Моделирование и анализ информационных систем. Т. 24, №4 (2017), с. 496–507
Modeling and Analysis of Information Systems. Vol. 24, No 4 (2017), pp. 496–507

DOI: 10.18255/1818-1015-2017-4-496-507

УДК 519.7

Синтез тестов с гарантированной полнотой для недетерминированных временных автоматов

Твардовский А. С.[1], **Эль-Факи К.**, **Громов М. Л.**[1], **Евтушенко Н. В.**[1]

получена 22 декабря 2016

Аннотация. В настоящее время при описании поведения дискретных систем достаточно часто необходимо принимать во внимание временные аспекты, и соответственно появляется необходимость в распространении автоматных методов синтеза тестов с гарантированной полнотой на временные автоматы. В данной статье мы предлагаем метод построения проверяющих тестов с гарантированной полнотой для полностью определенного, возможно, недетерминированного автомата с одной временной переменной. Такие временные автоматы используются при описании поведения программного обеспечения и цифровых устройств. Область неисправности содержит все полностью определенные автоматы с заданным числом состояний и известной верхней оценкой на интервалы, описывающие временные ограничения. Предлагаемый метод опирается на построение по заданному временному автомату соответствующей конечно автоматной абстракции (абстрактного автомата). По абстрактному автомату строится проверяющий тест, последовательности которого суть временные входные последовательности. Более короткие тесты можно построить, если ввести дополнительные ограничения на область неисправности, например, для случая, когда известна наименьшая продолжительность каждого временного интервала в тестируемой реализации, и её величина больше двух. Кроме того, тест можно сократить с сохранением его полноты в случае, когда все интервалы для временных ограничений закрыты справа (или все интервалы закрыты слева). Приводятся результаты проведенных компьютерных экспериментов по сравнению длин тестов, построенных по временному автомату различными методами.

Ключевые слова: временные автоматы, недетерминированные автоматы, синтез тестов, область неисправности

Для цитирования: Твардовский А. С., Эль-Факи К., Громов М. Л., Евтушенко Н. В., "Синтез тестов с гарантированной полнотой для недетерминированных временных автоматов", *Моделирование и анализ информационных систем*, **24**:4 (2017), 496–507.

Об авторах:
Твардовский Александр Сергеевич, orcid.org/0000-0001-7705-7214, магистрант,
Национальный исследовательский Томский государственный университет,
пр. Ленина, 36, г. Томск, 634050 Россия, e-mail: tvardal@mail.ru

Калед Эль-Факи, д-р компьютерных наук, доцент,
Американский университет Шарджа, Университетский город, г. Шарджа, 26666 Объединенные Арабские Эмираты,
e-mail: kelfakih@aus.edu

Громов Максим Леонидович, orcid.org/0000-0002-2990-8245, канд. физ.-мат. наук, доцент,
Национальный исследовательский Томский государственный университет, пр. Ленина, 36, г. Томск, 634050 Россия,
e-mail: maxim.leo.gromov@gmail.ru

Евтушенко Нина Владимировна, orcid.org/0000-0002-4006-1161, д-р техн. наук, профессор,
Национальный исследовательский Томский государственный университет, пр. Ленина, 36, г. Томск, 634050 Россия,
e-mail: nyevtush@gmail.com

Благодарности:
[1]Работа выполнена при поддержке гранта РНФ No. 16-49-03012.

Введение

Автоматные методы предполагают построение тестовых последовательностей по заданному автомату-спецификации, для того чтобы определить, соответствует ли тестируемая реализация, поведение которой также описано автоматом, спецификации. При этом достаточно часто реализация рассматривается как «чёрный ящик». Для проверки соответствия на тестируемую реализацию подаются входные последовательности, и генерируемые при этом выходные последовательности сравниваются с ожидаемыми выходными реакциями (согласно спецификации). Если наблюдаемые реакции не соответствуют спецификации, то в тестируемой реализации имеется ошибка, т. е. реализация не является *конформной* спецификации. Для конечных автоматов существуют методы построения полных проверяющих тестов относительно определенных моделей неисправности [1–3] без явного перечисления неконформных автоматов. Достаточно часто рассматривается случай, когда поведение автомата-спецификации является недетерминированным, однако поведение тестируемой системы описывается детерминированным автоматом, и автомат-реализация считается конформным спецификации, если его поведение содержится в поведении спецификации. Другими словами, в этом случае предполагается, что недетерминизм в спецификации является следствием опциональности в неформальных описаниях, и поведение конформной реализации «не выходит за рамки», предписанные спецификацией. При условии, что поведение исследуемой системы описано временным автоматом (см., например, [4–8]), появляется необходимость в распространении автоматных методов синтеза тестов с гарантированной полнотой на временные автоматы.

В настоящей статье мы рассматриваем полностью определенные, возможно, недетерминированные автоматы с входными временными ограничениями и выходными задержками на переходах [5, 8] и используем конечно автоматный метод построения проверяющих тестов относительно редукции в предположении, что известны максимальное число состояний проверяемого автомата и максимальная конечная граница для входных временных интервалов. Построенный тест можно существенно сократить, если все входные интервалы автомата-спецификации и автомата-реализации закрыты слева (или все интервалы закрыты справа). Дальнейшее сокращение проверяющего теста с использованием подходящей конечно автоматной абстракции можно получить для случая, когда известно, что минимальная длина временных интервалов в автомате-спецификации и проверяемом автомате больше двух.

Структура статьи следующая. Раздел 1 содержит основные определения и обозначения. Алгоритм построения полного проверяющего теста относительно редукции по конечно автоматной абстракции представлен в разделе 2. Раздел 3 содержит экспериментальные результаты.

1. Основные определения и обозначения

В данном разделе мы вводим основные определения и обозначения, взятые преимущественно из [3, 6]. Под *конечным автоматом* S понимается пятёрка $(S, I, O, \lambda_{\mathsf{S}}, s_0)$, где S, I, и O – конечные непустые множества состояний, входных и выходных символов соответственно, s_0 – начальное состояние, $\lambda_{\mathsf{S}} \subseteq S \times I \times O \times S$ – отношение

переходов. Временной, возможно, недетерминированный и частичный автомат (далее – временной автомат) есть конечный автомат с временной переменной и временными ограничениями на переходах. Таким образом, под *временным автоматом* S понимается пятёрка $(S, I, O, \lambda_{\mathsf{S}}, s_0)$, где S, I, и O – конечные непустые множества состояний, входных и выходных символов соответственно, s_0 – начальное состояние, $\lambda_{\mathsf{S}} \subseteq S \times I \times O \times S \times \Pi \times Z$ – отношение переходов. Для λ_{S} через Π обозначено множество входных временных интервалов и через Z – множество выходных задержек. Временной входной интервал g_i описывает промежуток времени, когда переход может быть совершён и задаётся в виде $\lceil min, max \rceil$, где $\lceil \in \{(, [\}, \rceil \in \{),]\}$. В этом обозначении min и max суть неотрицательные целые числа, $min \leq max$, кроме того, max может быть равно ∞. В случае, когда $min = max$, допускается единственный интервал $[min;\ min]$. Выходная задержка представляет собой целое неотрицательное число, которое описывает число тактов времени, затраченное на обработку входного символа, т.е. время между подачей входного и получением выходного символа. Пусть $(s, i, o, s', g_i, d) \in S \times I \times O \times S \times \Pi \times Z$, и на автомат S, находящийся в состоянии s, поступает входной символ i в момент времени $t \in g_i$, измеряемый с момента перехода автомата в состояние s. Тогда временная переменная устанавливается в 0, автомат S выдаёт выходной символ o в момент времени d, и автомат переходит в состояние s' вновь с обнулением временной переменной.

Для временного автомата $\mathsf{S} = (S, I, O, \lambda_{\mathsf{S}}, s_0)$ пара (i, t), где $i \in I$ и t неотрицательное действительное число, называется *временным входным символом*, который означает, что входной символ i подается в момент времени t после выдачи автоматом последнего выходного символа. Аналогично определяется *временной выходной символ* для целого числа t. Последовательность временных входных (выходных) символов называется *временной входной* (*выходной*) *последовательностью*. Временная выходная последовательность, соответствующая временной входной последовательности α, поступившей на автомат в состоянии s, называется (*выходной*) *реакцией* автомата в состоянии s на последовательность α. Последовательность $(i_1,\ t_1)/(o_1,\ d_1) \ldots (i_m,\ t_m)/(o_m,\ d_m)$ пар временных входных и выходных символов есть временная *входо-выходная* последовательность автомата S в состоянии s, если во временном автомате существует последовательность переходов $(s_j, i_j, o_j, s_{j+1}, g_j, d_j)$, такая что $s_1 = s$ и для любого $j = 1, \ldots, m$, справедливо, что $t_j \in g_j$.

На рисунке 1 представлен автомат S с временными ограничениями, начальным состоянием которого является состояние 1. Под действием входного символа $(i,\ 0)$ автомат остаётся в состоянии 1 и выдает выходную реакцию o либо через один такт времени (после подачи входного символа), либо через 2 такта времени. Если входной символ i подать при значении временной переменной 1, то автомат выполнит переход в состояние 3 и выдаст выходную реакцию o через один такт времени, либо перейдёт в состояние 2 и выдаст выходную реакцию o через три такта.

Автомат с временными ограничениями называется *полностью определённым*, если поведение автомата определено в каждом состоянии для каждой временной входной последовательности. Если для любых двух кортежей $(s,\ i,\ o_1,\ s_1,\ g_1,\ d_1)$, $(s,\ i,\ o_2,\ s_2,\ g_2,\ d_2) \in \lambda_{\mathsf{S}}$ справедливо соотношение $g_1 \bigcap g_2 = \varnothing$, то временной автомат называется *детерминированным*, иначе – *недетерминированным*. Недетерминированный автомат называется *наблюдаемым*, если для каждой тройки «состояние, временной входной символ, временной выходной символ» следующее состояние

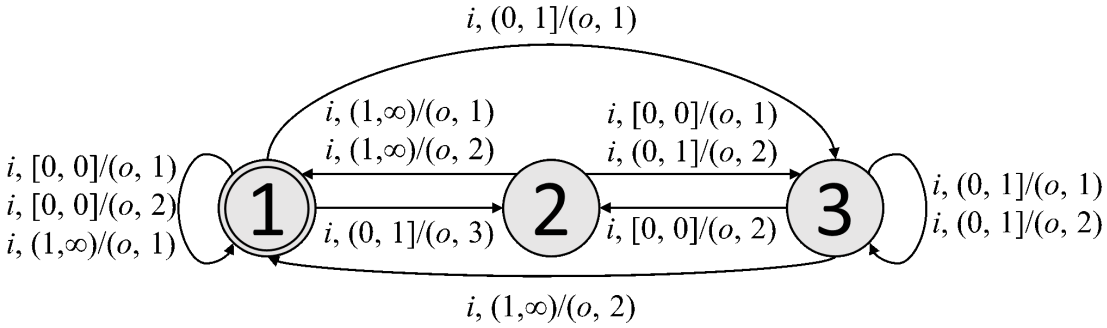

Рис. 1. Временной автомат S
Fig. 1. Timed Finite State Machine S

определяется единственным образом. Для любого недетерминированного автомата существует эквивалентный наблюдаемый автомат. В нашей работе все временные автоматы полагаются полностью определенными, автомат-реализация является детерминированным временным автоматом, в то время как автомат-спецификация может быть недетерминированным, но наблюдаемым.

Состояние p временного автомата P является *редукцией* состояния s временного автомата S ($p \leq s$), если множество входо-выходных последовательностей автомата P в состоянии p содержится в множестве входо-выходных последовательностей автомата S в состоянии s. Автомат P называется редукцией автомата S, если отношение редукции выполняется для начальных состояний этих автоматов.

Проверяющий тест для конечного автомата представляет собой множество входных последовательностей, позволяющее определить, конформна (соответствует) ли спецификации тестируемая реализация. В настоящей работе реализация *конформна* спецификации, если реализация есть редукция спецификации, т.е. реализация конформна спецификации, если и только если для каждой временной входной последовательности выходная реакция реализации содержится в множестве выходных реакций спецификации. В соответствии с [8] полный проверяющий тест для временного автомата относительно редукции можно построить по его конечно автоматной абстракции.

Пусть S – временной автомат и B – целое число, не меньшее чем наибольшая конечная граница B_{S} для временных входных интервалов, в то время как D_{S} определяет наибольшую выходную задержку. Построим *конечно автоматную абстракцию* временного автомата, которая является полностью определённым конечным автоматом $\mathsf{A}_{\mathsf{S}}(B) = (S, I_A, O_A, \lambda_{\mathsf{AS}}, s_0)$, где $I_A = \{(i,0),(i,(0,1)),\ldots,(i,B),(i,(B,\infty)) : i \in I\}$, а $O_A = \{(o,0),(o,1),\ldots,(o,D_{\mathsf{S}}) : o \in O\}$. Для состояния s автомата $\mathsf{A}_{\mathsf{S}}(B)$ и входного символа $(i,t_j), t_j = 0,\ldots,B$, множество λ_{AS} содержит переход $(s,(i,t_j),(o,d),s')$, если и только если существует переход $(s,i,o,s',g_i,d) \in \lambda_{\mathsf{S}}$, такой что $t_j \in g_i$. Для входного символа $(i,g), g = (0,1),\ldots,(B-1,B),(B,\infty)$, множество λ_{AS} содержит переход $(s,(i,g),(o,d),s')$, если и только если существует переход $(s,i,o,s',g_i,d) \in \lambda_{\mathsf{S}}$, такой что $g \subseteq g_i$. Если временной автомат S недетерминированный, то конечный автомат $\mathsf{A}_{\mathsf{S}}(B)$ также будет недетерминированным. Для временного автомата на рисунке 1 конечно автоматная абстракция изображена на рисунке 2.

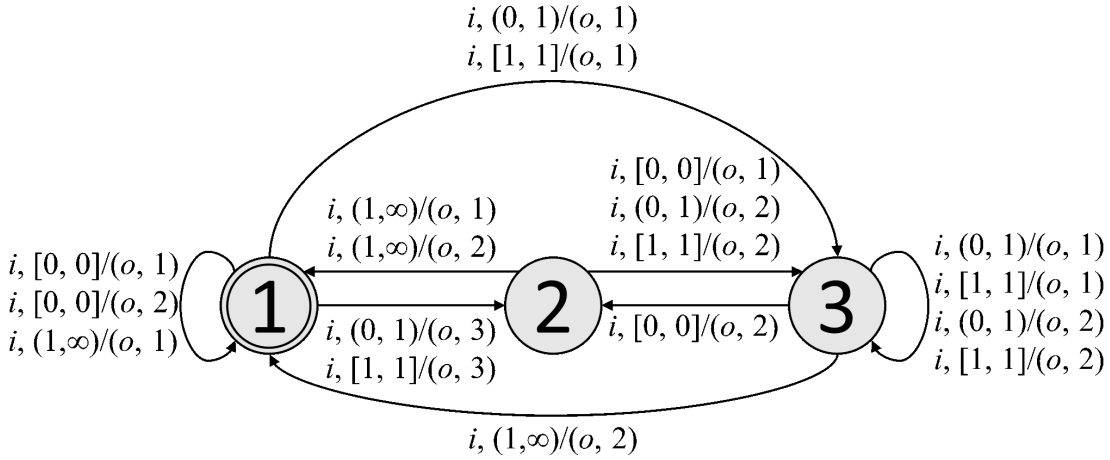

Рис. 2. Конечный автомат $\mathsf{A_S}(1)$
Fig. 2. Finite State Machine $\mathsf{A_S}(1)$

Аналогично [9] можно сформулировать и доказать следующее утверждение.

Утверждение 1. *Для двух полностью определённых наблюдаемых, возможно, недетерминированных временных автоматов* P *и* S *с наибольшей входной границей* B*, автомат* P *есть редукция автомата* S*, если и только если отношение редукции выполняется для их конечно автоматных абстракций, т.е. если и только если* $\mathsf{A_P}(B)$ *есть редукция* $\mathsf{A_S}(B)$*.*

Действительно, в начальном состоянии конечно автоматной абстракции $\mathsf{A_S}(B)$ возможен переход по временному входному символу (i, t), если и только если в начальном состоянии исходного автомата для входного символа i существует интервал g, где $t \in g$. Далее утверждение доказывается по индукции.

Мы рассматриваем *модель неисправности* $< \mathsf{S}, \leq, \mathfrak{J}_m(B) >$ [9], где S – временной автомат-спецификация, который является полностью определённым и наблюдаемым, $\leq$ – отношение редукции и $\mathfrak{J}_m(B)$ – область неисправности, содержащая каждый полностью определённый детерминированный временной автомат с таким же входным алфавитом, как и у спецификации, не более чем с m состояниями и наибольшей конечной границей B для входных временных интервалов.

Проверяющий тест есть конечное множество конечных временных входных последовательностей спецификации. Тест является *полным* относительно $< \mathsf{S}, \leq, \mathfrak{J}_m(B) >$, если для каждого временного автомата $\mathsf{P} \in \mathfrak{J}_m(B)$, такого что P есть редукция S, выходная реакция на каждую последовательность теста содержится в множестве выходных реакций S на эту последовательность, в то время как для каждого временного автомата $\mathsf{P} \in \mathfrak{J}_m(B)$, такого что P не является редукцией S, в тесте есть последовательность, выходная реакция на которую не принадлежит множеству выходных реакций S на эту последовательность. Согласно утверждению 1, полный проверяющий тест относительно $< \mathsf{S}, \leq, \mathfrak{J}_m(B) >$ может быть построен с использованием классических конечно автоматных методов относительно модели неисправности $< \mathsf{A_S}(B), \leq, \mathfrak{J}_m(B) >$. Здесь $\mathsf{A_S}(B)$ – конечно автоматная абстракция

временного автомата S и $\mathfrak{J}_m(B)$ – множество всех полностью определённых детерминированных временных автоматов не более чем с m состояниями и таким же входным алфавитом, как у $\mathsf{A}_\mathsf{S}(B)$. Мы далее вводим ряд понятий, которые используются в следующем разделе при построении полного проверяющего теста.

Состояния s_1 и s_2 автомата $\mathsf{A}_\mathsf{S}(B)$ *разделимы*, если существует входная последовательность α, такая что множества выходных реакций на α в состояниях s_1 и s_2 не пересекаются; последовательность α называется *разделяющей* для состояний s_1 и s_2.

Входная последовательность α называется *адаптивной*, если следующий входной символ зависит от реакции автомата на предыдущие входные символы. Адаптивную входную последовательность удобно представлять в виде специального автомата, который во многих работах называется *тестовым примером* (test case) [10]. В тестовом примере в каждом состоянии определен переход либо только по одному входному символу со всеми выходными символами, либо не определён ни один переход (это состояние назовём *терминальным*). Диаграмма переходов тестового примера есть ациклический граф (рис. 3).Тестовый пример представляет *адаптивную различающую последовательность* для состояний s_1 и s_2 автомата $\mathsf{A}_\mathsf{S}(B)$, если каждая входо-выходная последовательность из начального в терминальное состояние тестового примера возможна не более чем в одном из состояний s_1 или s_2. В первом случае распознаётся состояние s_1, во втором – s_2. Состояния s_1 и s_2 автомата $\mathsf{A}_\mathsf{S}(B)$ *адаптивно различимы*, если существует тестовый пример, который представляет адаптивную различающую последовательность для состояний s_1 и s_2. Если адаптивная последовательность различает каждую пару состояний автомата S, то такая последовательность есть адаптивная различающая последовательность для автомата S. Тестовый пример на рис. 3 представляет адаптивную различающую последовательность для автомата $\mathsf{A}_\mathsf{S}(1)$ на рис. 2. В вершинах графа переходов тестового примера содержатся подмножества состояний, достижимые по соответствующей входо-выходной последовательности. Заметим, что для этого автомата не существует разделяющей последовательности, существует только адаптивная различающая последовательность. В терминальных состояниях также указано состояние (`in_st`), в котором находился автомат перед поступлением адаптивной различающей последовательности.

Состояние s *детерминировано* (*∂-*) *достижимо* из начального состояния, если существует входная последовательность α такая, что при любой выходной реакции β на последовательность α автомат $\mathsf{A}_\mathsf{S}(B)$ переходит из начального состояния в состояние s. В этом случае α называется (*∂-*) передаточной последовательностью для состояния s.

Тестовый пример представляет *адаптивную передаточную последовательность* из начального состояния автомата $\mathsf{A}_\mathsf{S}(B)$ в состояние s, если каждая входо-выходная последовательность из начального в терминальное состояние тестового примера заканчивается в состоянии s [10]. В этом случае состояние s *адаптивно достижимо* из начального состояния. Тестовые примеры на рис. 4 представляют адаптивные передаточные последовательности из состояния 1 в состояния 2 и 3 для автомата на рис. 2.

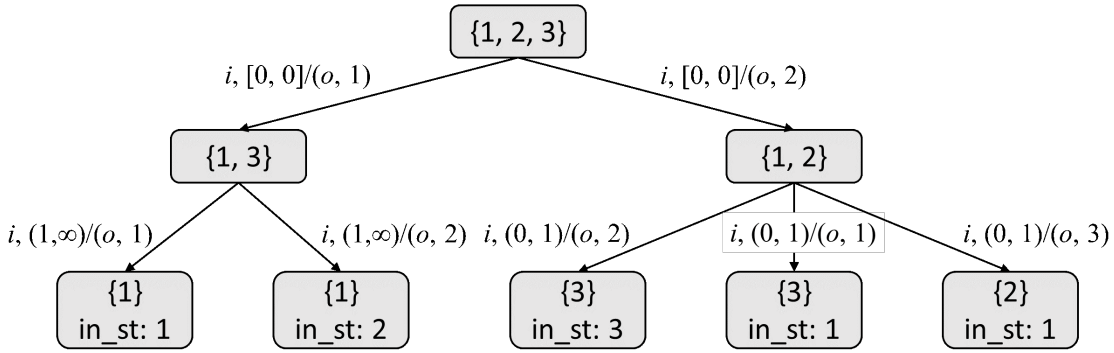

Рис. 3. Тестовый пример, представляющий адаптивную различающую последовательность для автомата $\mathsf{A_S}(1)$

Fig. 3. A distinguishing test case for $\mathsf{A_S}(1)$

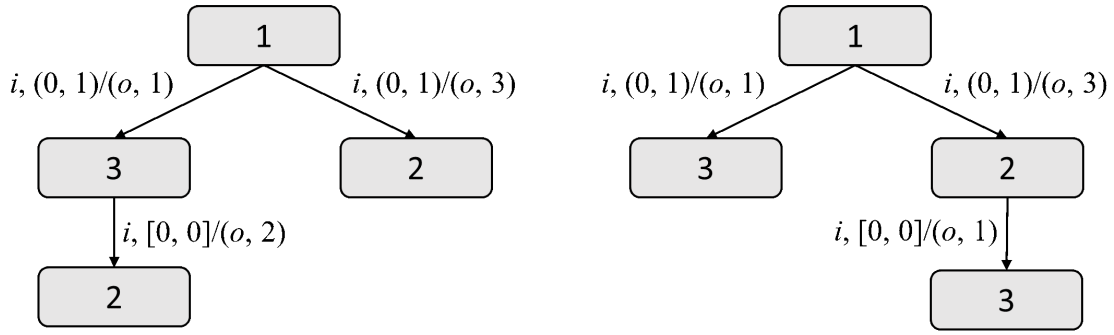

Рис. 4. Тестовые примеры, представляющие адаптивные передаточные последовательности для автомата $\mathsf{A_S}(1)$

Fig. 4. Transfer test cases for $\mathsf{A_S}(1)$

2. Синтез полного проверяющего теста

Общий алгоритм синтеза полного проверяющего теста относительно редукции для недетерминированного конечного автомата описан в [7]. Метод основан на использовании ∂-передаточных и различающих последовательностей, которые, как известно, не всегда существуют. В настоящей работе мы предлагаем использовать адаптивные передаточные и различающие последовательности, поскольку, во-первых, такие последовательности существуют значительно чаще, а во-вторых, в таком случае, длина полного проверяющего теста может быть значительно меньше [11].

Если все состояния конечно автоматной абстракции $\mathsf{A_S}(B)$ автомата S ∂-достижимы (адаптивно достижимы), автомат $\mathsf{A_S}(B)$ обладает разделяющей последовательностью (адаптивной различающей последовательностью), и число состояний реализации не превышает числа состояний спецификации, то алгоритм построения полного проверяющего теста относительно модели неисправности $< \mathsf{A_S}(B), \leq, \mathfrak{J}_m(B) >$, где m — число состояний автомата $\mathsf{A_S}(B)$, включает следующие шаги.

1. Строится *множество д-достижимости* (адаптивной достижимости) для автомата $\mathsf{A}_{\mathsf{S}}(B)$, которое содержит $д$-передаточную (адаптивную передаточную) последовательность для каждого состояния автомата $\mathsf{A}_{\mathsf{S}}(B)$.

2. Каждая последовательность из множества $д$-достижимости конкатенируется с разделяющей (адаптивной различающей) последовательностью и каждым входным символом; после каждого входного символа добавляется разделяющая (адаптивная различающая) последовательность.

Таким образом, в соответствии с утверждением 1, аналогичный подход может быть использован для построения полного проверяющего теста для спецификации, описанной временным автоматом.

Для того чтобы построить полный проверяющий тест для временного недетерминированного автомата, мы используем его конечно автоматную абстракцию. Полный тест строится для конечно автоматной абстракции; построенные входные последовательности (входо-выходные последовательности при построении адаптивного теста) являются временными входными (временными входо-выходными) последовательностями для исходного временного автомата. Построенный тест обладает полиномиальной длиной относительно числа состояний временного автомата-спецификации, когда длины разделяющей последовательности и $д$-передаточных последовательностей (адаптивных различающей и передаточных последовательностей) полиномиальны относительно числа состояний временного автомата-спецификации. Известно, что построенный тест обнаруживает все неконформные реализации с не более чем m состояниями и наибольшей конечной границей B для входных временных интервалов, где m есть число состояний временного автомата-спецификации.

Утверждение 2. *Если автомат $\mathsf{A}_{\mathsf{S}}(B)$ обладает разделяющей (адаптивной различающей) последовательностью и каждое состояние $\mathsf{A}_{\mathsf{S}}(B)$ д-достижимо (адаптивно достижимо) из начального состояния, то выше описанный алгоритм возвращает полный проверяющий тест относительно $< \mathsf{S}, \leq, \mathfrak{J}_m(B) >$.*

Действительно, по построению, автомат $\mathsf{A}_{\mathsf{S}}(B)$ имеет m состояний. Если каждая последовательность множества достижимости конкатенируется с разделяющей (адаптивной различающей) последовательностью, и проверяемый автомат реагирует на эту часть проверяющего теста согласно спецификации, то проверяемый автомат имеет ровно m состояний. Кроме того, можно установить взаимно однозначное соответствие между состояниями спецификации и тестируемого автомата по реакции в каждом состоянии на разделяющую (адаптивную различающую) последовательность. На следующем шаге с использованием этих реакций и по реакции тестируемого автомата на каждый входной символ и последующую разделяющую (адаптивную различающую) последовательность устанавливается взаимно однозначное соответствие между переходами автомата-спецификации и тестируемого автомата. После этого утверждение следует из утверждения 1.

Если спецификация не имеет разделяющей (адаптивной различающей) последовательности или не каждое состояние является детерминированно достижимым (адаптивно достижимым) из начального состояния, то проверяющий тест будет значительно длиннее [3]. В работе [12] показано, что можно сократить спецификацию,

чтобы получить тест разумной длины. В нашем примере на рисунке 1 предположим, что исходная спецификация имеет дополнительный переход $(1, i, o, 3, [0, 0], 1)$ из состояния 1 в состояние 3. Для такой спецификации не существует разделяющей (адаптивной различающей) последовательности, поскольку для любого входного символа существуют два состояния, из которых есть переход в одно и то же состояние с одним и тем же выходным символом. Однако после удаления этого перехода автомат-спецификация обладает необходимыми свойствами. Известно, что необходимое сокращение спецификации не всегда возможно, в частности, для детерминированного автомата-спецификации, и вопросы существования и оптимизации такого сокращения требуют дальнейших исследований.

Аналогично методу синтеза тестов по детерминированному автомату-спецификации [5], длину теста можно уменьшить с сохранением его полноты, если все временные входные интервалы спецификации закрыты слева (или все временные входные интервалы закрыты справа), или продолжительность входных временных интервалов в реализации и спецификации больше двух. В первом случае временные входные символы теста достаточно подавать только в целочисленные моменты времени, т.е. при построении теста из конечно автоматной абстракции спецификации удаляются все входные символы вида (i, g), где g – интервал ненулевой длины. Во втором случае временные входные символы достаточно подавать таким образом, чтобы на каждый входной временной интервал приходился как минимум один временной входной символ. В таком случае, входной алфавит конечно автоматной абстракции имеет вид $I_A = \{(i, 0), (i, (0, 1)), (i, w), (i, (w, w + 1)), (i, 2w), (i, (2w, 2w + 1)), \ldots, (i, n), (i, (n, \infty))) : i \in I, n < B\}$, где $w > 2$ – наименьшая длина временного интервала в спецификации, т.е. входные символы подаются в целочисленные моменты времени, кратные w и один раз в каждом интервале вида $(kw, kw + 1)$.

3. Экспериментальные результаты

В настоящем разделе мы кратко описываем полученные нами экспериментальные результаты. В частности, мы рассматриваем

- проверяющие тесты TS, полученные с использованием выше описанного алгоритма, т.е. тесты без использования какого-либо метода оптимизации;

- тесты $TS1$, построенные для спецификаций, в которых все временные интервалы закрыты слева (или все временные интервалы закрыты справа), и таким образом, достаточно рассматривать только целочисленные моменты времени при подаче входных символов;

- тесты $TS2$, когда все временные интервалы в спецификации и реализации имеют длительность больше, чем два, и для построения тестовых последовательностей используются только моменты времени, кратные минимальной длине временных интервалов и один из моментов времени из интервала $(kw, kw + 1)$;

- тесты $TS3$, когда оба выше упомянутых сокращения использованы в процессе синтеза теста.

Эксперименты проводились над временными автоматами с 20 состояниями и с не более чем десятью временными интервалами для каждой пары «состояние, входной

символ», для каждого автомата были построены тесты TS, $TS1$, $TS2$, и $TS3$ с использованием разделяющей и ∂-передаточных последовательностей. Приведённые экспериментальные данные иллюстрируют существенное различие между длинами тестов TS и $TS3$, которое практически не зависит от размеров автомата и числа входных временных интервалов. На рис. 5 представлено усредненное процентное соотношение (по 1000 экспериментов).

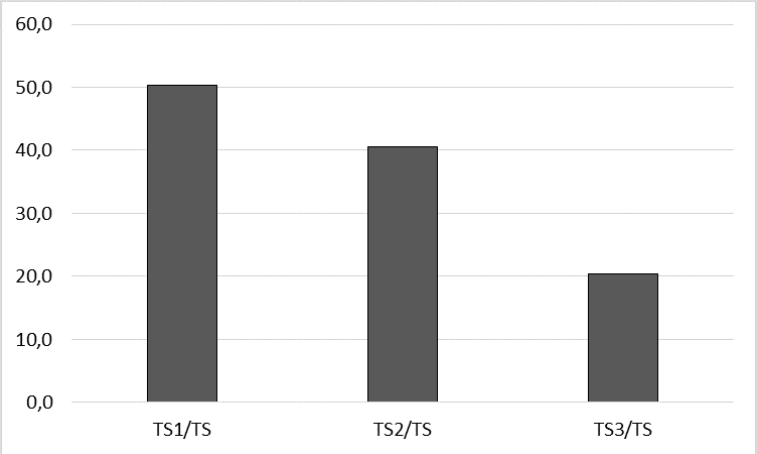

Рис. 5. Зависимость между длинами тестов, построенных различными методами
Fig. 5. Dependence of lengths of tests derived by different methods

При построении тестов $TS1$, $TS2$, $TS3$ нет гарантии, что проверяющий тест будет полным относительно модели неисправности $< \mathsf{S}, \leq, \mathfrak{J}_m(B) >$. Тем не менее, для большого количества случайно сгенерированных реализаций из множества $\mathfrak{J}_m(B)$, для которых $|I| = |O| = 4, |S| = 20$ и каждая из которых отличается от автомата-спецификации только на одном переходе, тест $TS3$ обнаружил каждую неконформную реализацию.

Следует также отметить, что адаптивные различающие и передаточные последовательности всегда существуют при наличии разделяющей и ∂-передаточных последовательностей. Более того, длины адаптивных последовательностей обычно существенно короче, и поэтому полученные экспериментальные результаты справедливы для временных автоматов при использовании адаптивных тестов.

4. Заключение

В настоящей статье предложен подход к синтезу полных проверяющих тестов для недетерминированных временных автоматов относительно редукции. Предложенный подход можно использовать для построения тестов для временных автоматов, когда автомат-спецификация обладает разделяющей (адаптивной различающей) последовательностью и любое состояние ∂-достижимо (адаптивно достижимо) из начального состояния. Кроме того, построенные тесты можно оптимизировать, и несмотря на то, что укороченные тесты теоретически не являются полными, все случайно сгенерированные временные автоматы, неконформные спецификации и

отличающиеся от нее только на одном переходе такими тестами были обнаружены. В дальнейшем для анализа полноты тестов, построенных оптимизированными методами, мы предполагаем исследовать специальные мутации спецификации, а также оценить процент необнаружимых неконформных реализаций для автоматов-спецификаций небольшого размера посредством генерации всех возможных реализаций. Заметим, что если автомат-спецификация не имеет разделяющей (адаптивной различающей) последовательности или не каждое состояние является ∂-(адаптивно) достижимым, то проверяющий тест будет значительно длиннее. Одной из возможностей сокращения проверяющих тестов является сокращение автомата-спецификации, но условия существования и оптимизация такого сокращения требуют дальнейших исследований.

Список литературы / References

[1] Chow T. S., "Test design modeled by finite-state machines", *IEEE Transactions on Software Engineering*, **4**:3 (1978), 178–187.

[2] Dorofeeva R., El-Fakih K., Maag S., Cavalli A., Yevtushenko N., "FSM-based conformance testing methods: a survey annotated with experimental evaluation", *Information and Software Technology*, **52** (2010), 1286–1297.

[3] Petrenko A., Yevtushenko N., "Conformance Tests as checking experiments for partial nondeterministic FSM", *Proc. of the 5th International Workshop on Formal Approaches to Testing of Software (FATES 2005)*, LNCS, **3997**, 2005, 118–133.

[4] Springintveld J., Vaandrager F., D'Argenio P., "Testing timed automata", *Theoretical Computer Science*, **254**:1–2 (2001), 225–257.

[5] El-Fakih K., Yevtushenko N., Fouchal H., "Testing timed finite state machines with guaranteed fault coverage", *Proceedings of the 21st IFIP WG 6.1 International Conference on Testing of Software and Communication Systems and 9th International FATES Workshop*, 2009, 66–80.

[6] Krichen M., Tripakis S., "Conformance testing for real-time systems", *Formal Methods in System Design*, **34**:3 (2009), 238–304.

[7] Merayo M.G., Nunez M., Rodriguez I., "Formal testing from timed finite state machines", *Computer Networks: The International Journal of Computer and Telecommunications Networking*, **52**:2 (2008), 432–460.

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

[9] Petrenko A., Yevtushenko N., Bochmann G.v., "Fault Models for Testing in Context", *FORTE*, 1996, 163–178.

[10] Yevtushenko N., El-Fakih K., Ermakov A., "On-the-fly construction of adaptive checking sequences for testing deterministic implementations of nondeterministic specifications", *LNCS*, **9976** (2016), 139–152.

[11] Petrenko A., Yevtushenko N., "Adaptive Testing of Deterministic Implementations Specified by Nondeterministic FSMs", *Proceedings of ICTSS*, 2011, 162–178.

[12] Tvardovskii A., Yevtushenko N., "Synthesis complete test suite for nondeterministic finite state machines with time guards", *Russian Physics Journal*, **58**:11/2 (2015), 107–110.

Tvardovskii A. S., El-Fakih K., Gromov M. L., Yevtushenko N. V., "Testing Timed Nondeterministic Finite State Machines with the Guaranteed Fault Coverage", *Modeling and Analysis of Information Systems*, **24**:4 (2017), 496–507.

DOI: 10.18255/1818-1015-2017-4-496-507

Abstract. Nowadays, the behaviour of many systems can be properly described by taking into account time constraints, and this motivates the adaptation of existing Finite State Machine (FSM)-based test derivation methods to timed models. In this paper, we propose a method for deriving conformance tests with the guaranteed fault coverage for a complete possibly nondeterministic FSM with a single clock; such Timed FSMs (TFSMs) are widely used when describing the behaviour of software and digital devices. The fault domain contains every complete TFSM with the known upper bounds on the number of states and finite boundary of input time guards. The proposed method is carried out by using an appropriate FSM abstraction of the given TFSM; the test is derived against an FSM abstraction and contains timed input sequences. Shorter test suites can be derived for a restricted fault domain, for instance, for the case when the smallest duration of an input time guard is larger than two. Moreover, the obtained test suites can be reduced, while preserving the completeness, when all input time guards of the specification and an implementation are right closed (or all intervals are left closed). Experiments are conducted to study the length of test suites constructed by different methods.

Keywords: timed finite state machines, nondeterministic finite state machines, test derivation, fault coverage

On the authors:
Aleksandr S. Tvardovskii, orcid.org/0000-0001-7705-7214, Master student,
National Research Tomsk State University,
36 Lenin Ave., Tomsk, 634050, Russia, e-mail: tvardal@mail.ru

Khaled El-Fakih, PhD in computer science, professor,
American University of Sharjah,
University City, Sharjah, 26666, United Arab Emirates, e-mail: kelfakih@aus.edu

Maksim L. Gromov, orcid.org/0000-0002-2990-8245, PhD in computer science, associate professor,
National Research Tomsk State University,
36 Lenin Ave., Tomsk, 634050, Russia, e-mail: maxim.leo.gromov@gmail.ru

Nina V. Yevtushenko, orcid.org/0000-0002-4006-1161, doctor of technical sciences, professor,
National Research Tomsk State University,
36 Lenin Ave., Tomsk, 634050, Russia, e-mail: nyevtush@gmail.com

Acknowledgments:
This work is partly supported by RSF Project No. 16-49-03012.