность переходов $(p_1, i_1, o_1, p_2), (p_2, i_2, o_2, p_3), \ldots, (p_n, i_n, o_n, p_{n+1})$. При этом $\alpha = i_1, i_2, \ldots, i_n$ — входная последовательность, а $\gamma = o_1, o_2, \ldots, o_n$ — соответствующая выходная последовательность (реакция автомата). Вход-выходная последовательность в этом случае может быть обозначена как α/γ, и в автомате существует переход $(p_1, \alpha, \gamma, p_{n+1})$.

Для состояния p и вход-выходной пары i/o вводится понятие i/o-преемника состояния p: i/o-преемник состояния p в автомате P содержит каждое состояние p' такое, что в автомате существует переход (p, i, o, p'). Для вход-выходной последовательности α/γ, α/γ-преемником состояния p автомата P называется множество всех состояний p', в которые автомат переходит из состояния p по вход-выходной последовательности α/γ. Заметим, что α/γ-преемник состояния p может быть пустым, если в автомате не существует ни одного перехода вида (p, α, γ, p'), и в этом случае мы иногда говорим, что α/γ-преемник состояния p не существует. В наблюдаемом полностью определенном автомате α/γ-преемник состояния p является одноэлементным или не существует. Для подмножества P' состояний автомата, α/γ-преемник определяется как объединение α/γ-преемников состояний из P'. Автомат $\mathsf{Q} = (Q, I_Q, O_Q, h_\mathsf{Q})$ является подавтоматом автомата $\mathsf{P} = (P, I, O, h_\mathsf{P})$, если $Q \subseteq P, I_Q \subseteq I, O_Q \subseteq O$ и $h_\mathsf{Q} \subseteq h_\mathsf{P}$. В данной работе мы рассматриваем частный случай подавтоматов, в которых $Q = P$.

1.2. Автомат с временными ограничениями

В настоящей работе мы рассматриваем автомат с временными ограничениями, который представляет собой конечный автомат, дополненный одной временной переменной и временными интервалами на переходах [21, 23, 24].

Под автоматом с временными ограничениями понимается четвёрка $\mathsf{S} = (S, I, O, \lambda_\mathsf{S})$, где S — конечное непустое множество состояний, I — входной алфавит, O — выходной алфавит, $\lambda_\mathsf{S} \subseteq (S \times I \times O \times S \times \Pi)$ – конечное отношение переходов, Π – множество интервалов из промежутка $[0; \infty)$ вида $\langle a, b \rangle$, где $\langle \in \{[, (\}, \rangle \in \{],)\}$, a и b — целые неотрицательные числа или ∞. Соответственно, кортеж (s, i, o, s', g) описывает переход из состояния s в состояние s' под действием входного символа i, поступившего в момент времени t, $t \in g$, после перехода автомата в текущее состояние, с выдачей выходного символа o. Временная переменная увеличивает своё значение при ожидании входного символа и сбрасывается в 0 при выполнении перехода автоматом.

Временной автомат S называется полностью определённым, если поведение автомата в любом состоянии определено для любого входного символа, и для любой пары $(s, i) \in S \times I$, то есть для любого входного символа i, поступающего на вход автомата в состоянии s, объединение всех временных интервалов g из переходов $(s, i, o, s', g) \in \lambda_\mathsf{S}$ равно $[0; \infty)$. Аналогично классическим автоматам, частичный автомат может быть доопределён до полностью определённого автомата различными способами в зависимости от интерпретации неопределенных переходов [1, 27]. Если для любых двух кортежей $(s, i, o_1, s_1, g_1), (s, i, o_2, s_2, g_2) \in \lambda_\mathsf{S}$ справедливо, что $g_1 \bigcap g_2 = \varnothing$, то временной автомат называется детерминированным, иначе – недетерминированным. Полностью определённый недетерминированный временной автомат называется наблюдаемым, если для каждой пары переходов $(s, i, o, s_1, g_1), (s, i, o, s_2, g_2) \in \lambda_\mathsf{S}$, такой что $g_1 \bigcap g_2 \neq \varnothing$, справедливо, что $s_1 = s_2$, т.е. следующее состояние определяется единственным образом по соответствующей выходной реакции.

Временным входным символом называется пара (i, t), где i – входной символ, t – текущее значение временной переменной, т.е. время поступления входного символа, отсчитываемое от момента выдачи автоматом последнего выходного символа или начала эксперимента.

Для временного автомата S последовательность временных входных символов $\alpha = (i_1, t_1), (i_2, t_2), \dots, (i_n, t_n)$ называется временной входной последовательностью. Реакция на временную входную последовательность α автомата S в состоянии s_1 вычисляется итеративно и представляет собой последовательность выходных символов γ. Для входного символа (i_1, t_1) автоматом выполняется переход $(s_1, i_1, o_1, s_2, g_1)$, для которого $t_1 \in g_1$; после выполнения перехода временная переменная сбрасывается в 0. Процедура повторяется в достигнутом состоянии s_2 со следующим временным входным символом (i_2, t_2) и так далее. Аналогично классическим автоматам, последовательность пар (временной входной символ / выходной символ) называется временной вход-выходной последовательностью автомата S в состоянии s_1, если в S существует соответствующая последовательность переходов $(s_1, i_1, o_1, s_2, g_1), (s_2, i_2, o_2, s_3, g_2), \dots, (s_n, i_n, o_n, s_{n+1}, g_n)$, где $t_i \in g_i$, и такая последовательность обозначается $\alpha/\gamma = (i_1, t_1)/o_1 \dots (i_n, t_n)/o_n$.

Например, если на временной автомат S (рисунок 1) в состоянии s_1 подается входной символ i_1 при значении временной переменной 1.7, т.е. на автомат поступает временной входной символ $(i_1, 1.7)$, то автомат перейдёт в состояние s_3 с выходным символом o_2, после чего временная переменная «сбросится» в 0.

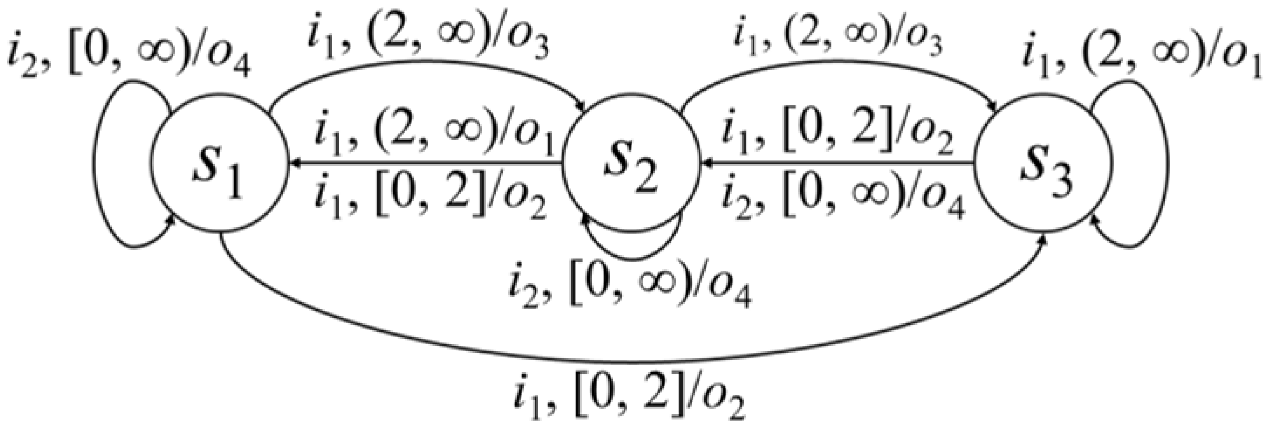

Fig. 1. FSM with timed guards S

Рис. 1. Автомат с временными ограничениями S

Понятие α/γ-преемника для временных автоматов определяется аналогично классическим автоматам. По определению, $(i, t)/o$-преемник состояния s содержит каждое состояние s' такое, что в автомате существует переход (s, i, o, s', g), где $t \in g$. Для временной вход-выходной последовательности α/γ, α/γ-преемником состояния s временного автомата S называется множество всех состояний s', в которые автомат переходит из состояния s по последовательности α/γ. В наблюдаемом автомате α/γ-преемник состояния s является одноэлементным или не существует. Для подмножества S' состояний временного автомата α/γ-преемник определяется как объединение α/γ-преемников состояний из S'.

В настоящей работе мы рассматриваем задачу синтеза безусловных и адаптивных установочных последовательностей для полностью определённых наблюдаемых, возможно недетерминированных автоматов с временными ограничениями.

1.3. Конечно-автоматные абстракции временного автомата

Известно [23], что поведение временных автоматов в ряде случаев можно адекватно описать при помощи конечного автомата, т.е. с использованием некоторой конечно-автоматной абстракции. Конечно-автоматная абстракция полностью определенного автомата с временными ограничениями $\mathsf{S} = (S, I, O, \lambda_{\mathsf{S}})$ с наибольшей конечной границей для временных входных интервалов B_{S} определяется как полностью определённый конечный автомат $\mathsf{A}_{\mathsf{S}} = (S, I_A, O, h_{\mathsf{AS}})$, где $I_A = \{(i, [0, 0]), (i, (0, 1)), \dots, (i, (B_{\mathsf{S}} - 1, B_{\mathsf{S}})), (i, [B_{\mathsf{S}}, B_{\mathsf{S}}]), (i, (B_{\mathsf{S}}, \infty)) : i \in I\}$. Для состояния s автомата A_{S} и входного символа (i, g_j), множество h_{AS} содержит переход $(s, (i, g_j), o, s')$, если и только если существует переход $(s, i, o, s', g) \in \lambda_{\mathsf{S}}$, такой что $g_j \subseteq g$. Таким образом, при построении абстракции к каждому входному символу добавляются открытые интервалы времени длины 1 и целочисленные значения временной переменной.

Для наблюдаемого недетерминированного автомата S на рисунке 1, соответствующая конечно-автоматная абстракция представлена на рисунке 2. Строки таблицы соответствуют входным символам, столбцы – состояниям, в ячейке на пересечении состояния s и входного символа i содержится множество пар (выходной символ o / io-преемник s'), таких что $(s, i, o, s') \in h_{AS}$.

Временная входная последовательность α для автомата с временными ограничениями может быть переведена в соответствующую последовательность входных символов α_{FSM} для конечно-автоматной абстракции. При этом каждый временной входной символ (i, t) представляется входным символом (i, t), если t целое, либо символом $(i, (a, b))$, где $t \in (a, b)$, в противном случае. Аналогично, входная последовательность конечно-автоматной абстракции может быть переведена во временную входную последовательность. При этом, для входного символа $(i, (a, b))$ значением временной переменной соответствующего входного временного символа может быть выбрано любое значение $t \in (a, b)$.

i/s	s_1	s_2	s_3
$(i_1, [0, 0])$	s_3/o_2	s_1/o_2	s_2/o_2
$(i_1, (0, 1))$	s_3/o_2	s_1/o_2	s_2/o_2
$(i_1, [1, 1])$	s_3/o_2	s_1/o_2	s_2/o_2
$(i_1, (1, 2))$	s_3/o_2	s_1/o_2	s_2/o_2
$(i_1, [2, 2])$	s_3/o_2	s_1/o_2	s_2/o_2
$(i_1, (2, \infty))$	s_2/o_3	$s_1/o_1, s_3/o_3$	s_3/o_1

i/s	s_1	s_2	s_3
$(i_2, [0, 0])$	s_1/o_4	s_2/o_4	s_2/o_4
$(i_2, (0, 1))$	s_1/o_4	s_2/o_4	s_2/o_4
$(i_2, [1, 1])$	s_1/o_4	s_2/o_4	s_2/o_4
$(i_2, (1, 2))$	s_1/o_4	s_2/o_4	s_2/o_4
$(i_2, [2, 2])$	s_1/o_4	s_2/o_4	s_2/o_4
$(i_2, (2, \infty))$	s_1/o_4	s_2/o_4	s_2/o_4

Fig. 2. Flow table of the FSM abstraction of the TFSM in figure 1

Рис. 2. Конечно-автоматная абстракция временного автомата на рисунке 1

Отметим некоторые основные свойства конечно-автоматной абстракции [23].

- Конечно-автоматная абстракция полностью определённого временного автомата является полностью определённым автоматом.
- Конечно-автоматная абстракция детерминированного (недетерминированного) временного автомата является детерминированным (недетерминированным) автоматом.
- Конечно-автоматная абстракция наблюдаемого (ненаблюдаемого) временного автомата является наблюдаемым (ненаблюдаемым) автоматом.

Такая конечно-автоматная абстракция достаточно полно описывает поведение временного автомата и, в частности, может быть использована при синтезе тестов с гарантированной полнотой [25]. В [23], было доказано утверждение о соответствии между вход-выходными последовательностями автомата с временными ограничениями и его конечно-автоматной абстракции для инициальных автоматов, в которых выделено единственное начальное состояние. Аналогичное утверждение можно сформулировать и для неинициальных временных автоматов.

Утверждение 1. Временная вход-выходная последовательность α/γ существует в состоянии s неинициального временного автомата S, если и только если для конечно-автоматной абстракции A_S существует вход-выходная последовательность α_{FSM}/γ в состоянии s.

2. Установочные последовательности для временных автоматов

Установочная последовательность позволяет по реакции автомата установить достигнутое по такой последовательности состояние, независимо от начального состояния. Такие последовательности могут быть безусловными, если следующий входной символ не зависит от реакции автомата на предыдущие входные воздействия, или адаптивными, если следующий входной символ зависит от реакции автомата на предыдущие входные символы. Для детерминированных полностью определенных приведенных сильно связных автоматов всегда существует безусловная установочная последовательность, причем длина кратчайшей установочной последовательности для автомата с n состояниями не превосходит величины $n(n-1)/2$, и переход к адаптивным установочным последовательностям не позволяет сократить длину последовательности. В то же время, для недетерминированных автоматов безусловная установочная последовательность существует не всегда, и такая кратчайшая последовательность может иметь экспоненциальную длину относительно числа состояний автомата. Для недетерминированных автоматов адаптивные установочные последовательности существуют чаще, чем безусловные [28], и в ряде случаев имеют меньшую длину. В следующих разделах мы адаптируем известные конечно-автоматные методы построения установочных последовательностей к построению таких последовательностей для временных автоматов.

2.1. Безусловные установочные последовательности для временных автоматов

Временная входная последовательность α является установочной для полностью определенного временного автомата $\mathsf{S} = (S, I, O, \lambda_\mathsf{S})$, если для любой выходной реакции γ на α, α/γ-преемник множества S является одноэлементным или не существует, т.е. является пустым множеством. Для наблюдаемых полностью определённых конечных автоматов установочная последовательность может быть построена по усечённому дереву преемников [1, 12, 13].

При рассмотрении временного автомата необходимо учитывать, что автомат может не только находиться в произвольном состоянии, но и обладать любым значением временной переменной в начале эксперимента. Тем не менее, поскольку любое входное воздействие сбрасывает значение временной переменной в 0, то достаточно рассмотреть проблему синтеза установочной последовательности в предположении, что временная переменная равна 0 в момент начала установочного эксперимента.

Утверждение 1 и взаимно однозначное соответствие между состояниями временного автомата и его конечно-автоматной абстракцией позволяет сформулировать следующее утверждение.

Утверждение 2. Временная входная последовательность α является установочной последовательностью для полностью определённого автомата с временными ограничениями S, если и только если соответствующая входная последовательность α_{FSM} является установочной последовательностью для конечно-автоматной абстракции A_S.

Доказательство. Пусть α является установочной последовательностью для автомата с временными ограничениями $\mathsf{S} = (S, I, O, \lambda_\mathsf{S})$. По определению, α является установочной последовательностью для S, если и только если для каждой вход-выходной последовательности $\alpha/\gamma = (i_1, t_1)/o_1 \ldots (i_k, t_k)/o_k$ справедливо, что α/γ-преемник S является одноэлементным или не существует. По определению конечно-автоматной абстракции и в силу утверждения 1, в автомате S существуют вход-выходная последовательность α/γ и последовательность переходов $(s_1, i_1, o_1, s_2, g_1), \ldots, (s_k, i_k, o_k, s', g_k) \in \lambda_\mathsf{S}$ если и только если в конечно-автоматной абстракции A_S существуют вход-выходная последовательность α_{FSM}/γ и последовательность переходов $(s, (i_1, g'_1), o_1, s_1), \ldots, (s_k, (i_k, g'_k), o_k, s') \in h_{\mathsf{A}\mathsf{S}}$, где $g'_j \subseteq g_j$ и $1 \leq j \leq k$. Соответственно, в конечно-автоматной абстракции A_S каждый α_{FSM}/γ-преемник множества S является одноэлементным множеством или не существует, если и только если α_{FSM} является установочной последовательностью для A_S.

Следовательно, установочная последовательность для автомата с временными ограничениями может быть построена как установочная последовательность для соответствующей конечно-автоматной абстракции. Далее входная последовательность абстракции преобразуется во временную входную последовательность.

Оценим длину установочных последовательностей для автоматов с временными ограничениями, которая определяется как количество временных входных символов в последовательности.

Состояния s и p полностью определённых детерминированных (временных) автоматов S и P называются эквивалентными, если реакции автоматов в этих состояниях совпадают на любую (временную) входную последовательность, в противном случае, состояния s и p различимы. Полностью определенный детерминированный автомат называется приведённым, если любые два различные его состояния не являются эквивалентными. Полностью определенный детерминированный автомат называется сильно связным, если для любой пары состояний s_1 и s_2 существует последовательность α, такая что α/γ-преемник состояния s_1 есть состояние s_2.

Известно [1], что для полностью определённых детерминированных приведённых сильно связных конечных автоматов установочная последовательность всегда существует и имеет длину не более $n(n-1)/2$, где n – число состояний автомата. Это свойство выполняется и для временного автомата.

Утверждение 3. Пусть S – полностью определённый детерминированный автомат с временными ограничениями.

1) Автомат S является сильно связным, если и только если конечно-автоматная абстракция $\mathsf{A_S}$ является сильно связным автоматом.
2) Автомат S является приведённым по состояниям, если и только если конечно-автоматная абстракция $\mathsf{A_S}$ является приведённым конечным автоматом.

Доказательство.

1) По определению конечно-автоматной абстракции $\mathsf{A_S}$, между множеством её состояний и множеством состояний соответствующего временного автомата S существует взаимно однозначное соответствие, более того, автомат $\mathsf{A_S}$ содержит переход $(s, (i, g_i), o, s') \in h_{\mathsf{AS}}$ если и только если $(s, i, o, s', g) \in \lambda_{\mathsf{S}}$, где $g_i \subseteq g$. Соответственно, временная входная последовательность α переводит автомат с временными ограничениями S из состояния s в состояние s', если и только если входная последовательность α_{FSM} переводит автомат $\mathsf{A_S}$ из состояния s в состояние s'.
2) Автомат с временными ограничениями S приведённый, если и только если для каждой пары различных состояний $s_1, s_2 \in S$ существует входная временная последовательность $\alpha(s_1, s_2)$, различающая эти состояния. Различающая временная входная последовательность $\alpha(s_1, s_2)$ существует для каждой пары состояний $s_1, s_2 \in S$, если и только если существует различающая последовательность $\alpha(s_1, s_2)_{FSM}$ для состояний s_1 и s_2 конечно-автоматной абстракции $\mathsf{A_S}$ (утверждение 1). Следовательно, различающая последовательность $\alpha(s_1, s_2)_{FSM}$ существует для каждой пары различных состояний конечно-автоматной абстракции $\mathsf{A_S}$, если и только если $\mathsf{A_S}$ является приведённым автоматом.

Поскольку между состояниями временного автомата и состояниями соответствующей конечно-автоматной абстракции существует взаимно однозначное соответствие, то на основе утверждений 2 и 3, можно сформулировать следующее утверждение о длине установочных последовательностей для временных автоматов.

Утверждение 4. Пусть S – полностью определённый детерминированный приведённый сильно связный автомат с временными ограничениями. Установочная последовательность всегда существует для автомата S и длина кратчайшей установочной последовательности не превышает $n(n-1)/2$, где n – количество состояний автомата.

Действительно, поскольку конечно-автоматная абстракция $\mathsf{A_S}$ полностью определённого детерминированного приведённого сильно связного автомата с временными ограничениями S является полностью определённым детерминированным приведённым и сильно связным автоматом (утверждения 1 и 3), то и $\mathsf{A_S}$, и соответствующий временной автомат S обладают установочной последовательностью длины не больше $n(n-1)/2$.

Таким образом, установочная последовательность всегда существует для полностью определённого детерминированного приведённого сильно связного автомата с временными ограничениями. Более того, поскольку верхняя оценка не может быть уменьшена для конечных детерминированных автоматов [3], эту оценку нельзя уменьшить и для временных детерминированных автоматов с соответствующими свойствами.

Отметим, что для недетерминированных конечных автоматов длина кратчайшей безусловной установочной последовательности может превышать величину $2^{n-1}-1$ [13], и, таким образом, для полностью определённого недетерминированного наблюдаемого автомата с временными ограничениями, который имеет n состояний, длина кратчайшей безусловной установочной последовательности может также превышать величину $2^{n-1}-1$.

Аналогично результатам для классических конечных автоматов, установочная последовательность не всегда существует для недетерминированных наблюдаемых полностью определённых автоматов с временными ограничениями. Проверка существования и синтез установочных последовательностей для таких временных автоматов могут быть выполнены аналогично классическому алгоритму на основе построения усечённого дерева преемников [13].

Алгоритм 1 проверки существования и построения установочных последовательностей для автомата с временными ограничениями.

Вход: полностью определённый наблюдаемый возможно недетерминированный автомат с временными ограничениями $\mathsf{S} = (S, I, O, \lambda_{\mathsf{S}})$.

Выход: установочная последовательность α для временного автомата S или сообщение "Для автомата S не существует установочной последовательности".

1. Строится конечно-автоматная абстракция $\mathsf{A_S} = (S, I_A, O, h_{\mathsf{AS}})$ для временного автомата S.
2. Для автомата $\mathsf{A_S}$ строится усечённое дерево преемников. Каждая вершина дерева преемников помечается подмножеством пар различных состояний; рёбра дерева помечаются входными символами. Корню дерева ставится в соответствие множество всех пар различных состояний. Из вершины, помеченной множеством P уровня $j, j \geq 0$, существует ребро, помеченное входным символом i, в вершину уровня $(j+1)$, помеченную множеством Q, где Q содержит пару состояний s_1, s_2, если и только если состояния s_1 и s_2 являются i/o-преемниками некоторой пары состояний из P для некоторого выходного символа o. Если таких пар нет, то вершина уровня $(j+1)$ помечается пустым множеством. Вершина уровня $k, k > 0$, помеченная множеством P, является листом, если множество P пусто (Правило усечения 1) или P содержит непустое множество R, помечающее вершину уровня $j, j < k$ (Правило усечения 2).
3. Если построенное дерево преемников не имеет вершин, помеченных пустым множеством, то выдается сообщение "Для автомата S не существует установочной последовательности". Иначе, кратчайшая последовательность входных символов α_{FSM}, помечающая путь из корня дерева в помеченную пустым множеством вершину, преобразуется в соответствующую временную входную последовательность, которая является кратчайшей установочной последовательностью для автомата S.

Корректность алгоритма 1 следует непосредственно из утверждения 2.

Для конечно-автоматной абстракции $\mathsf{A_S}$ (рисунок 2) временного автомата S (рисунок 1) соответствующее дерево преемников для подавтомата с входными символами, покрашенными серым цветом, представлено на рисунке 3. Непосредственной проверкой можно убедиться, что кратчайшей последовательностью входных символов, помечающей путь из корня дерева в вершину с пустым множеством, является последовательность $(i_1, (2, \infty)), (i_1, (2, \infty)), (i_1, (2, \infty))$ длины 3, которая может быть преобразована во временную установочную последовательность $(i_1, 2.1), (i_1, 2.3), (i_1, 2.2)$. Таким образом, эта последовательность является установочной последовательностью для временного автомата на рисунке 1. Обоснованность использования именно такого подавтомата при построении кратчайшей установочной последовательности обсуждается в разделе 3.

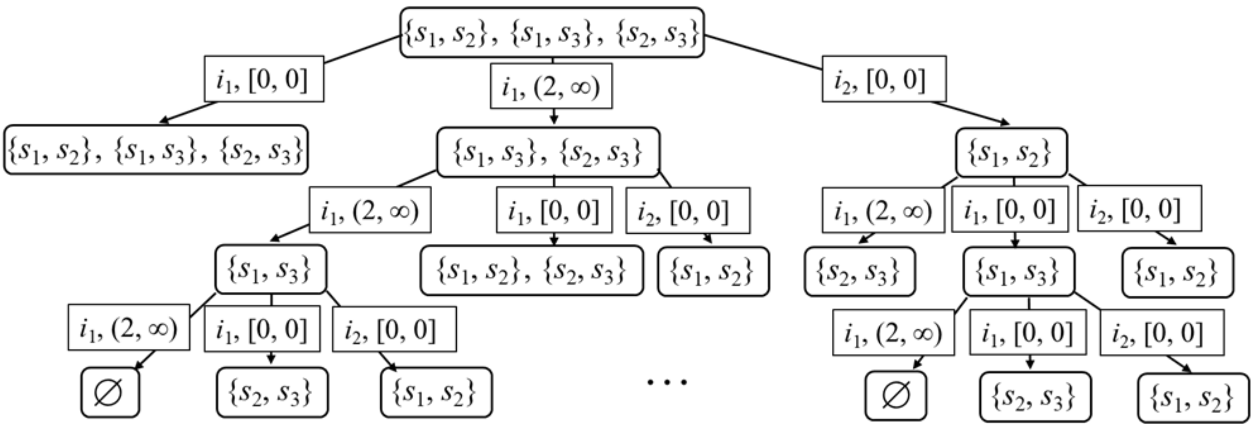

Fig. 3. The successor tree for a submachine of the FSM abstraction $\mathsf{A_S}$ in figure 2

Рис. 3. Дерево преемников для подавтомата конечно-автоматной абстракции $\mathsf{A_S}$ на рисунке 2

2.2. Адаптивные установочные последовательности

Безусловные установочные последовательности не всегда существуют для недетерминированных автоматов, и длина кратчайшей безусловной установочной последовательности (если таковая существует) может достигать экспоненциальной величины. Однако для неинициальных наблюдаемых полностью определенных автоматов, которые рассматриваются в настоящей статье, задача проверки существования и синтеза адаптивной установочной последовательности имеет полиномиальную сложность [13]. Для таких автоматов адаптивная установочная последовательность может существовать и в том случае, когда отсутствует безусловная установочная последовательность [28, 29]. Более того, в ряде случаев, кратчайшая адаптивная последовательность может оказаться короче, чем безусловная установочная последовательность. В настоящем разделе мы исследуем задачу синтеза адаптивных установочных последовательностей для временных автоматов.

2.2.1. Установочный тестовый пример

Для адаптивных последовательностей следующий входной символ может зависеть от реакции автомата на предыдущий входной символ и в данном разделе мы вводим понятие адаптивной установочной последовательности для временных автоматов. Адаптивная входная последовательность часто представляется тестовым примером [28, 29]. Пусть I и O — входной и выходной алфавиты некоторого конечного автомата P, тестовый пример $\mathsf{TC}(I, O)$ для автомата P представляет собой конечный автомат $\mathsf{TC}(I, O) = (T, I, O, h_{\mathsf{TC}}, t_0)$ с ацикличным графом переходов, в каждом состоянии которого либо определен переход только по одному входному символу со всеми выходными символами, либо не определён ни один переход; состояние без исходящих переходов называется терминальным. Тестовый пример называется установочным [28] для полностью определённого возможно недетерминированного конечного автомата $\mathsf{P} = (P, I, O, h_{\mathsf{P}})$, если для любой вход-выходной последовательности α/γ тестового примера $\mathsf{TC}(I, O)$, переводящей автомат из начального состояния в терминальное, α/γ-преемник множества всех состояний P автомата P является одноэлементным или не существует.

Аналогично синтезу безусловных установочных последовательностей, понятие тестового примера для автомата с временными ограничениями может быть введено в предположении, что временная переменная имеет нулевое значение в начале эксперимента, поскольку может быть сброшена в 0 подачей произвольного входного символа. Под временным тестовым примером для автомата с временными ограничениями $\mathsf{S} = (S, I, O, \lambda_{\mathsf{S}})$ будем понимать тестовый пример $\mathsf{TTC}(I_A, O) = (T, I_A, O, h_{\mathsf{TTC}}, q_0)$ над входным алфавитом $I_A = \{I \times [a, a], (a, a+1), [B_{\mathsf{S}}, B_{\mathsf{S}}], (B_{\mathsf{S}}, \infty) : 0 \le a < B_{\mathsf{S}}\}$ и выходным алфавитом O. Временной тестовый пример $\mathsf{TTC}(I_A, O)$ представляет адаптивную входную последовательность для временного автомата $\mathsf{S} = (S, I, O, \lambda_{\mathsf{S}})$, которая может быть подана на автомат S следующим образом. Пусть входной символ (i_1, g) определён в начальном состоянии q_0 автомата $\mathsf{TTC}(I_A, O)$, тогда первым подаётся временной входной символ (i_1, t), где $t \in g$, и автомат $\mathsf{TTC}(I_A, O)$ переходит в $(i_1, g)/o$-преемник q_1 начального состояния q_0 в соответствии с выходным символом o, выданным автоматом S. Процедура продолжается до тех пор, пока тестовый пример не достигнет терминального состояния. Временной тестовый пример $\mathsf{TTC}(I_A, O)$ называется установочным, т.е. представляет адаптивную установочную последовательность для временного автомата S, если для каждой вход-выходной последовательности $\alpha_{FSM}/\gamma = (i_1, g_1)/o_1 \ldots (i_k, g_k)/o_k$ тестового примера $\mathsf{TTC}(I_A, O)$ из начального состояния в терминальное и любых $t_j \in g_j, j = 1 \ldots k, \alpha/\gamma = (i_1, t_1)/o_1 \ldots (i_k, t_k)/o_k, \alpha/\gamma$-преемник множества состояний S во временном автомате S является одноэлементным или не существует. Таким образом, после подачи адаптивной установочной последовательности достигнутое автоматом состояние может быть однозначно определено.

Отметим, что для автомата с временными ограничениями S, временной тестовый пример $\mathsf{TTC}(I_A, O)$ имеет входной алфавит конечно-автоматной абстракции $\mathsf{A_S}$. Более того, по определению временного тестового примера установочный тестовый пример конечно-автоматной абстракции является установочным для исходного временного автомата. Таким образом, справедливо следующее утверждение.

Утверждение 5. Пусть $\mathsf{S} = (S, I, O, \lambda_{\mathsf{S}})$ – полностью определённый недетерминированный наблюдаемый автомат с временными ограничениями. Временной тестовый пример $\mathsf{TTC}(I_A, O)$ является установочным для автомата S, если и только если он является установочным тестовым примером для конечно-автоматной абстракции $\mathsf{A_S}$.

Действительно, каждой вход-выходной последовательности α_{FSM}/γ тестового примера $\mathsf{TTC}(I_A, O)$ из начального состояния в терминальное, для любого состоянии s временного автомата S и его абстракции $\mathsf{A_S}$ существуют как временная вход-выходная последовательность α/γ, так и соответствующая вход-выходная последовательность α_{FSM}/γ автомата $\mathsf{A_S}$ (в силу утверждения 1). Далее, аналогично утверждению 2, можно показать, что, для каждой вход-выходной последовательности α_{FSM}/γ тестового примера $\mathsf{TTC}(I_A, O)$ из начального состояния в терминальное, α/γ-преемник множества состояний S является одноэлементным или не существует во временном автомат S, если и только если α_{FSM}/γ-преемник множества состояний S является одноэлементным или не существует в конечно-автоматной абстракции $\mathsf{A_S}$.

Для слабо инициальных автоматов, в которых не каждое состояние может быть начальным, длина кратчайшей адаптивной установочной последовательности может достигать экспоненциальной величины относительно числа состояний автомата [30]. Однако известно, что для неинициального полностью определенного наблюдаемого автомата с n состояниями, т.е. автомата, в котором каждое состояние может быть начальным, длина кратчайшей адаптивной установочной последовательности не превосходит величины n^3 [17].

Иными словами, имеет место следующее утверждение.

Утверждение 6. Если S – неинициальный полностью определённый недетерминированный наблюдаемый автомат с временными ограничениями, который имеет n состояний и обладает адаптивной установочной последовательностью, то длина кратчайшей адаптивной установочной последовательности для автомата S не превосходит величины n^3.

В следующем подразделе мы адаптируем алгоритм из [28] к нахождению кратчайшей адаптивной установочной последовательности для полностью определенных наблюдаемых неинициальных временных автоматов.

2.2.2. Синтез адаптивных временных установочных последовательностей

Адаптивная установочная последовательность может быть построена по соответствующему установочному автомату [28, 31], и далее мы адаптируем данный подход к синтезу кратчайшей временной адаптивной установочной последовательности для автомата с временными ограничениями S на основе конечно-автоматной абстракции.

Для полностью определённого наблюдаемого, возможно недетерминированного конечного автомата $\mathsf{P} = (P, I, O, h_{\mathsf{P}}), |P| > 1$, и натурального числа $L \geq 0$ установочный автомат P^L_{home} [28, 31] строится итеративно, начиная с P^0_{home}. Входной и выходной алфавиты P^L_{home} совпадают с таковыми для автомата P, в то время как состояния P^L_{home} соответствуют подмножествам состояний автомата P мощности не менее двух; также в P^L_{home} добавляется специальное состояние F. Установочный автомат P^0_{home} состоит из начального состояния P, содержащего все состояния автомата P, и специального состояния F. Переходы из состояний P и F ведут в состояние F по всем вход-выходным парам. Начальное состояние P является помеченным состоянием.

Пусть автомат $\mathsf{P}^l_{home}, l \geq 0$, с помеченными состояниями уже построен, и b – помеченное состояние автомата P^l_{home}, соответствующее в автомате P подмножеству состояний b мощности не менее двух, все переходы из которого ведут в состояние F; эти переходы удаляются при построении автомата P^{l+1}_{home}. Новые переходы добавляются по следующим правилам:

- В автомате P^{l+1}_{home} существует переход (b, i, o, F), если и только если существует $o' \in O$, такой что непустой io'-преемник множества b совпадает с b.
- Если для любого $o' \in O$ непустой io'-преемник множества b не совпадает с b, то в автомате P^{l+1}_{home} существует переход (b, i, o, b'), если и только если b' является io-преемником множества b мощности не менее двух. Если b' не является состоянием автомата P^l_{home}, то b' добавляется в множество состояний автомата P^{l+1}_{home}, становится помеченным состоянием в автомате P^{l+1}_{home}, и в автомат P^{l+1}_{home} добавляются переходы из состояния b' в состояние F по каждой вход-выходной паре.
- Переход из состояния b по входному символу i считается неопределённым, если и только если каждый io-преемник множества b не существует или является одноэлементным.
- После определения всех переходов состояние b становится непомеченным в автомате P^{l+1}_{home}.

Если автомат P^{l+1}_{home} совпадает с автоматом P^l_{home}, то автомат P^l_{home} совпадает с автоматом P^k_{home} для любого $k > l$.

Автомат P обладает адаптивной установочной последовательностью длины $L > 0$, если и только если автомат P^L_{home} не имеет полностью определённого подавтомата [28]. Проверить автомат P^L_{home} на наличие полностью определённого подавтомата можно путём итеративного удаления из P^L_{home} состояний с неопределёнными входными символами. При удалении состояния b один из неопределённых входных символов отмечается как $i(b)$. Состояния с неопределёнными переходами удаляются до тех пор, пока либо в установочном автомате не закончатся такие состояния, либо в начальном состоянии P^L_{home} появится неопределённый входной символ. Если в начальном состоянии установочного автомата P^L_{home} появился неопределённый входной символ, то автомат P не имеет полностью определённого подавтомата. Соответствующий установочный тестовый пример T высоты L может быть построен на основе сохранённых неопределённых входных символов автомата P^L_{home}. В этом случае начальное состояние t_0 автомата T соответствует начальному состоянию P автомата P^L_{home}. Входной символ $i(P), P = b$, представляет собой единственный входной символ, определённый в состоянии $t = b$ автомата T, и переход $(t, i(b), o, t')$ существует в T, если и только если в P^L_{home} (до удаления состояний) существует переход $(b, i(b), o, b')$, где $t' = b'$. Если для некоторого $o \in O$ не существует перехода $(b, i(b), o, b')$ в автомате P^L_{home}, то в T добавляется переход $(t, i(b), o, D)$, где D – тупиковое состояние. Переходы для других состояний установочного тестового примера T формируются аналогично. Состояние T становится терминальным, если не имеет исходящих переходов. Если автомат P^L_{home} обладает полностью определенным подавтоматом и $L = n^3$, то адаптивной установочной последовательности для автомата P не существует.

Другой способ проверки существования адаптивной установочной последовательности и построения установочного тестового примера, если такая последовательность существует, предложен в [29], однако алгоритм не гарантирует построения кратчайшей установочной последовательности.

Таким образом, алгоритм построения кратчайшей установочной последовательности для временного полностью определенного наблюдаемого автомата содержит следующие шаги.

Алгоритм 2 проверки существования и построения кратчайшей адаптивной установочной последовательности для автомата с временными ограничениями

Вход: полностью определённый недетерминированный наблюдаемый автомат с временными ограничениями $\mathsf{S} = (S, I, O, \lambda_{\mathsf{S}})$

Выход: установочный тестовый пример $\mathsf{TTC}(I_A, O)$ для временного автомата S или сообщение "Для автомата S не существует адаптивной установочной последовательности"

1. Строится конечно-автоматная абстракция $\mathsf{A_S}$ автомата S; $L := 1$.
2. Если $L \leq n^3$ строится автомат $(\mathsf{A_S})^L_{home}$ и Шаг 3, иначе выдаётся сообщение "Для автомата S не существует адаптивной установочной последовательности", и алгоритм завершает работу.
3. Если $(\mathsf{A_S})^{L-1}_{home} = (\mathsf{A_S})^L_{home}$, то выдаётся сообщение "Для автомата S не существует адаптивной установочной последовательности", и алгоритм завершает работу; иначе, Шаг 4;
4. Если автомат $(\mathsf{A_S})^L_{home}$ не содержит полностью определенных подавтоматов, то Шаг 5, иначе, $L = L + 1$ и Шаг 2;
5. Строится тестовый пример $\mathsf{TTC}(I_A, O)$ высоты L.

Корректность алгоритма 2 непосредственно следует из утверждений 5 и 6.

Для конечно-автоматной абстракции $\mathsf{A_S}$ (рисунок 2) временного автомата S (рисунок 1), фрагмент установочного автомата Q^2_{home} для подавтомата Q с выделенными серым цветом входными символами представлен на рисунке 4.

Соответствующий установочный тестовый пример, представляющий адаптивную установочную последовательность, изображён на рисунке 5 и имеет высоту два, в то время как кратчайшая безусловная установочная последовательность состоит из трёх входных символов.

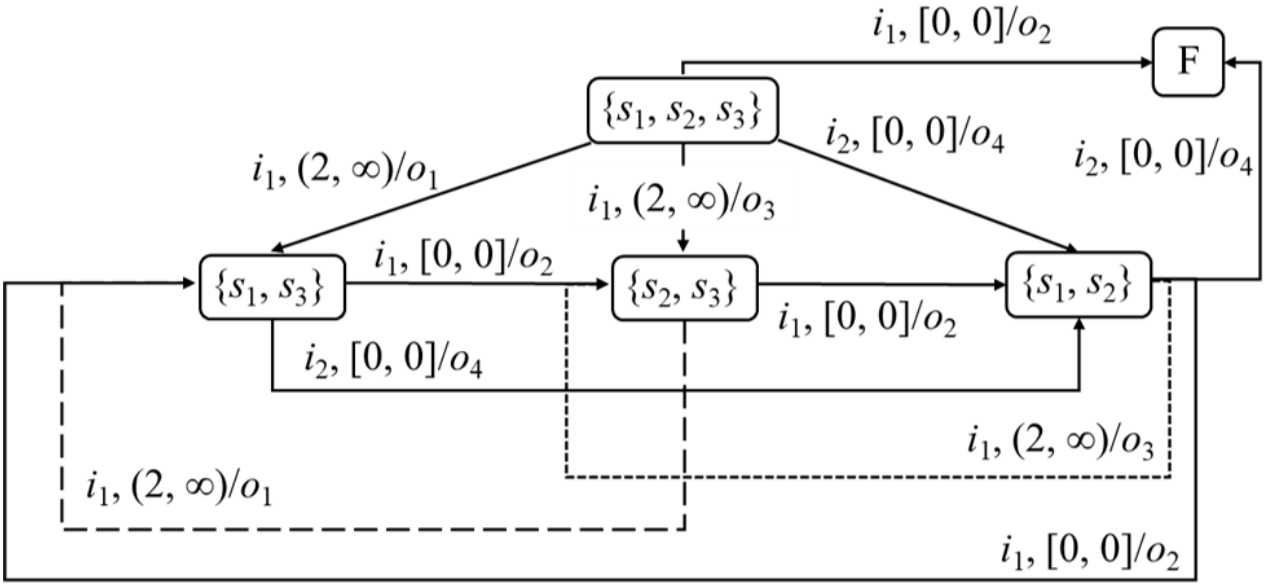

Fig. 4. A fragment of a Homing FSM for submachine Q of the FSM abstraction A_S in figure 2

Рис. 4. Фрагмент установочного автомата для подавтомата Q конечно-автоматной абстракции A_S на рисунке 2

3. Оптимизация конечно-автоматной абстракции автоматов с временными ограничениями

Согласно алгоритмам 1 и 2, сложность построения (адаптивной) установочной последовательности существенно зависит от числа входных символов автомата. Отметим, что при использовании конечно-автоматной абстракции число абстрактных входных символов существенно превышает количество входных символов исходного автомата с временными ограничениями, а именно, для временного автомата с мощностью входного алфавита $|I|$, число входных символов абстракции может достигнуть $(2(B_S+1) * |I|)$. Увеличение числа входных символов конечно-автоматной абстракции приводит к увеличению числа вершин в усечённом дереве преемников и увеличению входного алфавита установочного автомата. В настоящем разделе мы рассматриваем, каким образом можно сократить входной алфавит конечно-автоматной абстракции при проверке существования и построения адаптивной и безусловной установочных последовательностей для автомата с временными ограничениями.

Существуют различные возможности для оптимизации описанной в разделе 1.3 абстракции [21, 25], которые позволяют сократить входной алфавит при построении проверяющих тестов. В работе [26] показано, что при построении установочных последовательностей для временного автомата можно использовать одну из таких оптимизаций, а именно, рассматривать в абстракции только абстрактные входные символы, которые соответствуют целочисленным моментам времени. Однако, последнее возможно только при условии, что все временные интервалы в исходном временном автомате закрыты слева и далее мы предлагаем более общий алгоритм оптимизации, который заключается в построении *HS*-абстракции временного автомата.

Рассмотрим конечно-автоматную абстракцию (рисунок 2) временного автомата на рисунке 1. Непосредственной проверкой можно убедиться, что для некоторых пар различных абстрактных входных символов, строки таблицы переходов совпадают. Например, для входных символов $(i_1, [0, 0])$ и $(i_1, (0, 1))$, в любом состоянии автомата, множества пар (выходная реакция / следующее состояние) совпадают, и далее мы показываем, что для поиска установочных последовательностей достаточно рассмотреть только один из них. В работе [26] нами показано, что если в полностью определенном

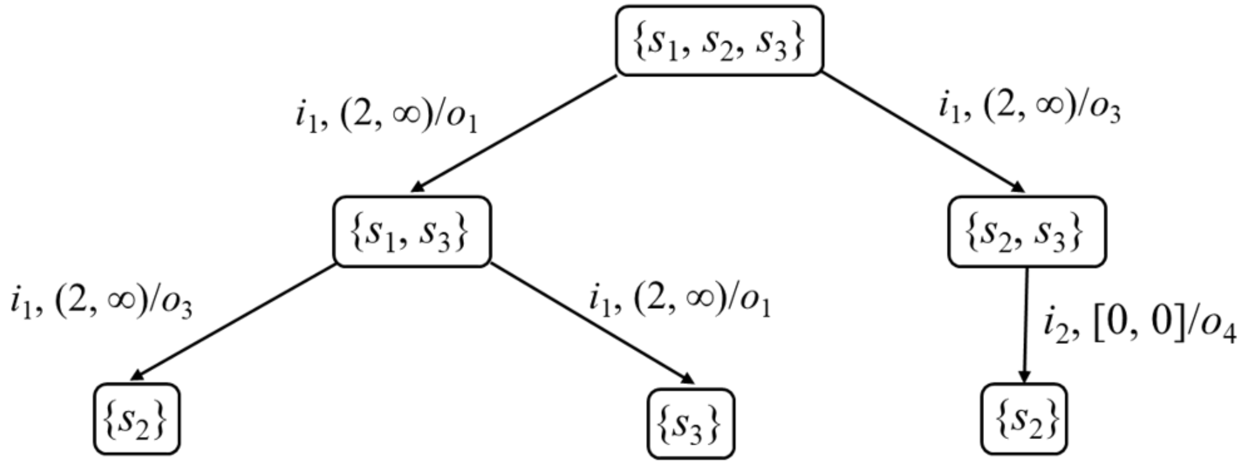

Fig. 5. A Homing test case for Timed FSM S in figure 1

Рис. 5. Установочный тестовый пример для временного автомата S на рисунке 1

автомате $\mathsf{P} = (P, I, O, h_\mathsf{P})$ существуют входные символы $i, i' \in I$, такие что в любом состоянии p, $(p, i, o, p') \in h_\mathsf{P}$, если и только если $(p, i', o, p') \in h_\mathsf{P}$, то справедливо следующее: автомат P обладает установочной последовательностью α, если и только такая последовательность существует для подавтомата, в котором отсутствует входной символ i'. Утверждение 5 иллюстрирует, что данное свойство сохраняется и для адаптивных установочных последовательностей, что позволяет ввести понятие *HS*-абстракции временного автомата, которая сохраняет свойства временного автомата относительно адаптивных и безусловных установочных последовательностей. Более того, в отличие от [26] мы также рассматриваем случай, когда множество пар (выходная реакция / следующее состояние) для любого состояния по одному входному символу включает в себя соответствующее множество для другого входного символа.

Утверждение 7. Пусть $\mathsf{P} = (P, I, O, h_\mathsf{P})$ – конечный полностью определенный автомат, где $i, i' \in I$, и для каждого состояния s справедливо, что если $(p, i, o, p') \in h_\mathsf{P}$, то $(p, i', o, p') \in h_\mathsf{P}$. Конечный автомат P обладает кратчайшей безусловной (адаптивной) установочной последовательностью α, если и только если безусловная (адаптивная) входная последовательность α', полученная из α заменой каждого входного символа i' на i, является кратчайшей установочной последовательностью для подавтомата $\mathsf{P}' = (P, I \setminus i', O, h_{\mathsf{P}'})$, где $h_{\mathsf{P}'}$ получен из h_P удалением переходов по входному символу i'.

Доказательство. Пусть для конечного автомата P существует кратчайшая установочная последовательность $\alpha = i_1 \dots i_l$. Следовательно, для каждой вход-выходной последовательности $\alpha/\gamma = i_1/o_1 \dots i_l/o_l$, α/γ-преемник множества состояний P является одноэлементным или не существует. Поскольку для входных символов i и i' множество пар $\left\{(o, p') : (p, i, o, p') \in h_\mathsf{P}\right\}$ содержится в множестве пар $\left\{(o, p') : (p, i', o, p') \in h_\mathsf{P}\right\}$ для любого состояния p, то для последовательности α', полученной из α заменой каждого входного символа i' на i, α/γ-преемник множества P содержит в себе α'/γ-преемник P, т.е. является одноэлементным или не существует. Таким образом, α' – кратчайшая установочная последовательность для автомата P', если и только если α – кратчайшая установочная последовательность для автомата P.

Следствие. Конечный автомат P обладает кратчайшей безусловной (адаптивной) установочной последовательностью α длины l, если и только если подавтомат $P' = (P, I \setminus i', O, h_{P'})$ обладает кратчайшей установочной последовательностью α' длины l.

Иными словами, если для пары входных символов i' и i для каждого состояния p имеет место условие: из $(p, i, o, p') \in h_{\mathsf{P}}$, следует что $(p, i', o, p') \in h_{\mathsf{P}}$, то при проверке существования и построении безусловной (адаптивной) установочной последовательности вместо конечно-автоматной абстракции достаточно рассмотреть ее полностью определенный подавтомат без входных символов i', обладающих описанным выше свойством, что приводит нас к следующем определению конечно-автоматной абстракции.

HS-абстракция $\mathsf{A}_{\mathsf{S}}^{\mathsf{HS}} = (S, I_A^{\mathsf{HS}}, O, h_{\mathsf{AS}}^{\mathsf{HS}})$ временного автомата S представляет собой полностью определенный подавтомат конечно-автоматной абстракции $\mathsf{A}_{\mathsf{S}} = (S, I_A, O, h_{\mathsf{AS}})$, получаемый итеративным удалением из A_{S} входных символов по следующему правилу. Входной символ i удаляется из конечно-автоматной абстракции, если существует входной символ $i' \neq i$, такой что для любого состояния s справедливо $\left\{(o, s') : (s, i', o, s') \in h_{\mathsf{AS}}\right\} \subseteq \left\{(o, s') : (s, i, o, s') \in h_{\mathsf{AS}}\right\}$. Входная последовательность α_{HFSM} конечно-автоматной абстракции $\mathsf{A}_{\mathsf{S}}^{\mathsf{HS}}$ может быть преобразована во временную входную последовательность исходного автомата аналогично такому преобразованию для конечно автоматной абстракции A_{S}.

Согласно сформулированным выше утверждениям, *HS*-абстракция временного автомата является конечным автоматом и «наследует» многие свойства исходного временного автомата, такие как полнота, наблюдаемость, приведенность по состояниям и связность. Кроме того, полностью определенный наблюдаемый временной автомат обладает адаптивной или безусловной установочной последовательностью, если и только если его *HS*-абстракция обладает такими установочными последовательностями.

Утверждение 8. Пусть $\mathsf{S} = (S, I, O, \lambda_{\mathsf{S}})$ — полностью определённый наблюдаемый автомат с временными ограничениями. Временная входная последовательность α является кратчайшей безусловной (адаптивной) установочной последовательностью для автомата S, если и только если соответствующая входная последовательность α_{HFSM} является кратчайшей безусловной (адаптивной) установочной последовательностью для *HS*-абстракции $\mathsf{A}_{\mathsf{S}}^{\mathsf{HS}}$.

Доказательство. Пусть α — кратчайшая установочная последовательность длины l для автомата с временными ограничениями S. Согласно утверждениям 2 и 5, α является кратчайшей временной безусловной (адаптивной) установочной последовательностью для временного автомата S, если и только если α_{FSM} является кратчайшей безусловной (адаптивной) установочной последовательностью для конечно-автоматной абстракции A_{S}. В силу утверждения 7, α_{FSM} является кратчайшей безусловной (адаптивной) установочной последовательностью для автомата A_{S}, если и только если соответствующая входная последовательность α_{HFSM} является кратчайшей безусловной (адаптивной) установочной последовательностью для *HS*-абстракции $\mathsf{A}_{\mathsf{S}}^{\mathsf{HS}}$.

Таким образом, входной алфавит конечно-автоматной абстракции может быть существенно сокращен при проверке существования или построении кратчайшей установочной последовательности по усечённому дереву преемников или установочному автомату.

Заключение

В настоящей работе была исследована задача синтеза установочных последовательностей для автоматов с временными ограничениями. Показано, как для проверки существования и синтеза адаптивных и безусловных временных установочных последовательностей можно адаптировать существующие алгоритмы для классических конечных автоматов. Оценки длин установочных последовательностей для детерминированных и недетерминированных временных автоматов совпадают с таковыми для классических автоматов. Полученные результаты основываются на построении по временному автомату соответствующей конечно-автоматной абстракции и в работе также предлагается подход к оптимизации известного метода синтеза таких абстракций.

Интересным продолжением данной работы является проверка существования и синтеза установочных последовательностей для автоматов с таймаутами, для которых переход автомата между состояниями возможен не только по входному воздействию, но и при достижении временной переменной определённого значения (таймаута). Такое расширение, с одной стороны, приводит к увеличению числа состояний в конечно-автоматной абстракции, а с другой – требует нового определения установочной последовательности.

References

[1] A. Gill, *Introduction to the Theory of Finite-State Machines*. McGraw-Hill, 1962.

[2] D. Lee and M. Yannakakis, “Testing finite state machines: state identification and verification”, *IEEE Transactions on Computers*, vol. 43, no. 3, pp. 306–320, 1994.

[3] T. N. Hibbard, “Least upper bounds on minimal terminal state experiments for two classes of sequential machines”, *Journal of the ACM*, vol. 8, no. 4, pp. 601–612, 1961.

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

[5] P. Starke, *Abstract Automata*. American Elsevier, 1972.

[6] G. Bochmann and A. Petrenko, “Protocol testing: review of methods and relevance for software testing”, in *In Proc. of International Symposium on Software Testing and Analysis*, 1994, pp. 109–124.

[7] G.-V. Jourdan, H. Ural, and H. Yenigun, “Reduced checking sequences using unreliable reset”, *Information Processing Letters*, vol. 115, no. 5, pp. 532–535, 2015.

[8] N. Kushik, J. López, A. Cavalli, and N. Yevtushenko, “Improving Protocol Passive Testing through ”Gedanken” Experiments with Finite State Machines”, in *IEEE International Conference on Software Quality, Reliability and Security*, 2016, pp. 315–322.

[9] F. Hennie, “Fault-detecting experiments for sequential circuits”, in *Proceedings of Fifth Annual Symposium on Circuit Theory and Logical Design*, 1965, pp. 95–110.

[10] T. S. Chow, “Testing software design modeled by finite-state machines”, *IEEE Transactions on Software Engineering*, vol. 4, no. 3, pp. 178–187, 1978.

[11] H.-E. Wang, K.-H. Tu, J.-H. R. Jiang, and N. Kushik, “Homing Sequence Derivation with Quantified Boolean Satisfiability”, in *Testing Software and Systems*, ser. LNCS, vol. 10533, 2017, pp. 230–242.

[12] H. Yenigun, N. Yevtushenko, N. Kushik, and J. López, “The effect of partiality and adaptivity on the complexity of FSM state identification problems”, *Trudy ISP RAN/Proceedings ISP RAS*, vol. 30, no. 1, pp. 7–24, 2018.

[13] N. Kushik and N. Yevtushenko, “On the Length of Homing Sequences for Nondeterministic Finite State Machines”, in *Implementation and Application of Automata*, ser. LNCS, vol. 7982, Springer, 2013, pp. 220–231.

[14] S. Sandberg, “Homing and Synchronizing Sequences”, in *Model-Based Testing of Reactive Systems*, ser. LNCS, vol. 3472, Springer, 2005, pp. 5–33.

[15] E. Bayse, A. R. Cavalli, M. Núñez, and F. Zaïdi, “A passive testing approach based on invariants: application to the WAP”, *Computer Networks*, vol. 48, no. 2, pp. 247–266, 2005.

[16] N. Kushik, K. El-Fakih, and N. Yevtushenko, “Preset and Adaptive Homing Experiments for Nondeterministic Finite State Machines”, in *Implementation and Application of Automata*, ser. LNCS, vol. 6807, Springer, 2011, pp. 215–224.

[17] N. Kushik, K. El-Fakih, and N. Yevtushenko, “Adaptive Homing and Distinguishing Experiments for Nondeterministic Finite State Machines”, in *Testing Software and Systems*, ser. LNCS, vol. 8254, Springer, 2013, pp. 33–48.

[18] N. Kushik and H. Yenigün, “Heuristics for Deriving Adaptive Homing and Distinguishing Sequences for Nondeterministic Finite State Machines”, in *Testing Software and Systems*, ser. LNCS, vol. 9447, Springer, 2015, pp. 243–248.

[19] H. Yenigün, N. Yevtushenko, and N. Kushik, “The complexity of checking the existence and derivation of adaptive synchronizing experiments for deterministic FSMs”, *Information Processing Letters*, vol. 127, pp. 49–53, 2017.

[20] M. Krichen and S. Tripakis, “Conformance testing for real-time systems”, *Formal Methods in System Design*, vol. 34, no. 3, pp. 238–304, 2009.

[21] K. El-Fakih, N. Yevtushenko, and H. Fouchal, “Testing timed finite state machines with guaranteed fault coverage”, in *Proc. of the 21st IFIP WG 6.1 Int. Conf. on Testing of Software and Communication Systems and 9th Int. FATES Workshop*, 2009, pp. 66–80.

[22] M. Merayo, M. Nunez, and I. Rodriguez, “Formal testing from timed finite state machines”, *Computer Networks: The International Journal of Computer and Telecommunications Networking*, vol. 52, no. 2, pp. 432–460, 2008.

[23] D. Bresolin, K. El-Fakih, T. Villa, and N. Yevtushenko, “Deterministic timed finite state machines: Equivalence checking and expressive power”, in *International conference GANDALF*, 2014, pp. 203–216.

[24] M. Gromov, K. El-Fakih, N. Shabaldina, and N. Yevtushenko, “Distinguishing non-deterministic timed finite state machines”, in *In Formal Techniques for Distributed Systems*, ser. LNCS, vol. 5522, Springer, 2009, pp. 137–151.

[25] K. Tvardovskii A.S.and El-Fakih, M. Gromov, and N. Yevtushenko, “Testing Timed Nondeterministic Finite State Machines with the Guaranteed Fault Coverage”, *Automatic Control and Computer Sciences*, vol. 51, no. 7, pp. 724–730, 2017.

[26] A. Tvardovskii and N. Yevtushenko, “Deriving homing sequences for Finite State Machines with timed guards”, *System Informatics*, vol. 17, pp. 1–10, 2020.

[27] J. Hartmanis and R. Stearns, *Algebraic structure theory of sequential machines*. Prentice-Hall, 1966.

[28] E. Vinarskii, A. Tvardovskii, L. Evtushenko, and N. Yevtushenko, “Deriving adaptive homing sequences for weakly initialized nondeterministic FSMs”, 2019, pp. 1–5.

[29] N. Yevtushenko, V. Kuliamin, and N. Kushik, “Evaluating the Complexity of Deriving Adaptive Homing, Synchronizing and Distinguishing Sequences for Nondeterministic FSMs”, in *Testing Software and Systems*, ser. LNCS, vol. 11812, Springer, 2019, pp. 86–103.

[30] E. Vinarskii and N. Yevtushenko, “Evaluating Length of a Shortest Adaptive Homing Sequence for Weakly Initialized FSMs”, 2020, pp. 1–5.

[31] N. Kushik and N. Yevtushenko, "Adaptive Homing is in P", *Electronic Proceedings in Theoretical Computer Science*, vol. 180, pp. 73–78, 2015.