

УДК 519.713.1; 519.718.7

Н.В. Шабалдина

ИССЛЕДОВАНИЕ ОТНОШЕНИЯ НЕРАЗДЕЛИМОСТИ ДЛЯ НЕДЕТЕРМИНИРОВАННЫХ АВТОМАТОВ С ТАЙМ-АУТАМИ

В статье исследуется отношение неразделимости для недетерминированных автоматов, в которых каждое состояние может обладать целочисленной задержкой (тайм-аут). Если тайм-аут заканчивается и на автомат не было подано ни одного входного символа, то автомат изменяет свое состояние согласно переходу по этой временной задержке. Предлагается алгоритм построения разделяющей последовательности для таких автоматов, т.е. такой временной входной последовательности, на которую множества выходных последовательностей автоматов не пересекаются, и соответственно достаточно подать разделяющую последовательность на эти автоматы один раз, чтобы по выходной реакции различить их.

Ключевые слова: *недетерминированные автоматы с тайм-аутами, временная входная последовательность, отношение неразделимости, разделяющая последовательность.*

В статье рассматривается одна из модификаций классической модели конечно-го автомата [1] – модель с тайм-аутами [2 – 4], которую в рамках данной работы мы будем называть временным автоматом. В этой модели, как и в модели конечного автомата, есть состояния, входные символы, выходные символы; переходы из состояния в состояние могут осуществляться при подаче входного символа и сопровождаются выдачей выходного символа и, кроме того, возможны переходы по временным задержкам (тайм-аутам), т.е. по прошествии некоторого времени, если в течение этого времени на автомат не поступил ни один входной символ. При переходе по тайм-ауту на автомат не поступают внешние воздействия, и эта дополнительная возможность позволяет более полно описать поведение многих современных технических систем. В качестве примера можно привести сотовый телефон, у которого по истечению некоторого времени гаснет экран, mp3-плеер и другие системы.

Для временных автоматов только начаты исследования различных отношений конформности [3]. Эти отношения и их свойства во многом заимствуются из теории конечных автоматов. Отношение неразделимости [5, 6], исследуемое в данной статье, также было введено изначально для конечных автоматов, а именно, для автоматов с недетерминированным поведением. Данное отношение замечательно тем, что при тестировании не требуется выполнения «всех погодных условий», т.е. необходимости подавать каждую тестовую последовательность до тех пор и при таких условиях, чтобы пронаблюдать все выходные последовательности системы на поданную входную последовательность. На практике затруднительно обеспечить выполнение такого предположения, и поэтому построение тестов с использованием отношения неразделимости актуально для моделей недетерминированных конечных автоматов [5, 6] и их модификаций [3, 4]. В данной работе предлагается алгоритм построения разделяющей последовательности для временных автоматов с тайм-аутами, на основе которого можно построить пол-

ный проверяющий тест относительно неразделимости при явном перечислении всех неисправностей.

1. Основные определения и обозначения

В данной работе *временным автоматом* (автоматом с тайм-аутами) называется шестерка $A = (S, I, O, s_0, \lambda_A, \Delta_A)$, где S – конечное непустое множество состояний с выделенным начальным состоянием s_0 , I и O – конечные непересекающиеся входной и выходной алфавиты, $\lambda_A \subseteq S \times I \times O \times S$ – *отношение переходов*. Функция $\Delta_A: S \rightarrow S \times (\mathbb{N} \cup \{\infty\})$, где $\mathbb{N}$ – множество натуральных чисел, есть *функция задержки*, определяющая для каждого состояния тайм-аут, т.е. время ожидания входного символа без изменения состояния, причём если для некоторого состояния s имеет место $\Delta_A(s) = (s', \infty)$, то $s = s'$. Таким образом, классический конечный автомат [1], в том числе недетерминированный [5, 6], можно рассматривать как временной автомат, в котором для всех состояний $s \in S$ имеет место $\Delta_A(s) = (s, \infty)$, т.е. конечный автомат, в отличие от временного, сколько угодно долго ожидает входного символа, не изменяя при этом текущее состояние.

Временному автомату ставится в соответствие внутренняя переменная (часы), принимающая целые неотрицательные значения и указывающая время, прошедшее с момента достижения текущего состояния [2 – 4] (или с момента выдачи последнего выходного символа). Если $(s, i, s', o) \in \lambda_A$, то автомат A в состоянии s на входной символ i может выдать выходной символ o и изменить своё состояние на s' , причём после этого соответствующая временная переменная принимает значение 0. В данной работе мы не рассматриваем задержки выходных символов, т.е. полагаем, что временной автомат выдаёт выходной символ в тот же самый момент, когда принимает входной символ i , возможно, меняет своё состояние.

Если для каждой пары $(s, i) \in S \times I$ существует не более одной пары $(o, s') \in O \times S$, такой, что $(s, i, o, s') \in \lambda_A$, то временной автомат A называется *детерминированным*; иначе временной автомат называется *недетерминированным*. Если для каждой пары $(s, i) \in S \times I$ существует, по крайней мере, одна пара $(o, s') \in O \times S$, такая, что $(s, i, o, s') \in \lambda_A$, то S называется *полностью определённым*. Временной автомат A называется *наблюдаемым*, если для любой тройки $(s, i, o) \in S \times I \times O$ существует не более одного состояния, такого, что $(s, i, o, s') \in \lambda_A$.

Временной входной символ есть пара $\langle i, t \rangle \in I \times Z_0^+$, где Z_0^+ – множество целых неотрицательных чисел; временной входной символ $\langle i, t \rangle$ показывает, что входной символ i подается в момент, когда значение временной переменной равно t . Последовательность $\langle i_1, t_1 \rangle \langle i_2, t_2 \rangle \dots \langle i_l, t_l \rangle$ временных входных символов называется *временной входной последовательностью* длины l .

Для того чтобы расширить отношение переходов временного автомата на временные входные символы, введем специальную функцию $time_A: S \times Z_0^+ \rightarrow S$, позволяющую определить по текущему состоянию автомата состояние автомата в момент времени, когда значение временной переменной равно t [4]. Для определения значения функции $time_A(s, t)$ рассмотрим последовательность тайм-аутов $\Delta_A(s) = (s_1, T_1)$, $\Delta_A(s_1) = (s_2, T_2)$, ..., $\Delta_A(s_{p-1}) = (s_p, T_p)$, такую, что $T_1 + T_2 + \dots + T_{p-1} \leq t$, но $T_1 + T_2 + \dots + T_p > t$. В этом случае $time_A(s, t) = s_{p-1}$. Если $\Delta_A(s) = (s, \infty)$, то $time_A(s, t) = s$ для каждого значения переменной t . Для каждого временного вход-

ного символа $\langle i, t \rangle$ к отношению λ_A добавляется переход $(s, \langle i, t \rangle, s', o)$, если и только если $(time_A(s, t), i, s', o) \in \lambda_A$.

Множество всех входных временных последовательностей обозначается I_t^* . Для полностью определенного временного автомата A функция $outs$, отображающая множество $S \times I_t^*$ во множество выходных последовательностей, т.е. $out_A(s, \alpha): S \times I_t^* \rightarrow O^*$, вводится следующим образом. Для состояния s и временной входной последовательности $\alpha = \langle i_1, t_1 \rangle \dots \langle i_l, t_l \rangle$ последовательность $o_1 \dots o_l \in outs(s, \alpha)$, если существуют состояния $s_1 = s, \dots, s_{l+1}$, такие, что для каждого $j \in \{1, \dots, l\}$ временной автомат A имеет временной переход $(s_j, \langle i_j, t_j \rangle, o_j, s_{j+1})$, и будем говорить в этом случае, что пара $(\alpha, out_A(s, \alpha))$ может *перевести* автомат A из состояния s в состояние s_{l+1} . Пара «временная_входная_последовательность α / выходная_последовательность β » называется *временной_входо-выходной* последовательностью или *временной трассой* временного автомата A в состоянии s , если $\beta \in out_A(s, \alpha)$. Можно сказать, что временные трассы характеризуют *функциональное* поведение временного автомата (относительно временных входных последовательностей).¹ Множество всех функциональных трасс в начальном состоянии автомата A обозначим TTr_A .

Состояние s' называется $\langle i, t \rangle$ -*преемником* состояния s , если $(s, \langle i, t \rangle, s', o) \in \lambda_A$. Состояние s в этом случае называется $\langle i, t \rangle$ -*предшественником* состояния s' . Соответственно любой $\langle i, t \rangle$ -преемник состояния s ($\langle i, t \rangle$ -предшественник состояния s') можно рассматривать как просто $\langle i, t \rangle$ -преемник состояния s ($\langle i, t \rangle$ -предшественник состояния s'). Множество всех $\langle i, t \rangle$ -преемников состояния s будем обозначать $suc_A(s, \langle i, t \rangle, o)$, множество всех $\langle i, t \rangle$ -преемников состояния s будем обозначать $suc_A(s, \langle i, t \rangle)$, и в обоих случаях для краткости символ $\langle i, 0 \rangle$ будем заменять символом i .

Если временной автомат A детерминированный, то для каждого состояния s и каждой временной входной последовательности α множество $out_A(s, \alpha)$ содержит не более одного элемента. Для полностью определенного временного автомата A множество $out_A(s, \alpha)$ не является пустым для любых $s \in S, \alpha \in I_t^*$.

Пусть $A = (S, I, O, s_0, \lambda_A, \Delta_A)$ и $B = (P, I, O, p_0, \lambda_B, \Delta_B)$ — полностью определенные автоматы с тайм-аутами. Автомат B называется *редукцией* автомата A , если $TTr_B \subseteq TTr_A$; если $TTr_B \not\subseteq TTr_A$, то автомат B не является редукцией автомата A . Временные автоматы A и B называются *неразделимыми*, если выходные реакции автоматов на любую временную входную последовательность α пересекаются, т.е. $out_A(s_0, \alpha) \cap out_B(p_0, \alpha) \neq \emptyset$; в противном случае автоматы называются *разделимыми*. Временная входная последовательность α , такая, что $out_A(s_0, \alpha) \cap out_B(p_0, \alpha) = \emptyset$, называется *разделяющей* последовательностью для временных автоматов A и B .

Пересечением $A \cap B$ называется наибольший связный подавтомат $C = (Q, I, O, q_0, \lambda_C, \Delta_C)$ автомата $(R, I, O, q_0, \lambda_C, \Delta_C)$, в котором $R = S \times K \times P \times K$, где $K = \{0, \dots, k\}$, $k = \min(\max \Delta_A(s), \max \Delta_B(p))$, начальное состояние есть четверка $(s_0, 0, p_0, 0)$ и отношение переходов λ_C и функция Δ_C определены по следующим правилам [3]:

1. Отношение переходов λ_C содержит четверку $[(s, k_1, p, k_2), i, o, (s', 0, p', 0)]$, если и только если $(s, i, o, s') \in \lambda_A$ и $(p, i, o, p') \in \lambda_B$.

¹ В данной статье при определении временных трасс мы опускаем слово «функциональная», поскольку не рассматриваем задержки выходных символов.

2. $\Delta_C((s, k_1, p, k_2)) = [(s', k_1', p', k_2'), k]$, где $k = \min(\Delta_A(s)_{\downarrow N} - k_1, \Delta_B(p)_{\downarrow N} - k_2)$.
 Состояние $(s', k_1', p', k_2') = (\Delta_A(s)_{\downarrow S}, 0, \Delta_B(p)_{\downarrow P}, 0)$, если $(\Delta_A(s)_{\downarrow N} = \infty$ или $\Delta_B(p)_{\downarrow N} = \infty$, или $(\Delta_A(s)_{\downarrow N} - k_1) = (\Delta_B(p)_{\downarrow N} - k_2)$). Если $(\Delta_A(s)_{\downarrow N} - k_1), (\Delta_B(p)_{\downarrow N} - k_2) \in \mathbb{Z}_+$ и $(\Delta_A(s)_{\downarrow N} - k_1) < (\Delta_B(p)_{\downarrow N} - k_2)$, то состояние $(s', k_1', p', k_2') = (\Delta_A(s)_{\downarrow S}, 0, p, k_2 + k)$.
 Если $(\Delta_A(s)_{\downarrow N} - k_1), (\Delta_B(p)_{\downarrow N} - k_2) \in \mathbb{Z}_+$ и $(\Delta_A(s)_{\downarrow N} - k_1) > (\Delta_B(p)_{\downarrow N} - k_2)$, то состояние $(s', k_1', p', k_2') = (s, k_1 + k, \Delta_B(p)_{\downarrow P}, 0)$.

Алгоритм 1

Построение пересечения двух полностью определенных временных автоматов

Вход: Временные автоматы $A = (S, I, O, s_0, \lambda_A, \Delta_A)$ и $B = (P, I, O, p_0, \lambda_B, \Delta_B)$.

Выход: Пересечение $C = (Q, I, O, q_0, \lambda_C, \Delta_C)$.

- 1: Добавить в первоначально пустое множество Q состояние $q_0 = (s_0, 0, p_0, 0)$;
- 2: **Пока** во множестве Q есть нерассмотренные состояния;
- 3: **Для** каждого нерассмотренного состояния $q = (s, k_1, p, k_2)$ из множества Q ;
- 4: **Для** каждой пары i/o определить множество состояний Q' и пополнить множество переходов λ_C , добавляя состояние $q' = (s', 0, p', 0)$ в Q' и переход $[(s, k_1, p, k_2), i, o, (s', 0, p', 0)]$ в $\lambda_C \Leftrightarrow (s, i, s', o) \in \lambda_A$ и $(p, i, p', o) \in \lambda_B$;
- 5: $Q := Q \cup Q'$;
- 6: **Если** $(\Delta_A(s)_{\downarrow (\mathbb{N} \cup \{\infty\})} \neq \infty$ или $\Delta_B(p)_{\downarrow (\mathbb{N} \cup \{\infty\})} \neq \infty$), **то** выполнять:
 $k := \min(\Delta_A(s)_{\downarrow N} - k_1, \Delta_B(p)_{\downarrow N} - k_2)$;
- 7: **Если** $(\Delta_A(s)_{\downarrow (\mathbb{N} \cup \{\infty\})} = \infty$ или $\Delta_B(p)_{\downarrow (\mathbb{N} \cup \{\infty\})} = \infty$ или $(\Delta_A(s)_{\downarrow N} - k_1) = (\Delta_B(p)_{\downarrow N} - k_2)$,
- 8: **то** $q' := (\Delta_A(s), 0, \Delta_B(p), 0)$;
- 9: **Иначе**
- 10: **если** $(\Delta_A(s)_{\downarrow N} - k_1) < (\Delta_B(p)_{\downarrow N} - k_2)$,
- 11: **то** $q' := (\Delta_A(s)_{\downarrow S}, 0, p, k_2 + k)$;
- 12: **если** $(\Delta_A(s)_{\downarrow N} - k_1) > (\Delta_B(p)_{\downarrow N} - k_2)$,
- 13: **то** $q' := (s, k_1 + k, \Delta_B(p)_{\downarrow P}, 0)$;
- 14: Доопределить функцию $\Delta_C: \Delta_C(q) = (q', k)$; $Q := Q \cup \{q'\}$;
- 15: Автомат $C = (Q, I, O, q_0, \lambda_C, \Delta_C)$ – искомый автомат $A \cap B$; **Конеч.**

На рис. 1 приведены два временных автомата A и B , на рис. 2 – автомат пересечения $A \cap B$.

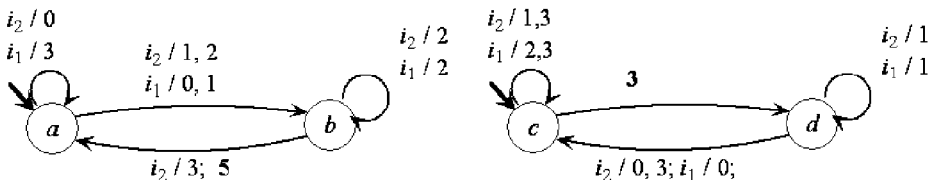
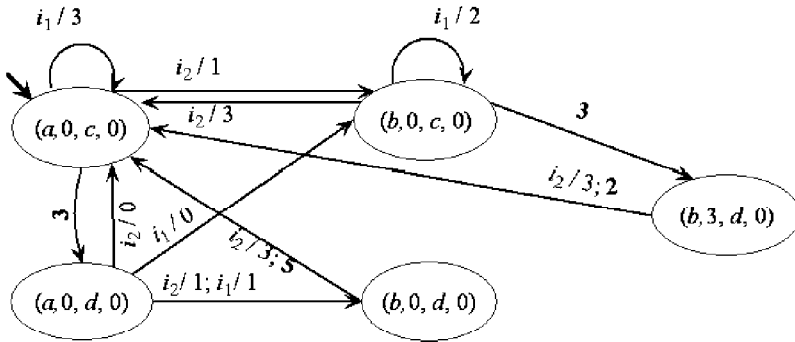


Рис. 1. Временные автоматы A и B

Рис. 2. Автомат $A \cap B$

2. Построение разделяющей последовательности для полностью определенных временных автоматов

Разделяющая последовательность для временных автоматов A и B строится на основе дерева преемников пересечения $A \cap B$. Если автоматы A и B не являются разделимыми, то на выходе алгоритма выдается соответствующее сообщение.

Алгоритм 2

Построение разделяющей последовательности для двух полностью определенных временных автоматов

Вход: Полностью определенные временные автоматы

$$A = (S, I, O, s_0, \lambda_A, \Delta_A) \text{ и } B = (P, I, O, p_0, \lambda_B, \Delta_B).$$

Выход: Разделяющая последовательность для автоматов A и B (если автоматы A и B разделимы) или сообщение, что автоматы A и B неразделимы.

- 1: Построить пересечение $C = A \cap B$;
- 2: Если C – полностью определенный автомат, то
выдать сообщение «автоматы A и B неразделимы»; **Конец.**
- 3: $k := 0$; $Edge := \emptyset$; $Q_{k0} := \{\langle q_0, 0 \rangle\}$; $Q_k := \{Q_{k0}\}$;
- 4: Пока не выполнится одно из следующих правил усеечения дерева преемников:
- 5: **Правило 1:** для некоторого множества $Q_{kj} \in Q_k, j \geq 0$, существует входной символ i - ser такой, что для каждого состояния $q, \langle q, t \rangle \in Q_{kj}$, переход из q по входному символу i - ser не определен;
- 6: Или
- 7: **Правило 2:** для каждого множества $Q_{kj} \in Q_k$ существует $Q_{am} \in Q_a, a < k$, такое, что $Q_{kj} \supseteq Q_{am}$;
- 8: **выполнять:**
- 9: $Q_{k+1} := \emptyset$;
- 10: Для каждого подмножества $Q_{kj} \in Q_k, j = 0, \dots, |Q_k|-1$, для которого не существует $Q_{am} \in Q_a, a < k$, такого, что $Q_{kj} \supseteq Q_{am}$, выполнять:
- 11: Для каждого входного символа i
- 12: Построить множество $M := \bigcup_{q \in Q_{kj}} suc_C(q \downarrow_Q, i) \times \{0\}$;
- 13: $Q_{k+1} = Q_{k+1} \cup M$; добавить тройку (Q_{kj}, i, M) во множество $Edge$;

- 14: Если не для всех $q, \langle q, t \rangle \in Q_{kj}, (\Delta_C(q)) \downarrow (\mathbb{N} \cup \{\infty\}) = \infty$, то определить для множества $Q_{kj} = \{\langle q_1, t_1 \rangle, \langle q_2, t_2 \rangle, \dots, \langle q_r, t_r \rangle\}$ минимальную задержку T и множество преемников R по этой задержке следующим образом:
- 15: $T := \min_{1 \leq u \leq r} \{T_u - t_u\}$, где $T_u = (\Delta_C(q_u)) \downarrow (\mathbb{N} \cup \{\infty\})$;
- 16: $R := \{\langle q_1', t_1' \rangle, \langle q_2', t_2' \rangle, \dots, \langle q_r', t_r' \rangle\}$,
где для всех $u \in \{1, \dots, r\}$
 $q_u' = \text{time}_C(q_u, t_u + T)$;
Если $(T_u = \infty \text{ или } T_u = t_u + T)$, то $t_u' = 0$;
Если $(T_u > t_u + T)$, то $t_u' = t_u + T$;
- 17: $Q_{k+1} = Q_{k+1} \cup R$; добавить тройку (Q_{kj}, T, R) во множество $Edge$;
- 18: $k := k + 1$;
- 19: Если выполнилось **Правило 2** усечения дерева преемников, то выдать сообщение «автоматы A и B неразделимы», **Конец**.
- 20: Построить последовательность ребер $(Q_{00}, g_1, Q_{1j_1}), (Q_{1j_1}, g_2, Q_{2j_2}), \dots, (Q_{(k-1)j_{k-1}}, g_k, Q_{kj_k})$, такую, что для множества Q_{kj_k} выполнилось **Правило 1** усечения дерева преемников, $(Q_{(l-1)j_{l-1}}, g_l, Q_{lj_l}) \in Edge$ для каждого $l \in \{1, \dots, k\}$, и $g_l \in \{I \cup N\}$.
- 21: По построенной последовательности ребер выписать последовательность $g_1 g_2 \dots g_k$ из входных символов и временных задержек.
- 22: Построить временную входную последовательность $\alpha = \langle i_1, t_1 \rangle \dots \langle i_m, t_m \rangle$ следующим образом:
- 23: $j := 0; T_j := 0; r := 0$;
- 24: Пока $(j \leq k)$
- 25: Если $g_j \in I$, то $i_r := g_j, t_r := T_j, r := r + 1, T_j := 0$;
- 26: иначе $T_j := T_j + g_j$;
- 27: $j := j + 1$;
- 28: $m := r; i_r := i\text{-sep}, t_r := T_j$;
- 29: Последовательность $\alpha = \langle i_1, t_1 \rangle \dots \langle i_m, t_m \rangle$ – искомая разделяющей последовательностью для автоматов A и B , **Конец**.

Теорема. Если алгоритм 2 возвращает входную последовательность α , то α есть разделяющая последовательность для автоматов A и B . Если алгоритм 2 выдает сообщение «автоматы A и B неразделимы», то автоматы A и B не являются разделимыми.

Доказательство. Алгоритм 2 является модификацией алгоритма построения разделяющей последовательности для классических конечных автоматов, который был предложен в работе [6]. Внесенные изменения связаны с особенностью модели временного автомата с задержками, а именно, с переходами по временным задержкам, в связи с чем некоторые состояния временного автомата могут быть достигнуты только по таким переходам. Поэтому в данном алгоритме вершинам дерева соответствуют не подмножества состояний автомата-пересечения, а подмножества пар <состояние, время>.

При переходе по обычным входным символам значение времени в преемнике <состояние, время> обнуляется. При переходе по временным задержкам временное значение накапливается, пока не будет достигнуто или превышено значение задержки, по которой в автомате-пересечении есть переход из данного состояния. Таким образом, в дереве, которое строится в алгоритме, добавляются дуги по минимальным временным задержкам (строки 14 – 17 алгоритма). Можно видеть, что для множества Q_{kj} пар <состояние, время> текущей вершины опреде-

ляется минимальная временная задержка. Вычисление множества преемников по этой задержке осуществляется так же, как при построении пересечения автоматов. Внесение данного дополнения в алгоритм связано с тем, что пара временных автоматов может быть разделима по временной входной последовательности, т.е. прежде, чем подать входной символ, необходимо выждать некоторое время, для того, чтобы автомат-пересечение достиг нужного подмножества состояний. Причем дуге, помеченной входным символом, может предшествовать последовательность из нескольких дуг, помеченных временными задержками.

Согласно алгоритму, если для некоторой вершины, помеченной множеством Q_{kj_k} , выполнилось Правило 2 усечения дерева преемников, то это означает, что нет нераскрытых вершин, не объявленных листьями, и множество пар <состояние, время> в каждой усеченной вершине дерева содержат в себе множество пар некоторой вершины на одном из предшествующих ярусов дерева, а значит, из такой вершины не может быть построена ветвь, соответствующая разделяющей последовательности.

Если же для некоторой вершины, помеченной множеством Q_{kj_k} , выполняется Правило 1 усечения дерева преемников, то в строках 22 – 28 последовательность α собирается по ветви дерева, ведущей из корня в данную вершину. При этом временной входной символ формируется в результате суммирования всех задержек до очередного входного символа. Поскольку Правило 1 усечения дерева преемников перешло из алгоритма работы [6] без изменений, а переходы по временным задержкам из множеств пар <состояние, время> формируются согласно правилам, регламентированным при определении совместного поведения временных систем (а именно, при построении пересечения), то последовательность α будет разделяющей последовательностью для автоматов A и B . ■

Пример. Для иллюстрации алгоритма 2 рассмотрим автоматы A и B на рис. 1. Автомат $A \cap B$ показан на рис. 2. По определению, множество $Q_0 = \{Q_{00}\}$, где $Q_{00} = \{ \langle (a, 0, c, 0), 0 \rangle \}$. При $k = 0$ мы получаем $Q_1 = \{Q_{10} = \{ \langle (a, 0, c, 0), 0 \rangle \}, Q_{11} = \{ \langle (b, 0, c, 0), 0 \rangle \}, Q_{12} = \{ \langle (a, 0, d, 0), 0 \rangle \} \}$, множество $Edge = \{ (Q_{00}, i_1, Q_{10}), (Q_{00}, i_2, Q_{11}), (Q_{00}, 3, Q_{12}) \}$. При $k = 1$ мы получаем $Q_2 = \{Q_{20} = \{ \langle (a, 0, c, 0), 0 \rangle \}, Q_{21} = \{ \langle (a, 0, c, 0), 0 \rangle \}, Q_{22} = \{ \langle (b, 3, d, 0), 0 \rangle \}, Q_{23} = \{ \langle (b, 0, c, 0), 0 \rangle \}, \langle (b, 0, d, 0), 0 \rangle \}$, $Q_{24} = \{ \langle (a, 0, c, 0), 0 \rangle, \langle (b, 0, d, 0), 0 \rangle \}$, множество $Edge = \{ (Q_{00}, i_1, Q_{10}), (Q_{00}, i_2, Q_{11}), (Q_{00}, 3, Q_{12}), (Q_{11}, i_1, Q_{20}), (Q_{11}, i_2, Q_{21}), (Q_{11}, 3, Q_{22}), (Q_{12}, i_1, Q_{23}), (Q_{12}, i_2, Q_{24}) \}$.

Можно видеть, что поведение автомата $A \cap B$ не определено в каждом состоянии множества Q_{22} по входному символу $i-sep = i_1$. Таким образом, временная входная последовательность $\langle i_2, 0 \rangle \langle i_1, 3 \rangle$ разделяет автоматы A и B .

В связи с тем, что при существовании разделяющей последовательности построение дерева прекращается, как только допустимо усечение по Правилу 1, последовательность (если существует) не обязательно будет кратчайшей по числу временных входных символов. Может оказаться, что если продолжать «тянуть» другие ветки дерева и вновь дойти до усечения по Правилу 1, то получим более длинную последовательность дуг $(Q_{00}, g_1, Q_{1j_1}), (Q_{1j_1}, g_2, Q_{2j_2}), \dots, (Q_{(k-1)j_{k-1}}, g_k, Q_{kj_k})$, но ей будет соответствовать более короткая временная последовательность (по числу входных символов), поскольку большее число дуг было помечено временными задержками.

Заключение

В работе представлено исследование отношения неразделимости для автоматов с тайм-аутами и предлагается алгоритм построения разделяющей последовательности для двух таких автоматов. Следует заметить, что построенная по данному алгоритму разделяющая последовательность не обязательно будет разделяющей последовательностью минимальной длины (по числу временных входных символов) для данной пары автоматов. Для нахождения разделяющей последовательности минимальной длины нужно при усечении дерева преемников по Правилу 1 запоминать построенную последовательность, но не прекращать дальнейших построений; затем выбрать самую короткую из множества построенных последовательностей.

Рассмотренная в работе модель временного автомата является упрощенным вариантом модели временного автомата, введенной в [2], поскольку в ней отсутствуют задержки по выходным символам. Однако предложенный алгоритм построения разделяющей последовательности может служить основой при анализе отношения неразделимости для «полноценного» варианта временного автомата. Развитием данного исследования может также стать разработка методов построения полных проверяющих тестов для временных автоматов относительно неразделимости.

ЛИТЕРАТУРА

1. Гилл А. Введение в теорию конечных автоматов. М.: Наука, 1966. 272 с.
2. Merayo M.G. Formal testing from timed finite state machines // Computer Networks. 2008. V. 52 No. 2. P. 432–460.
3. Громов М.Л., Евтушенко Н.В. Синтез различающих экспериментов для временных автоматов // Программирование. 2010. № 4. С. 1–11.
4. Zhigulin M., Maag S., Cavalli A., Yevtushenko N. FSM-based test derivation strategies for systems with time-outs // Proc. conf. QSIC'2011. 2011.
5. Starke P. Abstract automata. American Elsevier, 1972. 419 p.
6. Spitsyna N., El-Fakih K., Yevtushenko N. Studying the Separability Relation between Finite State Machines // Software Testing, Verification and Reliability. 2007. V. 17(4). P. 227–241.

Шабалдина Наталья Владимировна
Томский государственный университет,
E-mail: NataliaMailBox@mail.ru

Поступила в редакцию 31 марта 2011 г.