

Минимизация автоматов с таймаутами и временными ограничениями

А.С. Твардовский <tvardal@mail.ru>

Н.В. Евтушенко <nyevtush@gmail.com>

М.Л. Громов <maxim.leo.gromov@gmail.com>

Национальный исследовательский Томский государственный университет,
634050, Россия, г. Томск, пр. Ленина, 36

Аннотация. Конечные автоматы широко используются для анализа и синтеза управляющих систем. При описании систем, поведение которых зависит от времени, конечный автомат расширяется введением временных аспектов и вводится понятие временного автомата. В настоящей работе мы рассматриваем проблему минимизации автоматов с таймаутами и временными ограничениями, поскольку сложность многих задач в теории автоматов существенно зависит от размеров исследуемой системы. Поведение временного автомата может быть достаточно точно описано соответствующим конечным автоматом, и предлагаемый метод минимизации числа состояний системы основан на использовании такой конечно автоматной абстракции. Более того, далее мы минимизируем и временные аспекты автоматного описания, сокращая продолжительность таймаутов и число переходов с временными ограничениями. Мы также показываем, что для полностью определённого детерминированного временного автомата существует единственная минимальная (каноничная) форма, т. е. единственный приведённый по состояниям и временным аспектам автомат с таймаутами и временными ограничениями, поведение которого совпадает с исходным временным автоматом; например, такая минимальная форма может быть использована при построении проверяющих тестов для проверки функциональных и нефункциональных требований к тестируемой реализации. Предложенный метод к минимизации временных аспектов на основе конечно автоматной абстракции может быть применён и для частных случаев рассматриваемой модели, т. е. для минимизации детерминированных полностью определённых автоматов только с таймаутами или только с временными ограничениями.

Ключевые слова: временные автоматы; приведённая форма; минимальная форма

DOI: 10.15514/ISPRAS-2017-29(4)-8

Для цитирования: Твардовский А.С., Евтушенко Н.В., Громов М.Л. Минимизация автоматов с таймаутами и временными ограничениями. *Труды ИСП РАН*, том 29, вып. 4, 2017 г., стр. 139-154. DOI: 10.15514/ISPRAS-2017-29(4)-8

1. Введение

Конечно-автоматные модели широко используются при анализе и синтезе дискретных систем [1, 2], в частности при реализации компонент управляющих систем [3, 4, 5], причем сложность решения многих задач теории автоматов существенно зависит от числа состояний рассматриваемого автомата. Более того, многие алгоритмы синтеза тестов с гарантированной полнотой [2, 6, 7, 8] для дискретных систем разработаны только для приведённых автоматов, т.е. автоматов, в которых любая пара состояний различима.

При рассмотрении современных систем часто приходится учитывать временные аспекты в их поведении, и соответственно, понятие конечного автомата расширяется введением временных переменных [7, 9, 10]. Известны различные способы добавления временных переменных в описание цифровых компонентов, поведение которых описывается конечными автоматами. В настоящей работе мы рассматриваем автоматы с таймаутами и временными ограничениями [10], которые являются обобщением автоматов только с таймаутами и только с временными ограничениями [11, 12].

Мы предлагаем метод минимизации детерминированного полностью определённого временного автомата, используя, подобно [1], разбиение множества состояний автомата по отношению эквивалентности. По определению, состояния эквивалентны, если автомат в этих состояниях имеет одну и ту же выходную реакцию на любую входную последовательность. Мы вводим класс приведенных по состояниям и временным аспектам автоматов с таймаутами и временными ограничениями, и показываем, что такая минимальная форма является единственной с точностью до изоморфизма. Построение минимальной формы основано на специальной конечно автоматной абстракции, которая сохраняет последовательностные свойства временного автомата. Структура работы следующая. Раздел 2 содержит основные определения и обозначения. В разделе 3 рассматривается алгоритм построения конечно автоматной абстракции и обсуждаются её свойства. В разделе 4 представлен предложенный алгоритм минимизации состояний детерминированного полностью определённого автомата с таймаутами и временными ограничениями. В разделе 5 предлагается алгоритм минимизации временных аспектов автомата с таймаутами и временными ограничениями и показывается, что минимальная форма для такой модели временного автомата единственна.

2. Основные определения и обозначения

Под *конечным автоматом* понимается четвёрка $S = (S, I, O, h_S)$, где I – множество входных символов, O – множество выходных символов, S – конечное непустое множество *состояний*, $h_S \subseteq (S \times I \times O \times S)$ – *отношение переходов*. Соответственно, кортеж (s, i, o, s') описывает переход из состояния s в состояние s' под действием входного символа i с выдачей выходного

символа o . Последовательность пар входной символ/выходной символ называется входе-выходной последовательностью и обозначается α/γ , где α – входная последовательность, и γ – выходная последовательность.

Под *автоматом с таймаутами и временными ограничениями* понимается пятерка $S = (I, S, O, h_S, \Delta_S)$, где I – входной алфавит, O – выходной алфавит, S – конечное непустое множество состояний, $h_S \subseteq (S \times I \times O \times S \times \Pi \times \mathbb{Z}^+)$ – отношение переходов, $\Delta_S: S \rightarrow S \times (\mathbb{N} \cup \{\infty\})$ – функция *таймаута*, определяющая для каждого состояния максимальное время ожидания входного символа, \mathbb{N} – множество натуральных чисел, Π – множество интервалов из промежутка $[0; T)$ вида $\langle a, b \rangle$, где $\langle \in \{[, (,) \in [,]\}$, T есть таймаут в текущем состоянии и \mathbb{Z}^+ – множество целых неотрицательных чисел. Соответственно, *кортеж* (s, i, o, s', g, d) 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 s выдачей выходного символа o через d тактов времени после поступления входного символа. Иногда время d обработки входного символа называют *выходным таймаутом* или *временем задержки*. Если в некотором состоянии автомата входной сигнал не поступает в течение определенного времени, который превышает (входной) таймаут в текущем состоянии, то автомат может изменить своё состояние. Например, если $\Delta_S(s) = (s', T)$ и в состоянии s в течение T единиц времени на автомат не было подано ни одного входного символа, то автомат переходит в состояние s' . Для таймаута вида $\Delta_S(s) = (s', \infty)$ справедливо, что $s = s'$. После перехода в состояние s' по входному символу либо таймауту отсчет времени начинается с 0. Если $s = s'$, то после достижения таймаута в состоянии s отсчет времени начинается с 0. Функция $time(s, t) = s'$ [7] определяет, в каком состоянии s' находится автомат через t тактов времени, при условии, что входной символ не был подан в течение этого времени. Для временного автомата S через B будем обозначать наибольшую конечную границу для временных интервалов, которая совпадает с максимальной величиной входного таймаута.

Временным входным символом называется пара (i, t) , где i – символ входного алфавита, t – время поступления входного символа после выдачи автоматом последнего выходного символа. *Временным выходным символом* называется пара (o, d) , где o – символ выходного алфавита, d – число единиц времени между подачей входного символа и выдачей выходного символа. Для временного автомата S и входной временной последовательности $\alpha = (i_1, t_1), (i_2, t_2) \dots, (i_n, t_n)$ соответствующая выходная временная последовательность $\gamma = (o_1, d_1), (o_2, d_2) \dots, (o_n, d_n)$ называется (выходной) реакцией. При этом последовательность пар (временной входной символ / временной выходной символ) называется *временной входе-выходной последовательностью* и обозначается α/γ . Для нахождения реакции временного автомата на входной временной символ (i, t) в состоянии s , сначала определяется состояние $s' = time(s, t)$, в котором находится

автомат в момент времени t , далее определяется переход (s', i, o, s'', g, d) такой, что $(t - \Sigma) \in g$, где Σ есть сумма таймаутов при переходах по таймаутам из состояния s в состояние s' ; соответственно, при подаче входного временного символа (i, t) в состоянии s автомат производит выходную реакцию (o, d) и переходит в следующее состояние s'' . Таким образом, реакция автомата на временную входную последовательность в состоянии s вычисляется итеративно, начиная с состояния s . Вычитание величины Σ из времени поступления входного символа отражает тот факт, что отсчёт времени «сбрасывается» в 0 при выполнении автоматом переходов по таймауту.

В настоящей работе мы рассматриваем полностью определенные и детерминированные автоматы, т.е. для каждого состояния s и временного входного символа (i, t) единственным образом определяется следующее состояние s'' автомата и выходная реакция (o, d) .

На рисунке 1 представлен автомат с таймаутами и временными ограничениями S . Реакции автомата S в состоянии s_0 на временные входные символы $(i, 0)$ и $(i, 1)$ различаются и представляют собой выходные временные символы $(o_1, 1)$ и $(o_2, 3)$ соответственно. Если же входной символ не будет подан на автомат S в состоянии s_0 в течение двух единиц времени, то автомат перейдёт в состояние s_3 по таймауту.

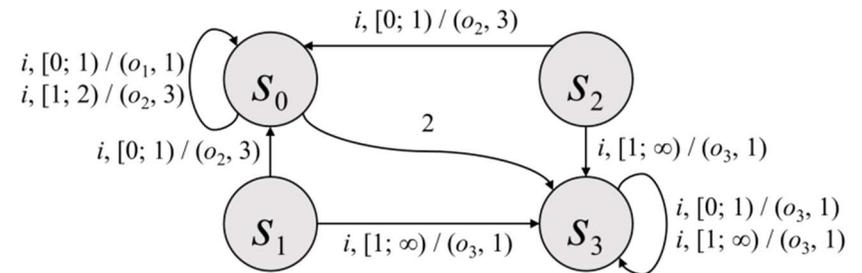


Рис. 1. Временной автомат S
Fig. 1. Timed Finite State Machines S

Состояния s и p полностью определённых детерминированных временных автоматов S и P называются *эквивалентными*, если реакции автоматов в этих состояниях совпадают на любую входную временную последовательность. Если состояния s и p не являются эквивалентными, то они называются *различимыми*. Отношение эквивалентности на множестве состояний автомата индуцирует разбиение на множестве состояний, которое называется *разбиением* по отношению эквивалентности и далее обозначается E . Любые два состояния, принадлежащие одному классу разбиения E , являются эквивалентными; любые два состояния, принадлежащие различным классам разбиения E , являются различимыми.

Два автомата S и P эквивалентны, если для каждого состояния автомата S существует эквивалентное состояние в автомате P , и для каждого состояния автомата P существует эквивалентное состояние в автомате S . Автомат называется *приведённым по состояниям*, если любые два состояния в нём различимы. *Приведённой по состояниям формой* временного автомата S называется приведённый автомат, эквивалентный S .

Временные автоматы S и P с одинаковыми входными и выходными алфавитами *изоморфны*, если между множествами состояний и переходов этих автоматов можно установить взаимно однозначное соответствие, т.е. существует взаимно-однозначное отображение $H: S \rightarrow P$, такое что $(s_i, i, o, s_j, g, d) \in h_s$, если и только если $(H(s_i), i, o, H(s_j), g, d) \in h_p$ и $\Delta_S(s_i) = (s_j, T)$, если и только если $\Delta_P(H(s_i)) = (H(s_j), T)$. Временной автомат, изоморфный заданному временному автомату, может быть получен переименованием состояний исходного автомата.

3. Конечно автоматная абстракция

Поведение автомата с таймаутами и временными ограничениями в ряде случаев можно адекватно описать при помощи конечного автомата, т.е. с использованием *конечно автоматной абстракции* [10]. В настоящей работе это понятие несколько расширяется для построения конечно автоматной абстракции для временных автоматов с выходными таймаутами. Пусть $S = (S, I, O, h_S, \Delta_S)$ – детерминированный полностью определённый временной автомат и B – наибольшая конечная граница для временных входных интервалов, в то время как D определяет наибольшую выходную задержку. Построим *конечно автоматную абстракцию временного автомата*, которая является полностью определённым конечным автоматом $A_S = (S_A, I \cup \{I\}, O_A, \lambda_{A_S}, s_0)$, где $S_A = \{(s, 0), (s, (0, 1)), \dots, (s, (B-1, B)), (s, B), (s, (B, \infty)) : s \in S\}$, $O_A = \{(o, 0), (o, 1), \dots, (o, D) : o \in O\} \cup \{I\}$, а I – специальный символ конечно автоматной абстракции. Для состояния (s, t_j) , $t_j = 0, \dots, B$, автомата A_S и входного символа i , множество λ_{A_S} содержит переход $((s, t_j), i, (o, d), (s', 0))$, если и только если существует переход $(s, i, o, s', g_i, d) \in \lambda_S$ такой что $t_j \in g_i$. Для состояния (s, g_j) , $g_j = (0, 1) \dots, (B-1, B), (B, \infty)$, автомата A_S и входного символа i , множество λ_{A_S} содержит переход $((s, g_j), i, (o, d), (s', 0))$, если и только если существует переход $(s, i, o, s', g, d) \in \lambda_S$ такой что $g_i \subseteq g$. Переходы по специальному символу I отражают изменение временной переменной между вещественными значениями из интервала вида (a, b) и целочисленными значениями, либо переход по таймаутам между состояниями. Переходы $((s, n), I, I, (s, (n, n+1))), ((s, (n-1, n)), I, I, (s, n)) \in \lambda_{A_S}$, если и только если $n < T < \infty$, где $\Delta_S(s) = (s', T)$. Переход $((s, (n-1, n)), I, I, (s', 0)) \in \lambda_{A_S}$, если и только если $n = T < \infty$, где $\Delta_S(s) = (s', T)$. Подобно [10], можно показать, что если исходный временной автомат полностью определённый и детерминированный, то и конечно автоматная абстракция также является полностью определённым детерминированным конечным автоматом.

Временная входная последовательность α может быть переведена в соответствующую последовательность входных символов для конечно автоматной абстракции α_{FSM} . При этом каждый временной входной символ (i, t) представляется последовательностью входных символов абстракции вида I, I, \dots, I, i , где число символов I равно числу переходов значений временной переменной из целого числа в элемент интервала вида $(a, a+1)$ и обратно за время t . В то же время выходная реакция абстракции на последовательность I, I, \dots, I, i , имеет вид $I, I, \dots, I, (o, d)$, где число символов I во входной и выходной последовательности совпадает, а (o, d) есть реакция исходного временного автомата на (i, t) . Таким образом, выходная последовательность абстракции γ_{FSM} может быть переведена в соответствующую временную выходную последовательность γ удалением всех символов I .

Утверждение 1. Выходно-входная последовательность α/γ существует во временном автомате S , если и только если входно-выходная последовательность $\alpha_{FSM}/\gamma_{FSM}$ существует в конечно автоматной абстракции A_S .

Доказательство. Рассмотрим состояние s временного автомата S и множество состояний $(s, 0), (s, (0, 1)), \dots, (s, (T-1, T))$ конечно автоматной абстракции A_S , где $\Delta_S(s) = (s', T)$. Каждое состояние A_S соответствует целому моменту времени либо интервалу вида $(a, a+1)$, в котором находится временная переменная автомата S , а переходы по входному символу I отражают изменение значения временной переменной автомата S из целого числа в элемент интервала вида $(a, a+1)$ и обратно. Таким образом, по правилам построения множества переходов абстракции, входно-выходная пара $(i, t) / (o, d)$ существует в автомате S , если и только если существует входно-выходная последовательность $I, I, \dots, I, i / I, I, \dots, I, (o, d)$ в автомате A_S , где число символов I равно числу изменений значений временной переменной из целого числа в элемент интервала вида $(a, a+1)$ и обратно за время $t < T$. В силу того, что переходу по таймауту $\Delta_S(s) = (s', T)$ автомата S соответствует переход $((s, (T-1, T)), I, I, (s', 0))$ автомата A_S , то аналогичные рассуждения можно провести для состояния s' и множества $(s', 0), (s', (0, 1)), \dots, (s', (T-1, T))$, а также всех последующих состояний, в которые автомат S может перейти по таймауту и соответствующего множества состояний абстракции. В то же время, по правилам построения переходов конечно автоматной абстракции, входной временной символ (i, t) переведёт автомат S в состояние s'' , если и только если входная последовательность I, I, \dots, I, i переведёт автомат A_S в состояние $(s'', 0)$. Далее, для пары состояний s'' и $(s'', 0)$ можно провести аналогичные рассуждения.

Утверждение 2. Состояния s_1 и s_2 временного автомата S эквивалентны, если и только если в соответствующем конечном автомате A_S эквивалентными являются состояния $(s_1, 0)$ и $(s_2, 0)$.

Доказательство. Пусть состояния s_1 и s_2 временного автомата S эквивалентны. Предположим, что состояния $(s_1, 0)$ и $(s_2, 0)$ соответствующего конечного

автомата не являются эквивалентными, т.е. для них существует различающая последовательность. Тогда в силу утверждения 1, найдётся различающая последовательность и для состояний s_1 и s_2 , что противоречит утверждению, что состояния s_1 и s_2 эквивалентны.

Пусть теперь состояния $(s_1, 0)$ и $(s_2, 0)$ эквивалентны в построенном конечном автомате A_S , однако состояния s_1 и s_2 временного автомата не являются эквивалентными, т.е. для них существует различающая последовательность. Тогда в силу утверждения 1, найдётся различающая последовательность и для состояний $(s_1, 0)$ и $(s_2, 0)$, что противоречит утверждению, что состояния $(s_1, 0)$ и $(s_2, 0)$ эквивалентны.

Таким образом, вывод об эквивалентности состояний временного автомата может быть сделан на основе соответствующих состояний конечно автоматной абстракции.

4. Минимизация числа состояний

Процедура минимизации состояний для классических конечных автоматов хорошо известна [1] и основана на разбиении множества состояний конечного автомата по отношению эквивалентности. Мы предлагаем использовать аналогичный подход для сокращения числа состояний временного автомата. В соответствии с результатами предыдущего раздела, разбиение множества состояний по отношению эквивалентности для временного автомата может быть получено на основе соответствующего разбиения для конечно автоматной абстракции. Далее, мы предлагаем алгоритм построения приведённой по состояниям формы временного автомата.

Алгоритм 1 построения приведённой по состояниям формы полностью определённого детерминированного временного автомата

Вход. Полностью определённый детерминированный временной автомат S

Выход. Приведённая по состояниям форма автомата S

Шаг 1. По исходному временному автомату S строится соответствующая конечно автоматная абстракция A_S , для которой строится разбиение E_{FSM} множества состояний на эквивалентные состояния.

Шаг 2. Строится разбиение E состояний автомата S на эквивалентные состояния следующим образом: состояния s_1 и s_2 автомата S принадлежат одному блоку разбиения E , если и только если состояния $(s_1, 0)$ и $(s_2, 0)$ принадлежат одному блоку разбиения E_{FSM} .

Шаг 3. Строится приведённая по состояниям форма B автомата S . Входной и выходной алфавиты автомата B совпадают с таковыми для автомата S , состояния b_1, b_2, \dots, b_l соответствуют блокам B_1, B_2, \dots, B_l разбиения E . Множество переходов h_B автомата B формируем следующим образом. Выбираем из каждого блока B_i состояние s_i . Для состояний b_i и b_j существует кортеж $(b_i, i, o, b_j, g, d) \in h_B$, тогда и только тогда, когда существует состояние

$s_j \in B_j$, такое что кортеж $(s_i, i, o, s_j, g, d) \in h_s$. Для функции таймаута $\Delta_B(b_i) = (b_j, T)$, тогда и только тогда, когда $\Delta_S(s_i) = (s_j, T)$, $s_j \in B_j$.

Следующее утверждение подтверждает корректность построения приведенной по состояниям формы временного автомата по описанному выше алгоритму.

Утверждение 3. Пусть B – автомат, построенный для автомата S по алгоритму 1. Автомат B является детерминированным полностью определенным автоматом, и состояние b_i автомата B эквивалентно состоянию s автомата S , если и только если $s \in B_i$.

Доказательство. Согласно утверждению 2, состояния s_1 и s_2 временного автомата S принадлежат одному блоку разбиения E , если и только если состояния $(s_1, 0)$ и $(s_2, 0)$ конечно автоматной абстракции A_S принадлежат одному блоку разбиения E_{FSM} . Таким образом, разбиение E , полученное на втором шаге алгоритма, есть разбиение множества состояний автомата S по отношению эквивалентности.

Автомат B является полностью определенным и детерминированным автоматом, поскольку автомат S обладает этими свойствами и для каждого состояния b_i множество переходов «повторяет» множество переходов для состояния $s \in B_i$.

Покажем, что состояния $s_i \in B_i$ и состояние b_i автомата B , соответствующее блоку B_i разбиения E , эквивалентны. Все переходы из состояния b_i дублируют переходы из некоторого состояния $s_i' \in B_i$, эквивалентного любому состоянию s_i из B_i (утверждение 2). В то же время, если $\Delta_S(s_i') = (s_k, T)$ и $\Delta_B(b_i) = (b_k, T)$, $T < \infty$, то $s_k \in B_k$ есть состояние, эквивалентное состоянию s_k' , переходы которого «дублируются» в состоянии b_k . Аналогичные рассуждения можно провести и для возможных преемников по таймаутам состояний s_k и b_k . Таким образом, реакции на любой временной входной символ в состояниях s_i и b_i совпадают. Поскольку переходу $(s_i, i, o, s_i', g_i', d)$ исходного автомата соответствует переход $(b_i, i, o, b_i', g_i', d)$ приведённой по состояниям формы, где $s_i' \in B_i'$, то для состояний s_i' и b_i' , и для всех их преемников по входным временным символам, можно провести аналогичные рассуждения. Таким образом, реакции автоматов B и S в состояниях b_i и s_i на любую входную последовательность совпадают. Покажем теперь, что состояния $s_j \in B_j$ и b_i , где $i \neq j$, не эквивалентны. Состояние $s_i \in B_i$ эквивалентно состоянию b_i согласно первой части доказательства. Состояния s_i и s_j не эквивалентны, так как находятся в различных блоках разбиения E . Таким образом, состояния s_j и b_i не эквивалентны.

Справедлива следующая теорема.

Теорема 1. Пусть S – полностью определённый детерминированный автомат с таймаутами и временными ограничениями. Полностью определённый детерминированный автомат B , построенный по алгоритму 1, является приведённой по состояниям формой автомата S .

В качестве примера рассмотрим автомат на рисунке 1. Приведённая по состояниям форма автомата S представлена на рисунке 2.

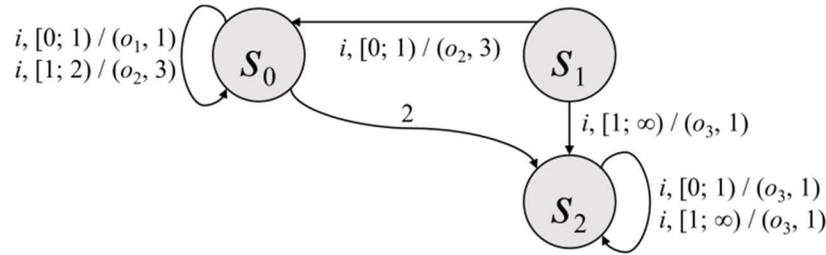


Рис. 2. Приведённая по состояниям форма временного автомата S
 Fig. 2. State reduced form of TFSM S

Сложность алгоритма минимизации временного автомата с n состояниями, вообще говоря, определяется сложностью минимизации конечно автоматной абстракции, число состояний которой не превышает величины $2Bn$ и множество входных символов совпадает с таковым для временного автомата. Таким образом, сложность алгоритма минимизации полиномиальна относительно произведения наибольшей конечной границы для временных входных интервалов и числа состояний.

5. Минимизация временных аспектов

В отличие от классических автоматов, приведенная по состояниям форма не является минимальной формой временного автомата, т.к. не учитывает минимизацию временных параметров. В настоящем разделе мы вводим понятие минимальной формы временного автомата, которая является единственной с точностью до изоморфизма. Автомат с таймаутами и временными ограничениями S называется *приведенным по времени (time reduced)*, если для любых двух кортежей $(s, i, o, s', g_1, d), (s, i, o, s', g_2, d) \in h_s$ справедливо, что g_1 и g_2 нельзя объединить в один интервал и для любого состояния s , такого что $\Delta_S(s) = (s', T)$, не существует состояния s'' и целого числа $T' < T$, таких что автомат S' , полученный изменением функции таймаута состояния s автомата S на $\Delta_S(s) = (s'', T')$ эквивалентен исходному автомату S .

Если для переходов $(s, i, o, s', g_1, d), (s, i, o, s', g_2, d) \in h_s$ временного автомата S интервалы g_1 и g_2 могут быть объединены в один интервал, то эти два перехода можно заменить одним переходом вида $(s, i, o, s', g_1 \cup g_2, d)$. Проверка существования и поиск таймаута меньшей величины могут быть выполнены по конечно автоматной абстракции A_S . Для поиска минимального таймаута в состоянии s необходимо рассмотреть состояния абстракции вида (s, n) , соответствующие поведению временного автомата в состоянии s в целочисленные моменты времени. Если для некоторого состояния (s, j) существует эквивалентное состояние вида $(s', 0)$, то переход по специальному входному символу $((s, (j-1, j)), \mathbf{I}, \mathbf{I}, (s, j))$ может быть заменён на переход $((s, (j-1, j)), \mathbf{I}, \mathbf{I}, (s', 0))$. Соответственно, таймаут в состоянии s автомата S , может

быть заменён на $\Delta_S(s) = (s', j)$. Таким образом, для каждого состояния может быть выбран минимальный возможный таймаут, который не изменяет поведение временного автомата.

Далее, мы предлагаем алгоритм построения приведенной по времени формы временного автомата.

Алгоритм 2 построения приведенной по времени формы автомата с таймаутами и временными ограничениями

Вход. Детерминированный полностью определённый временной автомат S

Выход. Приведенная по времени форма автомата S

Шаг 1. Каждая пара переходов $(s, i, o, s', g_1, d), (s, i, o, s', g_2, d)$, такая что g_1 и g_2 могут быть объединены в один интервал, заменяется на переход $(s, i, o, s', g_1 \cup g_2, d)$.

Шаг 2. По исходному автомату S строится соответствующий классический конечный автомат A_S , для которого строится разбиение по отношению эквивалентности.

Шаг 3. Для каждого состояния s автомата S , для которого $\Delta_S(s) = (s', T)$, среди состояний (s, j) автомата $A_S, j = 1, \dots, T - 1$, определяется состояние с минимальным j , для которого существует эквивалентное состояние вида $(s'', 0)$. Если такое состояние s'' существует, то таймаут в состоянии s заменяется на $\Delta_S(s) = (s'', j)$. Если такого состояния нет, то таймаут в состоянии s не изменяется.

Шаг 4. Для каждого состояния s автомата S , для которого $\Delta_S(s) = (s', \infty)$, определяется состояние вида $(s'', 0)$ в автомате A_S , эквивалентное состоянию $(s, (j, \infty))$. Если такое состояние s'' существует, то таймаут в состоянии s заменяется на $\Delta_S(s) = (s'', j + 1)$. Если такого состояния нет, то таймаут в состоянии s не изменяется.

Шаг 5. Для каждого состояния s , где $\Delta_S(s) = (s', T)$ и $T < \infty$, удаляются все переходы (s, i, o, s', g, d) , такие что $g \cap [0, T) = \emptyset$. Если для перехода $(s, i, o, s', \langle a, b \rangle, d)$ справедливо, что $\langle a, b \rangle \cap [0, T) \neq \emptyset$ и неверно, что $\langle a, b \rangle \subseteq [0, T)$, то переход $(s, i, o, s', \langle a, b \rangle, d)$ заменяется на переход $(s, i, o, s', \langle a, T \rangle, d)$.

Имеет место следующее утверждение.

Утверждение 4. Пусть P – временной автомат, построенный для приведенного по состояниям автомата S по алгоритму 2. Автомат P эквивалентен автомату S и является приведенным по времени автоматом.

Доказательство. Покажем, что автоматы S и P эквивалентны. Объединение переходов $(s, i, o, s', g_1, d), (s, i, o, s', g_2, d)$ в переход $(s, i, o, s', g_1 \cup g_2, d)$ (Шаг 1) не изменяют поведение автомата. Рассмотрим процедуру замены таймаута на третьем шаге алгоритма. Пусть для состояния p временного автомата P найдено состояние временного автомата p'' , такое, что существует состояние (p, j) из A_P , эквивалентное состоянию $(p'', 0)$ автомата A_P . Здесь $\Delta_P(p) = (p', T)$ и $j < T$. Последнее означает, в силу утверждения 2, что реакция автомата P в состоянии

p'' на любую входную последовательность совпадает с таковой для автомата в состоянии p в момент времени j . Таким образом, при замене функции таймаута $\Delta_P(p) = (p', T)$ в состоянии p на $\Delta_P(p) = (p'', j)$ поведение автомата P в состоянии p не меняется. Аналогично, замена таймаута на четвёртом шаге алгоритма не изменяет поведение автомата. На пятом шаге алгоритма удаляются лишь переходы (p, i, o, p', g, d) , временные интервалы которых не пересекаются с интервалом $[0, T]$, где $\Delta_P(p) = (p', T)$, соответственно, такие переходы не могут быть выполнены автоматом. Переходы, временные интервалы которых пересекаются с $[0, T]$, но выходят за границу таймаута, ограничиваются сверху величиной таймаута, что также не изменяет поведение автомата.

Покажем теперь, что автомат P является приведенным по времени автоматом. Допустим, что для некоторого состояния p_0 автомата P таймаут $\Delta_P(p_0) = (p_1, T_1)$ может быть заменён на таймаут $\Delta_P(p_0) = (p_2, T_2)$, где $T_1 > T_2$, таким образом, что поведение автомата в состоянии p_0 не изменится. Последнее означает, что в конечно автоматной абстракции A_P существует либо пара эквивалентных состояний (p_0, T_2) и $(p_2, 0)$, либо $(p_0, (T_2 - 1, \infty))$ и $(p_2, 0)$. Существование эквивалентных состояний (p_0, T_2) и $(p_2, 0)$ невозможно по третьему шагу алгоритма. Существование эквивалентных состояний $(p_0, (T_2 - 1, \infty))$ и $(p_2, 0)$ отражает тот факт, что реакция на любую входную временную последовательность автомата P в состоянии p_0 в момент времени $T > (T_2 - 1)$ совпадает с таковой для состояния $(p_2, 0)$. В таком случае, состояния (p_0, T_2) не существует по построению абстракции, однако, по шагу 4 алгоритма, таймаут был бы заменён на $\Delta_P(p_0) = (p_2, T_2)$. Таким образом, существование эквивалентных состояний $(p_0, (T_2 - 1, \infty))$ и $(p_2, 0)$ невозможно.

В то же время, любые два перехода вида (s, i, o, s', g_1, d) , (s, i, o, s', g_2, d) , которые возможно объединить в один переход $(s, i, o, s', g_1 \cup g_2, d)$, объединяются на шаге 1 алгоритма 2.

На основе полученных результатов, можно установить необходимое и достаточное условие эквивалентности временных автоматов.

Теорема 2. Детерминированные полностью определённые приведенные по состояниям и времени временные автоматы S и P эквивалентны, если и только если эти автоматы изоморфны.

Необходимость. Пусть S и P – детерминированные полностью определённые приведенные по состояниям и времени автоматы с временными ограничениями и таймаутами, которые являются эквивалентными. Покажем, что автоматы S и P изоморфны. Рассмотрим отображение $H: S \rightarrow P$, при котором $H(s)$ есть состояние автомата P , эквивалентное состоянию s . Отображение H является взаимно однозначным в силу приведённости по состояниям автоматов S и P . Покажем, что для любой пары состояний s и $p = H(s)$, такой что $\Delta_S(s) = (s', T_s)$, $\Delta_P(p) = (p', T_p)$, справедливо, что $T_s = T_p$ и $p' = H(s')$. Допустим, что $T_p < T_s$; тогда в силу эквивалентности S и P , существует состояние s'' , эквивалентное состоянию p' . А поскольку состояния s и p также являются эквивалентными, то

таймаут в состоянии s может быть заменён на $\Delta_S(s) = (s'', T_s')$, где $T_s' = T_p < T_s$. Последнее невозможно, поскольку S является приведенным по времени автоматом. Пусть теперь $T_p > T_s$, тогда в силу эквивалентности S и P , существует состояние p'' , эквивалентное состоянию s' . А поскольку состояния s и p также являются эквивалентными, то таймаут в состоянии p может быть заменён на $\Delta_P(p) = (p'', T_p')$, где $T_s = T_p' < T_p$. Последнее невозможно, поскольку P есть приведенный по времени автомат. В силу равенства таймаутов $T_p = T_s$ и эквивалентности состояний s и p , справедливо, что функции $time(s, T_s)$ и $time(p, T_p)$ совпадают, и, соответственно, $p' = H(s')$. Аналогично [11], ввиду эквивалентности S и P , справедливо, что для любого перехода (s, i, o, s', g, d) существует переход $(H(s), i, o, H(s'), g, d)$. Таким образом, автоматы S и P изоморфны.

Достаточность. Поскольку изоморфные автоматы совпадают с точностью до переименования состояний, то изоморфные автоматы эквивалентны.

Для приведённой по состояниям формы автомата S (рисунок 2) приведённая по времени форма показана на рисунке 3.

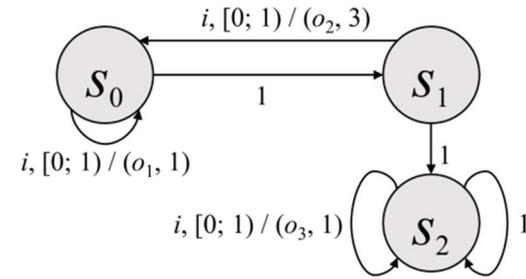


Рис. 3. Приведённая по времени и состояниям форма автомата S
Fig. 3. Time and state reduced form of reduced form of S

Как видно из рисунка 3, таймаут $\Delta_S(s_0) = (s_2, 2)$ был заменён на таймаут $\Delta_S(s_0) = (s_1, 1)$, а переход $(s_0, i, o_2, s_0, [1, 2), 3)$ удалён. В то же время, два перехода из состояния s_2 были объединены, а таймауты $\Delta_S(s_1) = (s_1, \infty)$ и $\Delta_S(s_2) = (s_2, \infty)$ заменены на $\Delta_S(s_1) = (s_2, 1)$ и $\Delta_S(s_2) = (s_2, 1)$ соответственно.

Таким образом, для временных автоматов необходимо ввести определение минимальной формы, учитывающей не только минимизацию состояний, но и минимизацию временных аспектов. *Минимальной формой* полностью определённого детерминированного автомата с таймаутами и временными ограничениями S называется приведенный по состояниям и времени автомат, эквивалентный S . По определению минимальной формы, справедлива следующая теорема.

Теорема 3. Минимальная форма автомата с таймаутами и временными ограничениями единственна с точностью до изоморфизма.

7. Заключение

В настоящей работе предложен метод построения минимальной формы автомата с таймаутами и временными ограничениями. Предложенный подход основан на построении разбиения множества состояний временного автомата на классы эквивалентности, которое строится на основе соответствующего разбиения для конечно автоматной абстракции. В отличие от классических конечных автоматов, предложенный подход минимизирует не только число состояний, но и временные аспекты. Мы также показываем, что минимальная форма временного автомата единственна с точностью до изоморфизма. Тем не менее, можно показать, что свойство минимальности временного автомата не сохраняется для его конечно автоматной абстракции. В конечно автоматной абстракции минимального временного автомата возможно наличие эквивалентных состояний. Таким образом, представляет интерес исследование свойств конечно автоматной абстракции временных автоматов, для которых свойство минимальности будет инвариантом.

Отметим также, что единственная минимальная форма для автоматов только с таймаутами и только с временными ограничениями может быть построена по алгоритму 2 с необходимыми сокращениями.

Представляет интерес изучение минимальной формы для инициальных автоматов с таймаутами, т.е. автоматов, в которых выделено начальное состояние. В этом случае в исходном автомате возможно наличие состояний, которые достижимы из начального состояния только по таймаутам и соответственно, не обязательно имеют эквивалентное состояние в минимальной форме. В качестве дальнейших исследований мы также предполагаем провести эксперименты по минимизации временных автоматов, описывающих поведение реальных управляющих систем.

Благодарности

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

Список литературы

- [1]. Gill A. Introduction to the Theory of Finite-State Machines. 1962, 207 p.
- [2]. Lee, D., Yannakakis, M. (1996) Principles and methods of testing finite state machines—a survey. *Proceedings of the IEEE*. 1996, 84(8), pp. 1090-1123.
- [3]. Murphy T.E., Geng X.-J., Hammer J.. On the control of asynchronous machines with races. *IEEE Transactions on Automatic Control*, 2003, 48(6), pp. 1073-1081.
- [4]. Kumar R., Garg V.K. Modeling and control of logical discrete event systems. Kluwer Academic Publishers, 1995, 143 p.
- [5]. Cassandras C. C., Lafortune S.. Introduction to discrete event systems. Kluwer Academic Publishers, 1999, 822 p.
- [6]. 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*, 2010, 52, pp. 1286-1297.

- [7]. Zhigulin M., Yevtushenko N, Maag S., Cavalli A. FSM-Based Test Derivation Strategies for Systems with Time-Outs. *Proc. of the 11th International Conference on Quality Software, QSIQ 2011, IEEE*, 2011. pp. 141-149.
- [8]. El-Fakih K., Yevtushenko N., Fouchal H. Testing Timed Finite State Machines with Guaranteed Fault Coverage. *TestCom/FATES*, 2009, pp. 66-80.
- [9]. Merayo M., Nunez M., Rodriguez I., Formal testing from timed finite state machines, *Comput. Netw*, 2008, 52 (2), 432-460.
- [10]. Bresolin D., El-Fakih K., Villa T., Yevtushenko N. Deterministic Timed Finite State Machines: Equivalence Checking and Expressive Power. *Intern Conf. GANDALF*, 2014, pp. 203-216.
- [11]. Твардовский А. С., Евтушенко Н. В. К минимизации автоматов с временными ограничениями. *Вестник ТГУ. Управление, вычислительная техника и информатика*, № 4 (29), 2014 г., стр. 77-83.
- [12]. Твардовский А. С. К минимизации автоматов с таймаутами. *Труды ИСП РАН*, т. 26, вып. 6, 2014 г., стр. 77-84. DOI: 10.15514/ISPRAS-2014-26(6)-7

Minimizing Finite State Machines with time guards and timeouts

A.S. Tvardovskii <tvardal@mail.ru>

N.V. Yevtushenko <nyevtush@gmail.com>

M.L. Gromov <maxim.leo.gromov@gmail.com>

National Research Tomsk State University

Lenin st. 36, Tomsk, 634050, Russia

Abstract. Finite State Machines (FSMs) are widely used for analysis and synthesis of components of control systems. In order to take into account time aspects, timed FSMs are considered. As the complexity of many problems of analysis and synthesis of digital and hybrid systems including high-quality test derivation significantly depends on the size of the system (component) specification, in this paper, we address the problem of minimizing a FSM with timed guards and input and output timeouts (TFSM). The behavior of a TFSM can be described using a corresponding FSM abstraction and a proposed approach for minimizing a TFSM is based on such abstraction. We minimize not only the number of states as it is done for classical FSMs but also the number of timed guards and timeout duration. We show that for a complete deterministic TFSM there exists the unique minimal (canonical) form, i.e., a unique time and state reduced TFSM that has the same behavior as the given TFSM; for example, this minimal form can be used when deriving tests for checking whether an implementation under test satisfies functional and nonfunctional requirements. A proposed approach for minimizing timed machines can be applied to particular cases of TFSM, i.e. for FSM with timeouts and FSM with timed guards.

Keywords: Timed Finite State Machines; reduced form; minimal form.

DOI: 10.15514/ISPRAS-2017-29(4)-8

For citation: Tvardovskii A.S., Yevtushenko N.V., Gromov M.L. Minimizing Finite State Machines with time guards and timeouts. *Trudy ISP RAN/Proc. ISP RAS*, vol. 29, issue 4, 2017 г., pp. 139-154 (in Russian). DOI: 10.15514/ISPRAS-2017-29(4)-8

References

- [1]. Gill A. Introduction to the Theory of Finite-State Machines. 1962, 207 p.
- [2]. Lee, D., Yannakakis, M. (1996) Principles and methods of testing finite state machines-a survey. *Proceedings of the IEEE*. 1996, 84(8), pp. 1090-1123.
- [3]. Murphy T.E., Geng X.-J., Hammer J.. On the control of asynchronous machines with races. *IEEE Transactions on Automatic Control*, 2003, 48(6), pp. 1073-1081.
- [4]. Kumar R., Garg V.K. Modeling and control of logical discrete event systems. Kluwer Academic Publishers, 1995, 143 p.
- [5]. Cassandras C. C., Lafortune S.. Introduction to discrete event systems. Kluwer Academic Publishers, 1999, 822 p.
- [6]. Dorofeeva R., El-Fakih K., Maag S., Cavalli A., Yevtushenko N. FSM-based conformance testing methods: A survey an-notated with experimental evaluation // *Information and Software Technology*, 2010, 52, pp. 1286-1297.

- [7]. Zhigulin M., Yevtushenko N, Maag S., Cavalli A. FSM-Based Test Derivation Strategies for Systems with Time-Outs. *Proc. of the 11th International Conference on Quality Software, QSIC 2011, IEEE*, 2011. pp. 141-149.
- [8]. El-Fakih K., Yevtushenko N., Fouchal H. Testing Timed Finite State Machines with Guaranteed Fault Coverage. *TestCom/FATES*, 2009, pp. 66-80.
- [9]. Merayo M., Nunez M., Rodriguez I., Formal testing from timed finite state machines, *Comput. Netw.*, 2008, 52 (2), 432-460.
- [10]. Bresolin D., El-Fakih K., Villa T., Yevtushenko N. Deterministic Timed Finite State Machines: Equivalence Checking and Expressive Power. *Intern Conf. GANDALF*, 2014, pp. 203-216.
- [11]. Tvardovskii A., Yevtushenko N. Minimizing timed Finite State Machines. *Tomsk State University Journal of Control and Computer Science*, 2014, № 4 (29), pp. 77-83 (in Russian).
- [12]. Tvardovskii A. On the minimization of timed Finite State Machines. *Trudy ISP RAN/Proc. ISP RAS*, vol. 26, issue 6, 2014, pp. 77-84 (in Russian). DOI: 10.15514/ISPRAS-2014-26(6)-7