

К синтезу адаптивных различающих последовательностей для конечных автоматов

¹ А.С. Твардовский <tvardal@mail.ru>

^{1,2,3} Н.В. Евтушенко <nyevtush@gmail.com>

¹ *Национальный исследовательский Томский Государственный университет, 634050, Российская Федерация, г. Томск, пр. Ленина, 36*

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

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

Аннотация. Конечные автоматы широко используются при построении проверяющих тестов для управляющих систем с гарантированной полнотой обнаружения неисправностей. В ряде случаев такие тесты достигают экспоненциальной длины относительно размеров автомата-спецификации, что мотивирует исследования по оптимизации проверяющих тестов. Существование последовательностей, различающих каждую пару состояний в автомате-спецификации, может существенно сократить длину теста, если такие последовательности достаточно короткие. Более того, при описании современных систем часто приходится учитывать опциональность неформальной спецификации, и соответственно, использовать методы синтеза тестов для недетерминированных автоматов; последнее в большинстве случаев повышает длину тестов. Адаптивные различающие последовательности существуют чаще, чем безусловные, и, как правило, имеют меньшую длину, что делает их выбор более предпочтительным для синтеза тестов. В настоящей работе мы исследуем свойства адаптивных различающих последовательностей и оптимизируем метод построения таковых для полностью определённых, возможно, недетерминированных конечных автоматов. Предложенный подход основан на ограничении размеров различающего автомата, по которому строится различающий тестовый пример, служащий удобной формой представления адаптивной различающей последовательности. Проведённые эксперименты позволили оценить длину и вероятность существования адаптивных различающих последовательностей для случайно сгенерированных автоматов с различной степенью недетерминизма. Также в работе рассмотрен специальный класс так называемых автоматов без слияний, которые описывают широкий класс реальных систем и обладают «хорошими» для синтеза тестов свойствами; в частности, для таких автоматов практически всегда существуют адаптивные различающие последовательности, если для каждой пары «состояние, входной символ» существует не более трех различных переходов, т.е. степень недетерминизма в автомате не больше трех.

Ключевые слова: конечный автомат; тестовый пример; адаптивная различающая последовательность

DOI: 10.15514/ISPRAS-2018-30(4)-9

Для цитирования: Твардовский А.С., Евтушенко Н.В. К синтезу адаптивных различающих последовательностей для конечных автоматов. Труды ИСП РАН, том 30, вып. 4, 2018 г., стр. 139-154. DOI: 10.15514/ISPRAS-2018-30(4)-9

1. Введение

Конечные автоматы [1] широко используются при построении тестов для управляющих систем с гарантированной полнотой обнаружения неисправностей. Хорошо известными методами синтеза тестов являются W-метод [2] и его различные модификации для детерминированных и недетерминированных автоматов [3, 4]. При тестировании на основе конечно автоматных моделей проверяющий тест строится относительно заданной модели неисправности, определяющей множество неконформных реализаций, которые могут быть обнаружены по выходным реакциям тестируемой системы на входные последовательности теста. Конечно-автоматные методы синтеза тестов часто основаны на идентификации состояний спецификации, осуществляемой при помощи безусловных или адаптивных различающих последовательностей [5]. Безусловные различающие последовательности фиксируются до начала эксперимента, в то время как следующий входной символ в адаптивной последовательности зависит от реакции тестируемой системы на предыдущие входные воздействия.

Построенные по конечному автомату тесты могут достигать экспоненциальной длины относительно числа состояний исследуемого автомата, что во многом обусловлено высокой длиной различающих последовательностей и мотивирует исследования по оптимизации методов её построения. Известно, что проверяющий тест может быть значительно короче, если спецификация обладает последовательностью, которая различает каждую пару состояний автомата [3, 10], однако, такие последовательности не всегда существуют. Адаптивные различающие последовательности существуют чаще, чем безусловные, и более того, обычно имеют меньшую длину, что делает их более предпочтительным выбором при синтезе тестов.

Для построения адаптивной различающей последовательности часто используется дерево преемников. Однако, для дерева преемников недетерминированного автомата не существует необходимых и достаточных условий существования различающего тестового примера. В [9] авторы предлагают альтернативный подход, основанный на построении различающего автомата, и устанавливают необходимые и достаточные условия существования адаптивной различающей последовательности для такого представления.

В настоящей работе мы рассматриваем проблему построения адаптивных различающих последовательностей для полностью определённых, возможно, недетерминированных автоматов и предлагаем оптимизированный подход к построению таких последовательностей на основе различающего автомата. Эксперименты, проведённые со случайно сгенерированными недетерминированными автоматами, позволили оценить эффективность предложенного подхода и сложность (длину) адаптивных различающих экспериментов.

Структура статьи следующая. Раздел 2 содержит необходимые определения из теории автоматов. Оптимизированная процедура построения адаптивной различающей последовательности с использованием различающего автомата представлена в разделе 3. Раздел 4 содержит результаты проведенных компьютерных экспериментов.

2. Основные определения

В данном разделе мы вводим основные определения и обозначения, взятые преимущественно из работ [6, 9].

2.1 Конечный автомат

Под *конечным автоматом* S понимается пятерка $(S, I, O, \lambda_S, s_0)$, где S, I , и O – конечные непустые множества состояний, входных и выходных символов соответственно, s_0 – начальное состояние, $\lambda_S \subseteq S \times I \times O \times S$ – отношение переходов. Автомат S называется *недетерминированным*, если для некоторой пары $(s, i) \in S \times I$, существует несколько различных пар $(o, s') \in O \times S$, таких что $(s, i, o, s') \in \lambda_S$, иначе, автомат называется *детерминированным*. Автомат S называется *полностью определённым*, если для каждой пары $(s, i) \in S \times I$ существует переход $(s, i, o, s') \in \lambda_S$; иначе, автомат называется *частичным*. Автомат S называется *наблюдаемым*, если для каждой пары $(s, i, o, s_1), (s, i, o, s_2) \in \lambda_S$ справедливо $s_1 = s_2$.

В настоящей работе мы рассматриваем полностью определённые, возможно, недетерминированные наблюдаемые конечные автоматы. Исключением являются тестовые примеры, которые по определению, являются частичными конечными автоматами и будут рассмотрены далее. Пример полностью определённого недетерминированного конечного автомата представлен на рисунке 1. Данный автомат имеет 4 состояния, 3 входных символа и 2 выходных символа.

Для входного символа i и выходного символа o автомата S , т.е. для входо-выходной пары i/o , состояние s' называется *io-преемником* состояния s , если в автомате существует переход (s, i, o, s') . В наблюдаемом автомате, *io-преемник* всегда определяется единственным образом. Отметим, что *io-преемник* для некоторого состояния s может быть пустым, если в автомате не существует ни одного перехода вида (s, i, o, s') , где s' – произвольное

состояние автомата. *Входо-выходной последовательностью* автомата S в состоянии s называется последовательность входо-выходных пар $i_1/o_1, \dots, i_n/o_n$, где *выходная последовательность* o_1, \dots, o_n представляет собой реакцию автомата на *входную последовательность* i_1, \dots, i_n в состоянии s . Например, для автомата S (рисунок 1) в состоянии 3, входной последовательности 1, 0, 2 соответствует выходная реакция 0, 0, 0. Таким образом, 1/0, 0/0, 2/0 – входо-выходная последовательность в состоянии s автомата S .

Конечный автомат S называется *автоматом без слияний* [7], если для каждой пары переходов $(s_1, i, o, s_1'), (s_2, i, o, s_2') \in \lambda_S, i \in I, o \in O$, имеет место $s_1' \neq s_2'$; другими словами, в автомате без слияний переходы из двух различных состояний по одной входо-выходной паре не могут переводить автомат в одно и то же состояние. Ситуация, когда не пустые *io*-преемники для некоторой пары различных состояний совпадают, называется *слиянием* таких переходов. Например, для автомата S (рис. 1) слияние существует для переходов (2, 2, 1, 3) и (3, 2, 1, 3).

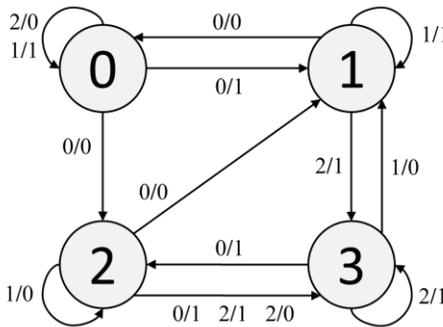


Рис. 1. Полностью определённый недетерминированный автомат S
 Fig. 1. Complete nondeterministic FSM S

2.2 Тестовый пример

Входная последовательность α называется *адаптивной*, если следующий входной символ в последовательности зависит от реакции автомата на предыдущий входной символ. Адаптивные входные последовательности часто представляются в форме специального конечного автомата, называемого тестовым примером [8].

Тестовым примером $TC(I, O)$ с входным алфавитом I и выходным алфавитом O называется инициальный связный наблюдаемый конечный автомат $T = (T, I, O, \lambda_T, t_0)$ с ациклическим графом переходов. В каждом состоянии тестового примера определены переходы не более чем по одному входному символу, а состояния, в которых не определено ни одного перехода, называются *тупиковыми (deadlock)*. Тестовый пример $TC(I, O)$ является частичным автоматом, если $|I| > 1$, и представляет собой адаптивную входную

последовательность для автомата $S = (S, I, O, \lambda_S, s_0)$. *Высота* тестового примера определяется как длина максимальной входо-выходной последовательности из начального в тупиковое состояние и соответствует максимальной длине входной последовательности, которая может быть подана на систему в ходе адаптивного эксперимента.

Тестовый пример ТС представляет *адаптивную различающую последовательность* для состояний s_1 и s_2 автомата S , если каждая входо-выходная последовательность из начального в терминальное состояние тестового примера возможна только в одном из состояний s_1 или s_2 . В таком случае, ТС называется *различающим тестовым примером*, а состояния s_1 и s_2 *адаптивно различимыми*. Адаптивная последовательность, различающая каждую пару состояний автомата S , называется *адаптивной различающей последовательностью* для автомата S . Различающий тестовый пример для автомата на рисунке 1 представлен на рис. 2.

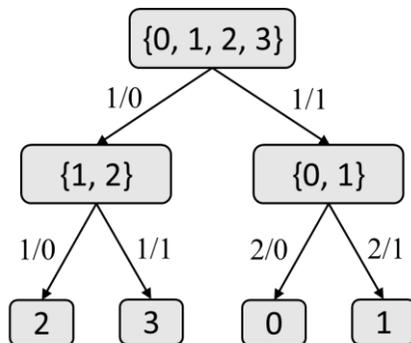


Рис. 2. Различающий тестовый пример для автомата S
Fig. 2. Distinguishing test case for FSM S

Для того чтобы идентифицировать начальное состояние автомата S , различающий тестовый пример ТС подаётся на автомат S следующим образом. Эксперимент начинается с подачи на автомат единственного входного символа i_1 , определённого в начальном состоянии s_0 автомата ТС. Далее различающий пример выполняет переход в состояние s_1 по входо-выходной паре i_1/o_1 , где o_1 – наблюдаемая реакция автомата S на входной символ i_1 . Далее на автомат S подаётся входной символ, определённый в состоянии s_1 . Эксперимент продолжается итеративно, пока в различающем тестовом примере не будет достигнуто тупиковое состояние; по соответствующей входо-выходной последовательности можно однозначно определить состояние, в котором находился автомат S до начала эксперимента.

Например, для различающего тестового примера на рисунке 2 первым будет подан входной символ 1. Далее, по реакции 0 или 1 автомата S (рисунок 1)

определяется следующий входной символ 1 или 2 соответственно. Если в ходе эксперимента была получена входо-выходная последовательности 1/1, 2/1, то до начала эксперимента автомат S находился в состоянии 1.

3. Построение различающего тестового примера

Метод построения различающего тестового примера на основе различающего автомата был предложен в [9]. Различающий автомат S_{dist} для полностью определённого наблюдаемого, возможно, недетерминированного автомата $S = (S, I, O, \lambda_S)$ строится по следующим правилам. Множества входных и выходных символов S_{dist} совпадает с таковыми для исходного конечного автомата S , в то время как состояния S_{dist} соответствуют непустым подмножествам состояний S , содержащим более одного элемента; кроме того, в различающем автомате присутствует специальное состояние F . Построение переходов S_{dist} начинается в начальном состоянии S_{dist} , которое соответствует подмножеству, содержащему все состояния автомата S . Различающий автомат S_{dist} представляет собой минимальный автомат, построенный по следующим правилам.

Пусть b состояние различающего автомата S_{dist} .

- 1) Переход (b, i, o, b') существует в автомате S_{dist} , если и только если b' не является одноэлементным подмножеством, для каждого $o' \in O$ непустые io' -преемники для любой пары различных состояний из b не совпадают, и b' – непустое множество io' -преемников состояний из множества b . В таком случае, состояние b' добавляется в множество состояний S_{dist} .
- 2) Переход (b, i, o, F) существует в автомате S_{dist} , если и только если существует выходной символ $o' \in O$, такой что непустые io' -преемники для некоторой пары различных состояний из b совпадают (имеет место слияние переходов). В таком случае, состояние F добавляется в множество состояний S_{dist} .
- 3) Переход автомата S_{dist} из состояния b по входному символу i не определён, если и только если для каждого $o' \in O$, не пустые io' -преемники любой пары различных состояний из b не совпадают, и каждый io' -преемник состояний из b является одноэлементным.

В состоянии F существует переход (F, i, o, F) для каждой входо-выходной пары i/o автомата S_{dist} .

Входной символ i различающего автомата S_{dist} называется *неопределённым* в состоянии b , если в автомате не существует переходов из состояния b по входному символу i , т.е. для каждого $o' \in O$, не пустые io' -преемники любой пары различных состояний из b не совпадают, и каждый io' -преемник состояний из b является одноэлементным подмножеством. Для конечного автомата S на рис. 1 соответствующий различающий автомат S_{dist} , представленный на рис. 3, имеет 10 состояний.

Для того чтобы определить, обладает ли конечный автомат S различающим тестовым примером, состояния различающего автомата S_{dist} с неопределённым входным символом удаляются из различающего автомата со всеми переходами в такие состояния. Состояния удаляются итеративно, пока либо в начальном состоянии не появляется неопределённый входной символ, либо в различающем автомате не заканчиваются неопределённые входные символы. При удалении состояния b из S_{dist} мы сохраняем один из входных символов i , неопределённых в состоянии b , в специальном массиве UN , $UN(b) = i$.

Если на некотором этапе удаления состояний, в начальном состоянии различающего автомата существует неопределённый входной символ, то различающий тестовый пример D может быть построен при помощи сохранённых в массиве UN входных символов. В этом случае начальное состояние d_0 автомата D соответствует начальному состоянию b_0 автомата S_{dist} . Входной символ $UN(b_0)$ представляет собой единственный входной символ, определённый в состоянии d_0 автомата D , и переход $(d_0, UN(b_0), o, d)$ существует в D , если и только если в S_{dist} (до удаления состояний) существует переход $(b_0, UN(b_0), o, b)$, где для состояний b и d , соответствующие подмножества состояний автомата S совпадают.

Если для некоторого $o \in O$ не существует перехода $(b_0, UN(b_0), o, b)$ в автомате S_{dist} , то в D добавляется переход $(d_0, UN(b_0), o, F)$, где F – тупиковое состояние. Переходы для других состояний различающего тестового примера D формируются аналогично. Отметим, что для более удобного представления различающего тестового примера, вместо единственного состояния F , можно использовать множество тупиковых состояний, каждое из которых помечается состоянием, в котором находился автомат до начала эксперимента при наблюдении соответствующей входе-выходной последовательности (если таковое существует). Для конечного автомата S на рис. 1 соответствующий различающий тестовый пример представлен на рисунке 2, имеет длину 2 и может быть построен по различающему автомату S_{dist} (рис. 3).

В [9] устанавливается следующее необходимое и достаточное условие существования различающего примера.

Утверждение 1. Для полностью определённого наблюдаемого, возможно, недетерминированного автомата S существует различающий тестовый пример, если и только если в соответствующем различающем автомате S_{dist} не существует полностью определённого подавтомата.

В [9] также показано, что длина различающего тестового примера может достигать значения $2^{n-1} - 1$ для полностью определённого недетерминированного наблюдаемого автомата S с n состояниями. В этом случае, все возможные подмножества состояний автомата S , не являющиеся одноэлементными, могут быть состояниями различающего автомата S_{dist} , и соответственно, процедура построения различающих тестовых примеров становится достаточно сложной.

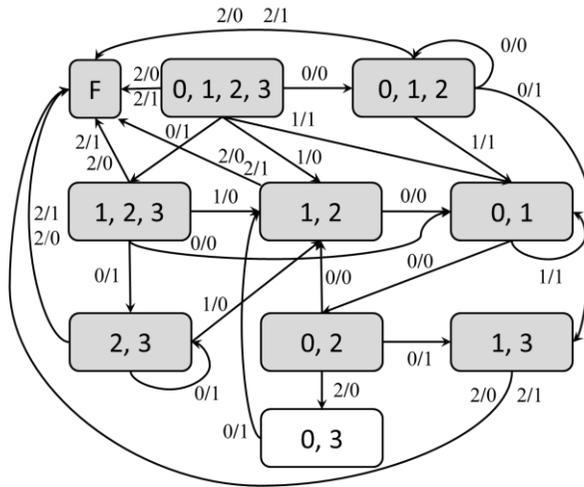


Рис. 3. Различающий автомат S_{dist}
 Fig. 3. Distinguishing FSM S_{dist}

Для того чтобы оптимизировать синтез адаптивной различающей последовательности, мы предлагаем ограничивать построение различающего автомата допустимой длиной адаптивной различающей последовательности, а именно, строить различающий автомат S_{dist}^L , содержащий лишь состояния, достижимые из начального состояния по входо-выходным последовательностям длины не больше некоторого $L \geq 1$. Такой сокращённый различающий автомат позволяет найти различающие тестовые примеры высоты не больше L . Предложенный подход согласуется с результатами из [9], где утверждается, что длина адаптивной различающей последовательности не превышает 15 для случайно сгенерированных конечных автоматов даже с большим (более 100) числом состояний. Ввиду того, что длина различающего тестового примера для большинства автоматов не слишком большая, имеет смысл ограничить длину, на которой осуществляется поиск тестового примера.

Таким образом, под автоматом S_{dist}^L понимается различающий автомат, содержащий лишь состояния, достижимые из начального состояния по входо-выходным последовательностям длины не больше L , где L – целое неотрицательное число, и состояние F . Отметим также, что все переходы S_{dist}^L из состояний, достижимых только по входной последовательности длины больше L ведут в состояние F . Например, для автомата на рис. 1, различающий автомат S_{dist}^0 содержит только начальное состояние $\{0, 1, 2, 3\}$, в то время как S_{dist}^1 содержит еще четыре состояния, достижимых из начального состояния по входной последовательности длины 1 (рис. 3).

Аналогично утверждению 1 из [9], справедлива следующая теорема.

Теорема 2. Для полностью определённого наблюдаемого, возможно, недетерминированного конечного автомата S существует различающий тестовый пример высоты не больше L , если и только если различающий автомат S^L_{dist} не имеет полностью определённого подавтомата.

Доказательство. Пусть в автомате S существует различающий тестовый пример длины L . Тогда по утверждению 1, различающий автомат S_{dist} не имеет полностью определённого подавтомата. Итеративным удалением состояний, достижимых по последовательности длины не больше L можно показать, что для автомата S^L_{dist} также не существует полностью определённого подавтомата. В то же время, если автомат S^L_{dist} не имеет полностью определённого подавтомата, то тестовый пример может быть непосредственно построен по приведённой выше процедуре, аналогичной процедуре из [9].

Таким образом, проверка существования различающего тестового примера высоты не больше L может быть проведена итеративным удалением состояний сокращённого различающего автомата S^L_{dist} . Если начальное состояние S^L_{dist} в результате имеет неопределённый входной символ, то S^L_{dist} не имеет полностью определённого подавтомата и соответствующий различающий тестовый пример может быть построен описанным выше методом.

Например, различающий автомат на рис. 3 имеет 10 состояний, однако, различающий тестовый пример может быть построен на основе автомата S^2_{dist} , который имеет 9 состояний (выделены серым цветом). При удалении состояний с неопределённым входным символом можно убедиться, что различающий автомат S^2_{dist} не имеет полностью определённого подавтомата, и соответствующий различающий тестовый пример (рисунок 2) может быть построен по описанной выше процедуре.

Таким образом, мы предлагаем следующую процедуру построения различающего тестового примера для полностью определённого, возможно недетерминированного, наблюдаемого конечного автомата.

Оптимизированная процедура построения различающего тестового примера

Вход: полностью определённый наблюдаемый автомат S , целое число $L \geq 1$

Выход: Различающий тестовый пример высоты не больше L для автомата S , сообщения ‘Автомат S не имеет различающего тестового примера высоты не больше L ’ или ‘Автомат S не имеет различающего тестового примера’

$l := 1$, массив неопределённых входных символов $UN := \emptyset$, различающий автомат S^0_{dist} с единственным начальным состоянием, совпадающим с множеством состояний автомата S ; $Q = S^0_{dist}$;

Шаг 1. Добавить в S^{l-1}_{dist} состояния, достижимые по входу-выходной последовательности длины l , т.е., построить автомат S^l_{dist} и копировать все новые переходы и состояния в автомат Q .

Если $S_{dist}^l = S_{dist}^{l-1}$, **то** вывести сообщение ‘Автомат S не имеет различающего тестового примера’ и **завершить процедуру**
иначе Шаг 2.

Шаг 2. Итеративно удалить из S_{dist}^l каждое состояние b , в котором существует неопределённый входной символ, выбрать неопределённый входной символ i , положив $UN(b) = i$.

Если в начальном состоянии S_{dist}^l существует неопределённый входной символ, **то** Шаг 3;

иначе

Если $l + 1 > L$ **то** вывести сообщение ‘Автомат S не имеет различающего тестового примера высоты не больше L ’ и **завершить процедуру**;

иначе $l = l + 1$ и Шаг 1.

Шаг 3. Построить различающий тестовый пример по массиву неопределённых входных символов UN и различающему автомату Q , который представляет собой автомат S_{dist}^l до удаления состояний.

Завершить процедуру

Теорема 3. Для полностью определённого наблюдаемого, возможно, недетерминированного автомата S и целого числа $L > 0$ предложенная оптимизированная процедура возвращает различающий тестовый пример, если и только если в автомате S существует различающий тестовый пример длины не больше L .

Доказательство. Из теоремы 2 и результатов [9] следует, что если автомат S имеет адаптивную различающую длины L , то этот различающий тестовый пример может быть построен по различающему автомату S_{dist}^L . Покажем далее, что удаление состояний на шаге 2 вышеописанного алгоритма не влияет на построение различающего примера. Пусть для числа L был построен автомат S_{dist}^L и состояние b различающего автомата S_{dist}^L имеет неопределённый входной символ. Последнее означает, что для подмножества состояний автомата S , соответствующего состоянию b , существует (адаптивная) различающая последовательность длины не более $L - L'$, где L' – длина входной последовательности из начального состояния S_{dist}^L в b .

Отметим также, что различающий автомат, соответствующий автомату S_{dist}^L до удаления состояний, сохраняется в автомате Q , на основе которого различающий тестовый пример может быть восстановлен по множеству неопределённых входных символов UN . Таким образом, состояние b будет удалено, как только для него будет найдена кратчайшая (адаптивная) различающая последовательность, которая позже может быть восстановлена по автомату Q и множеству UN , и включена в различающий тестовый пример.

Оптимизированная процедура построения различающего тестового примера была реализована программным образом, и далее мы приводим результаты проведённых компьютерных экспериментов.

4. Экспериментальные результаты

В настоящем разделе мы приводим результаты компьютерных экспериментов по оценке эффективности предложенного оптимизированного подхода к построению различающих тестовых примеров, а также исследуем длину адаптивных различающих последовательностей для различных классов конечных автоматов.

4.1 Случайно сгенерированные конечные автоматы

Первая серия экспериментов проводилась со случайно сгенерированными конечными автоматами для оценки вероятности существования адаптивной различающей последовательности в зависимости от числа недетерминированных переходов автомата. Количество nd переходов автомата для каждой пары ‘состояние, входной символ’ задается заранее. Проведённые эксперименты (рис. 4) показали, что различающий тестовый пример существует крайне редко, при $nd > 3$, и соответственно, последующие серии экспериментов проводились с автоматами, для которых $nd \leq 3$.

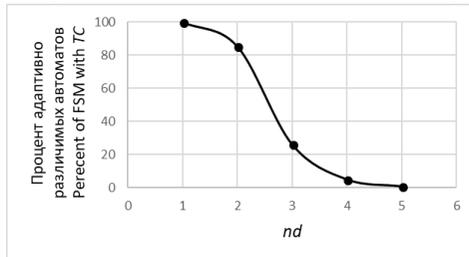


Рис. 4. Процент автоматов, для которых существует различающий тестовый пример

Fig. 4. Percentage of FSMs when a distinguishing test case exists

На рис. 5 представлена средняя высота различающего тестового примера в зависимости от числа состояний автомата. Эксперименты проводились с автоматами, для которых $|I| = |O| = 10$, и которые имели различное число недетерминированных переходов. Нижняя кривая построена для детерминированных автоматов, средняя и верхняя кривая для значений $nd = 2$ и 3 соответственно.

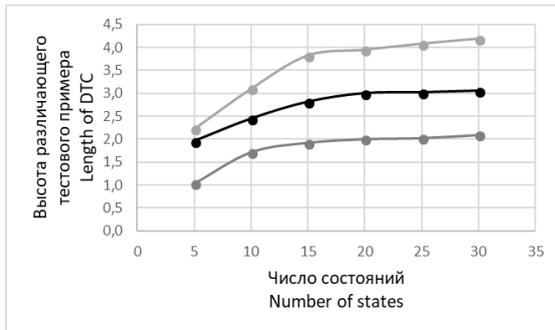


Рис. 5. Высота различающего тестового примера для детерминированных и недетерминированных автоматов

Fig. 5. Length of a distinguishing test case for deterministic and nondeterministic FSMs

Из проведённых экспериментов следует, что для случайно сгенерированных автоматов различающий тестовый пример либо имеет небольшую высоту, либо не существует. Последнее обусловлено тем, что при случайной генерации высока вероятность существования слияний для каждого входного символа. Более того, ввиду большого количества слияний, многие состояния различающего автомата имеют переходы лишь в состояние F , и соответственно, число состояний различающего автомата невелико. Таким образом, предложенная в предыдущем разделе оптимизация оправдана лишь при большом количестве недетерминированных переходов, при котором число состояний различающего автомата и длина различающего примера возрастают.

На рис. 6 отображена зависимость времени построения различающего тестового примера от числа состояний для оптимизированной (нижняя кривая) и неоптимизированной (верхняя кривая) процедур при $|I| = |O| = 10$ и $nd = 3$. Эффективность оптимизированной процедуры возрастает с увеличением числа состояний автомата. Отметим также, что для автоматов с $|S| = 30$, $|I| = |O| = 10$ и $nd = 2$, оптимизированная процедура работает примерно в два раза быстрее оригинальной, в то время как для $nd = 3$ предложенная оптимизация значительно ускоряет выполнение процедуры при прочих равных условиях (рисунок 6).

Заметим, что по результатам проведённых экспериментов случайно сгенерированные автоматы имеют много слияний, что не всегда соответствует автоматам, описывающим поведение реальных систем. В следующем разделе мы рассматриваем специальный класс конечных автоматов без слияний, обладающих рядом полезных свойств при построении различающих тестовых примеров.

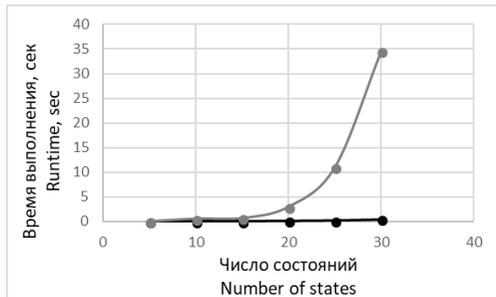


Рис. 6. Время выполнения оригинальной и оптимизированной процедуры построения различающего тестового примера

Fig. 6. The runtime of original and optimized procedures for the distinguishing test case derivation

4.2 Конечные автоматы без слияний

Автомат без слияний обладает адаптивной различающей последовательностью, если и только если каждая пара состояний в нём адаптивно различима, т.е. обладает адаптивной различающей последовательностью [7]. Таким образом, можно предполагать, что для такого класса автоматов тест полиномиальной длины существует чаще. Ввиду последнего, а также того, что автоматы без слияний описывают достаточно широкий класс реальных управляющих систем, такие автоматы представляют особый интерес для исследований.

Зависимость высоты различающих тестовых примеров от числа состояний для случайно сгенерированных автоматов без слияний совпадает с таковой для произвольных автоматов. В частности, для автоматов с $|S| = 100$, $|I| = |O| = 10$ и $nd \leq 2$, высота различающих тестовых примеров не превышает значения 5.

В то же время вероятность существования адаптивной различающей последовательности для автоматов без слияний существенно выше и стремится к 100 процентам для автоматов с $|S| = 100$, $|I| = |O| = 10$ и $nd \leq 2$. Отметим также, что для произвольных автоматов с $|S| = 100$, $|I| = |O| = 10$ и $nd = 2$ время выполнения неоптимизированной и оптимизированной процедур построения различающего тестового примера не превышает 60 секунд, при этом оптимизированная процедура работает примерно в два раза быстрее для автоматов со 100 состояниями. Эффективность оптимизированной процедуры увеличивается с ростом состояний и степенью недетерминизма.

5. Заключение

В настоящей работе была исследована задача построения адаптивных различающих последовательностей для полностью определённых наблюдаемых, возможно, недетерминированных конечных автоматов на

основе соответствующего различающего автомата. Предложена оптимизация построения различающего автомата, по которому строится адаптивная различающая последовательность. Было экспериментально показано, что длина адаптивной различающей последовательности для случайно сгенерированных конечных автоматов не достигает «худшей» экспоненциальной оценки.

Однако вероятность существования таких последовательностей уменьшается с увеличением недетерминизма и близка к нулю, когда число переходов для каждой пары «состояние, входной символ» превышает три. Эффективность предложенного подхода к оптимизации была исследована путём проведения компьютерных экспериментов со случайно сгенерированными автоматами. Было также показано, что для специального класса автоматов без слияний, адаптивная различающая последовательность почти всегда существует.

Благодарности

Работа выполнена при поддержке гранта РНФ No. 16-49-03012.

Список литературы

- [1]. Гилл А. Введение в теорию конечных автоматов. М., Наука, 1966, 272 стр.
- [2]. Chow, T.S. Test design modeled by finite-state machines. *IEEE Transactions on Software Engineering*, vol. 4, No 3, 1978, pp. 178-187
- [3]. Petrenko A. and Yevtushenko N. Conformance Tests as Checking Experiments for Partial Nondeterministic FSM. In *Proceedings of the 5th International Workshop on Formal Approaches to Testing of Software (FATES 2005)*, LNCS 3997, 2005, pp. 118-133
- [4]. Dorofeeva R., El-Fakih K., Maag S., Cavalli A.R., Yevtushenko N. FSM-based conformance testing methods: a survey annotated with experimental evaluation. *Information and Software Technology*, 52, 2010, pp. 1286-1297
- [5]. Alur R., Courcoubetis C., Yannakakis M. Distinguishing tests for nondeterministic and probabilistic machines, In *Proc. of the 27th ACM Symposium on Theory of Computing*, 1995, pp. 363-372.
- [6]. Petrenko A., Yevtushenko N. Adaptive testing of deterministic implementations specified by nondeterministic FSMs, In *Proc. of the International Conference on Testing Software and Systems*, LNCS, vol. 7019, 2011, pp. 162-178
- [7]. Yevtushenko N., Kushik N. Decreasing the length of adaptive distinguishing experiments for nondeterministic merging-free finite state machines. In *Proceedings of IEEE East-West Design & Test Symposium (EWDTS)*. 2015. P. 338–341
- [8]. Yevtushenko N., El-Fakih K., and Ermakov, A. On-the-fly construction of adaptive checking sequences for testing deterministic implementations of nondeterministic specifications, LNCS, vol. 9976, 2016, pp. 139–152
- [9]. El-Fakih K., Yevtushenko N., Kushik N. Adaptive distinguishing test cases of nondeterministic finite state machines: test case derivation and length estimation. *Formal Aspects of Computing* vol. 30, issue 2, 2018, pp. 319-332
- [10]. Tvardovskii A. Refining the Specification FSM When Deriving Test Suites w.r.t. the Reduction Relation. LNCS, vol 10533, 2017, pp. 333-339

- [11]. Shabaldina N. Gromov M. FSMTest-1.0: a manual for researches. In Proceedings of the 13th International symposium on IEEE EAST-WEST DESIGN & TEST SYMPOSIUM (EWDTS'15), 2015, pp. 216–219

Deriving adaptive distinguishing sequences for Finite State Machines

¹ A.S. Tvardovskii <tvardal@mail.ru>

^{1,2,3} N.V. Yevtushenko <nyevtush@gmail.com>

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

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

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

Abstract. FSM (Finite State Machines) are widely used for deriving tests with guaranteed fault coverage for control systems. Distinguishing sequences (DS) are used in FSM based testing for state identification and can significantly reduce the size of a returned complete test suite. In some cases, length of distinguishing sequence can be exponential with respect to the size of the FSM specification. Moreover, DS can be even longer for non-deterministic FSMs, which are used for the specification optionality description when deriving tests for real systems. Unfortunately, DS not always exist for deterministic and non-deterministic FSMs. Adaptive DS (or corresponding distinguishing test cases (DTC)) are known to exist more often and be much shorter than the preset ones that makes adaptive DS attractive for test derivation. In this paper, we investigate the properties of adaptive DS and propose an approach for optimizing the procedure for the adaptive DS derivation. For this purpose, we propose to limit the height of a DTC and correspondingly to reduce the size of a distinguishing FSM that is used for the DTC derivation in the original procedure. The efficiency of a proposed optimized procedure is evaluated by computer experiments for randomly generated FSMs up to 100 states. We also present the experimental results on checking the percentage of randomly generated FSMs when a DTC exists.

Keywords: Finite State Machine (FSM), test case, adaptive distinguishing sequence.

DOI: 10.15514/ISPRAS-2018-30(4)-9

For citation: Tvardovskii A.S., Yevtushenko N.V. Deriving adaptive distinguishing sequences for Finite State Machines. *Trudy ISP RAN/Proc. ISP RAS*, vol. 30, issue 4, 2018. pp. 139-154 (in Russian). DOI: 10.15514/ISPRAS-2018-30(4)-9

References

- [1]. Gill A. Introduction to the Theory of Finite-State Machines. McGraw-Hill, 1964, 207 p.
- [2]. Chow, T.S. Test design modeled by finite-state machines. *IEEE Transactions on Software Engineering*, vol. 4, No 3, 1978, pp. 178-187

- [3]. Petrenko A. and Yevtushenko N. Conformance Tests as Checking Experiments for Partial Nondeterministic FSM. In Proceedings of the 5th International Workshop on Formal Approaches to Testing of Software (FATES 2005), LNCS 3997, 2005, pp. 118-133
- [4]. Dorofeeva R., El-Fakih K., Maag S., Cavalli A.R., Yevtushenko N. FSM-based conformance testing methods: a survey annotated with experimental evaluation. *Information and Software Technology*, 52, 2010, pp. 1286-1297
- [5]. Alur R., Courcoubetis C., Yannakakis M. Distinguishing tests for nondeterministic and probabilistic machines, In Proc. of the 27th ACM Symposium on Theory of Computing, 1995, pp. 363-372.
- [6]. Petrenko A., Yevtushenko N. Adaptive testing of deterministic implementations specified by nondeterministic FSMs, In Proc. of the International Conference on Testing Software and Systems, LNCS, vol. 7019, 2011, pp. 162-178
- [7]. Yevtushenko N., Kushik N. Decreasing the length of adaptive distinguishing experiments for nondeterministic merging-free finite state machines. In Proceedings of IEEE East-West Design & Test Symposium (EWDTS). 2015. P. 338–341
- [8]. Yevtushenko N., El-Fakih K., and Ermakov, A. On-the-fly construction of adaptive checking sequences for testing deterministic implementations of nondeterministic specifications, LNCS, vol. 9976, 2016, pp. 139–152
- [9]. El-Fakih K., Yevtushenko N., Kushik N. Adaptive distinguishing test cases of nondeterministic finite state machines: test case derivation and length estimation. *Formal Aspects of Computing* vol. 30, issue 2, 2018, pp. 319-332
- [10]. Tvardovskii A. Refining the Specification FSM When Deriving Test Suites w.r.t. the Reduction Relation. LNCS, vol 10533, 2017, pp. 333-339
- [11]. Shabdina N. Gromov M. FSMTest-1.0: a manual for researches. In Proceedings of the 13th International symposium on IEEE EAST-WEST DESIGN & TEST SYMPOSIUM (EWDTS'15), 2015, pp. 216–219