

## СПИСОК ЛИТЕРАТУРЫ

1. Lin C.J., Reddy S.M. On Delay Fault Testing in Logic Circuits // IEEE Transactions on Computer-Aided Design. – 1987. – V. 6. – № 5. – P. 694–503.
2. Devadas S., Keitzer K. Synthesis of Robust Delay-Fault-Testable Circuits: Theory // IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems. – 1992. – V. 11. – № 1. – P. 87–101.
3. Kohavi Z., Kohavi I. Detection of Multiple Faults in Combinational Logic Networks // IEEE Transactions on Computers. – 1972. – V. 21. – № 6. – P. 556–568.

Поступила 08.02.2010 г.

УДК 618.5:519.68

## СИНТЕЗ ТЕСТОВ С ГАРАНТИРОВАННОЙ ПОЛНОТОЙ ДЛЯ ВРЕМЕННЫХ АВТОМАТОВ

М.В. Жигулин, И.М. Дмитриев, Н.В. Евтушенко

Томский государственный университет  
E-mail: maxzh81@gmail.com

Предложен алгоритм построения полного проверяющего теста для временного автомата относительно модели «черного ящика». Предполагается, что в проверяемом автомате известны только верхняя оценка числа состояний и величина максимальной конечной задержки в состоянии. Полный тест строится по автомату-спецификации без перебора проверяемых автоматов.

### Ключевые слова:

Временной конечный автомат, задержка по времени, полный проверяющий тест.

### Key words:

Timed finite state machine, timeout, complete test suite.

### Введение

Одной из современных проблем диагностики является тестирование телекоммуникационных систем, работающих в режиме реального времени. Для описания поведения таких систем была предложена модель временного конечного автомата [1, 2], которая, кроме переходов под действием внешних входных воздействий, описывает переходы по задержке: если в некотором состоянии автомата в течение определенного периода времени отсутствуют входные воздействия, то автомат переходит в другое состояние. В модели также учитывается временная задержка для выходной реакции автомата на полученное входное воздействие. Тот факт, что проверяемая система должна выдавать только выходные последовательности, предписанные спецификацией, формально описывается отношением  $f$ -эквивалентности между временными автоматами.

В работе предлагается метод построения полного проверяющего теста относительно модели «черного ящика», т. е. для случая, когда проверяемая система имеет не больше состояний, чем спецификация, и максимальная конечная задержка в состоянии не превышает таковую в спецификации. Кроме того, показано, что такой тест также обнаруживает несоответствие проверяемой системы спецификации по времени обработки входных воздействий. В нашей работе мы рассматриваем инициальные автоматы, т. е. предполагается, что спецификация и проверяемая система обладают ис-

правным сигналом сброса, который переводит систему в начальное состояние из любого состояния.

### 1. Определения

В данном разделе мы вводим определение автомата, который явно учитывает временные аспекты. В каждом состоянии автомат может иметь временную задержку (*таймаут*): если по истечении таймаута на автомат не поступило ни одного входного воздействия, то автомат переходит в другое (предписанное) состояние. Кроме того, в явном виде задается временная задержка для выходной реакции после поступления входного воздействия. Как обычно,  $N$  используется для обозначения множества натуральных чисел;  $Z_0^+$  обозначает множество неотрицательных целых чисел; точка ‘.’ между двумя символами обозначает конкатенацию.

*Временным автоматом* [1, 2] (или просто *автоматом* в данной работе) называется семерка  $A = (S, s_0, I, O, \lambda_A, \Delta_A, \sigma_A)$ , 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,  $\Delta_A: S \rightarrow S \times (N \cup \{\infty\})$  – функция задержки, определяющая время, по истечении которого состояние автомата должно измениться, если на автомат в текущем состоянии не поступил ни один входной символ, функция  $\sigma_A: \lambda_A \rightarrow Z_0^+$  определяет время задержки выходного символа после подачи входного воздействия на соответствующем переходе. Если  $\Delta_A(s)_{i,n} = \infty$ , то предполагается, что

автомат в состоянии  $s$  может ожидать поступления входного воздействия бесконечно долго. В данной работе мы рассматриваем только полностью определенные детерминированные автоматы. Иными словами, для каждой пары  $(s,i) \in S \times I$  существует в точности одна пара  $(o,s') \in O \times S$  такая, что  $(s,i,o,s') \in \lambda_A$ . Поэтому функция  $\sigma_A$  может быть определена как отображение множества  $S \times I$  в множество  $Z_0^+$ .

Примером системы, поведение которой описывается с помощью такого временного автомата, может служить *Philip's Bounded Retransmission Protocol* (BRP). Данный протокол предназначен для отправки информации (файлов) от передатчика к приемнику через выделенные каналы связи с потерями. Передача данных происходит частями (пакетами). На передачу каждого пакета отводится определенное количество времени. Если за выделенное время после отправки данных от получателя не приходит подтверждение, то считается, что произошла потеря пакета. Попыток передачи пакета приемнику может быть несколько. Если после всех попыток подтверждение так и не пришло, то считается что канал связи неисправен, и передача данных по нему невозможна. В работе протокола участвуют четыре основных устройства: приемник, передатчик, канал связи от приемника к передатчику и канал связи от передатчика к приемнику.

Опишем работу передатчика при помощи модели временного автомата с задержками (рис. 1). В начальном состоянии (в состоянии *init*) на вход автомата поступает файл, который затем разбивается на пакеты определенной длины, и автомат переходит в состояние *next*. В этом состоянии автомат отправляет первый пакет приемнику и увеличивает значение переменной, соответствующей числу попыток передачи текущего пакета, на единицу. Автомат переходит в состояние *wait*, где в течение  $T_1$  тактов ожидается ответ от приемника. Если за время  $T_1$  ответ не пришел, то автомат возвращается в состояние *next* и производит следующую попытку передачи. Если после допустимого максимального числа попыток ответа от приемника не поступило, то автомат переходит в состояние *err* с выходным символом I\_DK (передача файла оборвалась на последнем пакете) или I\_NOK (передать файл не удалось). В этом состоянии система ожидает время  $T_{sync}$  для синхронизации соединения с другими устройствами и вновь переходит в состояние *init*. Если в состоянии *wait*, ответ от приемника пришел за время, меньшее  $T_1$ , то автомат переходит в состояние *succ*. Если передаваемый пакет был не последним, т. е. остались пакеты для передачи, то автомат из состояния *succ* переходит в состояние *next* и передает следующий пакет приемнику. Если все пакеты были переданы, то автомат из состояния *succ* переходит в состояние *init* с вход-выходным символом null/I\_OK (передача файла завершена).

Автомат, представленный на рис. 1, является временным недетерминированным частичным ав-

томатом. Причиной недетерминизма является тот факт, что описание протокола содержит так называемые контекстные переменные (наибольшее допустимое число попыток передачи, флаг для определения того, является ли пакет последним). Более подробное описание протокола, учитывающее значения этих переменных, будет детерминированным автоматом. Если в некотором состоянии не определен переход по некоторому входному символу, то, как обычно предполагается, что автомат остается в этом состоянии с выходным символом *ignore*. Таким образом, поведение BRP передатчика можно описать временным детерминированным полностью определенным автоматом.

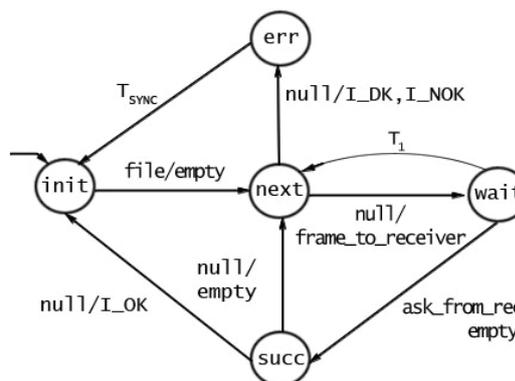


Рис. 1. Временной автомат, описывающий поведение BRP передатчика

Поскольку поведение временного автомата зависит не только от входного воздействия, но и от времени подачи этого воздействия, вводится понятие временного входного символа. *Временным входным символом* называется пара  $(i,t) \in I \times Z_0^+$ . Временной входной символ  $(i,t)$  означает, что входной символ  $i$  поступает на вход автомата в  $t$ -й такт времени после появления выходной реакции на предыдущее входное воздействие. Для того чтобы определить выходную реакцию автомата на временной входной символ  $(i,t)$  в состоянии  $s$ , необходимо вычислить в каком состоянии будет находиться автомат в  $t$ -й момент времени после перехода автомата в состояние  $s$ . Введем в каждом состоянии автомата  $A$  отображение  $T_A: S \times Z_0^+ \rightarrow S$ . По определению,  $T_A(s,0) = s$  для любого состояния  $s$ . Воспользовавшись функцией  $\Delta_A$ , можно определить состояние  $s' = T_A(s,t)$  как состояние, достижимое из состояния  $s$  по задержкам, пока сумма задержек из состояния  $s$  меньше  $t$ , и становится  $\geq t$ , если прибавить задержку в состоянии  $s'$ . Тогда во временном автомате существует временной переход  $(s,(i,t),o,s')$ , если и только если существует переход  $(T_A(s,t),i,o,s')$ . Таким образом, выходная реакция временного автомата в состоянии  $s$  на временной входной символ  $(i,t)$  (обозначение:  $out_A(s,(i,t))$ ) есть выходной символ автомата в состоянии  $T_A(s,t)$  на входной символ  $i$ .

Последовательность из временных входных символов называется *временной входной последовательностью*, и, как обычно, поведение временного авто-

мата расширяется на временные входные и выходные последовательности. Пусть задана временная входная последовательность  $\alpha=(i_1, t_1) \cdot (i_2, t_2) \cdot \dots \cdot (i_k, t_k)$  и  $\beta=o_1 \cdot o_2 \cdot \dots \cdot o_k$  – последовательность в выходном алфавите временного автомата. Четверка  $(s, \alpha, \beta, s') \in \lambda_A$ , если и только если во временном автомате существует последовательность переходов  $(s, (i_1, t_1), o_1, s_1), (s_1, (i_2, t_2), o_2, s_2), \dots, (s_{k-1}, (i_k, t_k), o_k, s')$ . В этом случае говорят, что временной автомат  $A$  в состоянии  $s$  *выдает (производит)* выходную последовательность  $\beta$  на временную входную последовательность  $\alpha$ , и состояние  $s'$  *достижимо* из состояния  $s$  по временной входной последовательности  $\alpha$ . Пара  $\alpha/\beta$  называется *функциональной временной вход-выходной последовательностью* автомата в состоянии  $s$ . Множество всех функциональных вход-выходных последовательностей автомата  $A$  в состоянии  $s$  обозначается  $FTTr_A(s)$ . Выходную последовательность автомата  $A$  в состоянии  $s$  на временную входную последовательность  $\alpha$  будем обозначать  $out_A(s, \alpha)$ .

Состояние автомата  $A$ , достижимое из начального состояния по некоторой временной входной последовательности, называется *ir-состоянием*. По определению, начальное состояние является *ir-состоянием*, т. к. оно достижимо по пустой входной последовательности (иными словами, это состояние достижимо после подачи сигнала сброса). Автомат называется *связным*, если каждое состояние  $s'$  этого автомата является *ir-состоянием* или существуют *ir-состояние*  $s$  и такое  $t$ , что  $T_A(s, t)=s'$ . Состояние связного автомата, которое не является *ir-состоянием*, т. е. достижимо из некоторого *ir-состояния* только по задержкам, называется *iur-состоянием*.

При построении проверяющих тестов с гарантированной полнотой обычно предполагается, что для описания поведения спецификации системы и ее реализации используется одна и та же математическая модель. В нашей работе мы полагаем, что поведение спецификации и любой реализации (возможно, с ошибками) описано некоторым полностью определенным детерминированным временным автоматом. Поэтому выходная реакция конформной реализации системы на каждую временную входную последовательность должна совпадать с выходной реакцией спецификации на эту последовательность. Для формального описания такого соответствия вводится отношение *f-эквивалентности*. При введении этого отношения нет необходимости учитывать задержку выходного символа после подачи входного воздействия, поэтому в разделах 2 и 3 временной автомат рассматривается как шестерка  $(S, s_0, I, O, \lambda_A, \Delta_A)$ .

Пусть  $A=(S, s_0, I, O, \lambda_A, \Delta_A)$  и  $B=(P, p_0, I, O, \lambda_B, \Delta_B)$  – временные автоматы. Состояния  $s$  автомата  $A$  и  $p$  автомата  $B$  *f-эквивалентны*, если для любой временной последовательности  $\alpha$  справедливо  $out_A(s, \alpha)=out_B(p, \alpha)$ , т. е.  $FTTr_A(s)=FTTr_B(p)$ . В противном случае состояния  $s$  и  $p$  называются *f-различимыми*.

*Временной автомат  $A$  называется f-приведенным*, если любые два различных состояния автомата *f-различимы*. Автоматы  $A$  и  $B$  *f-эквивалентны*, если *f-эквивалентны* их начальные состояния. Если автоматы  $A$  и  $B$  *f-различимы*, то временная входная последовательность  $\alpha$  такая, что  $out_A(s_0, \alpha) \neq out_B(p_0, \alpha)$  называется *f-различающей* последовательностью автоматов  $A$  и  $B$ .

## 2. Проверка f-эквивалентности двух временных автоматов

Рассмотрим временной автомат  $A=(S, s_0, I, O, \lambda_A, \Delta_A)$ . Подобно [3], для того чтобы сформулировать необходимые и достаточные условия *f-эквивалентности*, мы используем понятие *k-f-эквивалентных состояний*. Состояния  $s_1, s_2 \in S$  называются *1-f-эквивалентными*, если для любой временной входной последовательности  $\alpha=(i_1, 0) \dots (i_k, 0)$  справедливо:  $out_A(s_1, \alpha)=out_A(s_2, \alpha)$ . Если состояния  $s_1$  и  $s_2$  не являются *1-f-эквивалентными*, то существует временная входная последовательность  $\alpha=(i_1, 0) \dots (i_k, 0)$  такая, что  $out_A(s_1, \alpha) \neq out_A(s_2, \alpha)$ , т. е. последовательность  $\alpha$  *f-различает* состояния  $s_1$  и  $s_2$ .

**Утверждение 1.** Если состояния  $s_1, s_2$  автомата  $A$  не являются *1-f-эквивалентными*, то эти состояния являются *f-различимыми*.

Проверить, являются ли два состояния *1-f-эквивалентными*, можно используя классические методы из теории автоматов, удалив из временного автомата все переходы по задержкам. Однако *1-f-эквивалентности* недостаточно для *f-эквивалентности* двух состояний.

Рассмотрим автомат  $A$ , изображенный на рис. 2. Достаточно просто убедиться, что состояния 0 и 1 являются *1-f-эквивалентными*. Однако если в состоянии 0 после двух тактов времени ни одно входное воздействие ( $a$  или  $b$ ) не поступило, то автомат перейдет в состояние 2. Если же такая ситуация имеет место в состоянии 1, то автомат останется в состоянии 1. Состояния 1 и 2 *f-различимы* входным символом ( $a, 0$ ), т. е. состояния 0 и 1 окажутся различимыми, если подать входной символ  $a$  в момент времени 2, т. е. состояния 0 и 1 *f-различимы* временным входным символом ( $a, 2$ ).

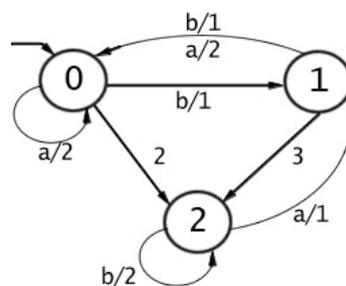


Рис. 2. Временной автомат  $A$

Состояния  $s_1$  и  $s_2$  являются *k-f-эквивалентными*, если эти состояния  $(k-1)$ -*f-эквивалентны*, и для любой входо-выходной пары  $i/o$  и времени  $t$ , спра-

ведливо:  $i/o$ -преемники состояний  $s_1$  и  $s_2$  и состояния  $T_A(s_1, t)$  и  $T_A(s_2, t)$  являются  $(k-1)$ - $f$ -эквивалентными.

**Утверждение 2.** В автомате  $A$  состояния  $s_1, s_2 \in S$  являются  $f$ -эквивалентными, если и только если состояния  $s_1$  и  $s_2$  являются  $k$ - $f$ -эквивалентными для любого  $k \geq 1$ .

**Следствие 1.** Пусть  $s_1, s_2 \in S$  суть  $f$ -эквивалентные состояния автомата  $A$ . Тогда для любого такта времени  $t$  состояния  $T_A(s_1, t)$  и  $T_A(s_2, t)$  являются  $f$ -эквивалентными, и для любого временного входного символа  $(i, t)$  и  $o \in out_A(s_1, (i, t))$  преемники состояний  $s_1$  и  $s_2$  по вход-выходному символу  $((i, t)/o)$  являются  $f$ -эквивалентными.

**Следствие 2.** Пусть  $s_1, s_2 \in S$  суть  $f$ -эквивалентные состояния полностью определенного детерминированного временного автомата  $S$ . Тогда любая временная вход-выходная последовательность из множества  $FTTr_A(s_1) = FTTr_A(s_2)$  переводит автомат  $A$  из состояний  $s_1$  и  $s_2$  в  $f$ -эквивалентные состояния.

**Утверждение 3.** Связные приведенные автоматы  $A$  и  $B$  являются  $f$ -эквивалентными, если и только если существует взаимно однозначное отображение  $h: S \rightarrow P$  такое, что  $h(s_0) = p_0$  и для любого состояния  $s \in S$ , для каждого перехода  $s \xrightarrow{i} s'$  и каждого перехода  $s \xrightarrow{i/o} s'$  автомат  $B$  имеет переходы  $h(s) \xrightarrow{i} h(s')$  и  $h(s) \xrightarrow{i/o} h(s')$ .

### 3. Построение проверяющего теста с гарантированной полнотой

Согласно классической теории автоматов [3], полный проверяющий тест можно построить только при условии, что множество рассматриваемых неисправностей ограничено, т. е. известны некоторые свойства проверяемого автомата. В данной работе мы предполагаем, что неисправность не увеличивает число состояний в реализации, т. е. каждый проверяемый автомат имеет не больше состояний, чем автомат-спецификация. Проверяемая система считается исправной, если автомат, описывающий ее поведение (проверяемый автомат), является  $f$ -эквивалентным автомату-спецификации, и на каждую временную входную последовательность проверяемая система реагирует с быстротой, предписанной автоматом-спецификацией.

#### 3.1. Синтез теста для проверки $f$ -эквивалентности

Пусть  $A = (S, s_0, I, O, \lambda_A, \Delta_A)$  – автомат-спецификация; автомат  $A$  имеет  $n$  состояний и является связным и приведенным, и  $\theta_A$  есть максимальное конечное значение таймаута в состояниях автомата  $A$ . Мы предполагаем, что неисправность не увеличивает число состояний по сравнению со спецификацией, и максимальная конечная задержка в состоянии не превосходит таковую в спецификации, т. е. проверяемый автомат есть некоторый автомат  $B$  из множества  $\mathfrak{T}_n(\theta_A)$ : автомат  $B$  имеет не более  $n$  состояний, входной и выходной алфавиты  $I$  и  $O$  и максимальную конечную задержку  $\theta_A$  в каждом из

состояний. Проверяющие тесты при использовании такой модели неисправности (модели «черного ящика») строятся на основе установления соответствия между состояниями и переходами эталонного и проверяемого автоматов [4, 5] (аналог метода Василевского ( $W$ -метода)). Поскольку в этом случае проверяются все переходы в системе, то тесты, построенные для этой модели, являются достаточно качественными. В случае, когда число состояний проверяемого автомата не больше числа состояний автомата-спецификации, все такие методы доставляют тесты полиномиальной длины относительно числа состояний.

В данной работе мы кратко описываем построение таких проверяющих тестов для временных автоматов.

Для различения состояний проверяемого автомата используются  $f$ -различающие последовательности. Множество  $W$  временных входных последовательностей называется *множеством различимости* автомата  $A$ , если для любых двух различных состояний  $s_j$  и  $s_i$  существует временная входная последовательность  $\gamma \in W$ , которая  $f$ -различает  $s_j$  и  $s_i$ , т. е.  $out_A(s_j, \gamma) \neq out_A(s_i, \gamma)$ . Такие последовательности можно достаточно просто построить на основе так называемого  $f$ -пересечения [2], и мы не останавливаемся на построении таких последовательностей.

*Проверяющий тест* есть конечное множество конечных временных входных последовательностей. Тест  $TS$  называется  $(n, \theta_A)$ -полным для автомата  $A$ , если для каждого автомата  $B \in \mathfrak{T}_n(\theta_A)$ , который не является  $f$ -эквивалентным автомату  $A$ , существует  $\alpha \in TS$ ,  $f$ -различающая автоматы  $A$  и  $B$ .

Для каждого  $ir$ -состояния  $s$  автомата  $A$ , построим временную входную последовательность  $\alpha_s$ , которая переводит автомат  $A$  из начального состояния в состояние  $s$ . Префикс замкнутое множество таких последовательностей есть множество достижимости  $V$  для  $ir$ -состояний спецификации. На первом шаге мы проверяем, что каждая из этих последовательностей переводит проверяемый автомат в новое состояние, т. е. после каждой из таких последовательностей мы подаем множество различимости и получаем первую часть  $TS_1$  проверяющего теста.

Для каждого  $iur$ -состояния  $s'$ , определим  $ir$ -состояние  $s$  такое, что через  $\tau$  тактов существует переход из состояния  $s$  в состояние  $s'$  по задержкам, т. е. если в течение  $\tau$  тактов на автомат в состоянии  $s$  не было подано ни одного входного воздействия, то автомат перейдет в состояние  $s'$ . По множеству различимости  $W$  мы строим  $W^\tau$ , заменяя первый входной символ  $(i, t)$  каждой последовательности в  $W$  на входной символ  $(i, t + \tau)$ . Объединение множеств  $\alpha_s \cdot W^\tau$  для всех  $iur$ -состояний, есть вторая часть  $TS_2$  проверяющего теста.

Если проверяемый временной автомат  $B$  правильно реагирует на все последовательности множества  $TS_1 \cup TS_2$ , то проверяемый автомат имеет  $n$  различных состояний, причем любые два различ-

ных состояния не являются  $f$ -эквивалентными, т. к. в этих состояниях автомат  $B$  имеет различные выходные реакции на последовательности из множества  $W$ . Соответственно, через  $h$  мы обозначаем взаимно однозначное отображение  $h: S \rightarrow P$  такое, что для любого  $s \in S$  и любого  $p \in P$  автоматы имеют одинаковые реакции на все последовательности из множества  $W$ , если и только если  $p=h(s)$ .

Теперь для построения полного проверяющего теста необходимо для каждого состояния  $s$  проверить все переходы из этого состояния под действием входных символов  $(i,0)$  и убедиться, что задержка в проверяемом автомате соответствует задержке в спецификации. Иными словами, необходимо проверить, что если  $(s, i, o, s') \in \lambda_A$ , то и  $(h(s), i, o, h(s')) \in \lambda_B$ , и  $\Delta_A(s)_{\downarrow N} = \Delta_B(h(s))_{\downarrow N}$  и  $h(\Delta_A(s)_{\downarrow S}) = \Delta_B(h(s))_{\downarrow P}$ .

Для того чтобы проверить переход из  $ir$ -состояния  $s$  под действием временного входного символа  $(i,0)$ , в тест включаются последовательности  $\alpha_s \cdot (i,0) \cdot W$ ,  $\alpha_s \in V$ . Для каждого  $iur$ -состояния  $s'$ , определяется  $ir$ -состояние  $s$  такое, что через  $\tau$  тактов существует переход из состояния  $s$  в состояние  $s'$  по задержкам, и в тест включаются последовательности  $\alpha_s \cdot (i, \tau) \in W$ . Совокупность всех построенных последовательностей есть третья часть  $TS_3$  проверяющего теста.

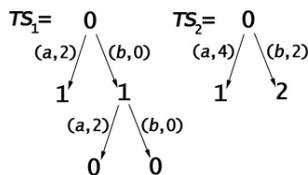
Убедимся теперь, что для каждого состояния  $s$  имеет место  $\Delta_A(s)_{\downarrow N} = \Delta_B(h(s))_{\downarrow N}$  и  $h(\Delta_A(s)_{\downarrow S}) = \Delta_B(h(s))_{\downarrow P}$ . Для того чтобы проверить переход по задержке из  $ir$ -состояния  $s$ , необходимо проверить выходные реакции на множества  $\alpha_s \cdot W^t$ ,  $t=1, \dots, \Delta_A(s)_{\downarrow N}$ , если  $\Delta_A(s)_{\downarrow N} \neq \infty$ . Если  $\Delta_A(s)_{\downarrow N} = \infty$ , то в четвертую часть  $TS_4$  проверяющего теста необходимо добавить последовательности  $\alpha_s \cdot W^t$ ,  $t=1, \dots, \theta_A$ .

Для того чтобы проверить переход по задержке из  $iur$ -состояния  $s'$ , находим  $ir$ -состояние  $s$  такое, что через  $\tau$  тактов существует переход из состояния  $s$  в состояние  $s'$  по задержкам. Соответственно, в четвертую часть  $TS_4$  проверяющего теста необходимо добавить последовательности  $\alpha_s \cdot W^{t+\tau}$ ,  $t=1, \dots, \Delta_A(s)_{\downarrow N}$ , если  $\Delta_A(s)_{\downarrow N} \neq \infty$ . Если  $\Delta_A(s)_{\downarrow N} = \infty$ , то в четвертую часть  $TS_4$  проверяющего теста необходимо добавить последовательности  $\alpha_s \cdot W^{t+\tau}$ ,  $t=1, \dots, \theta_A$ .

**Утверждение 4.** Множество  $TS = TS_1 \cup TS_2 \cup TS_3 \cup TS_4$  есть  $(n, \theta_A)$ -полный тест для автомата  $A$ .

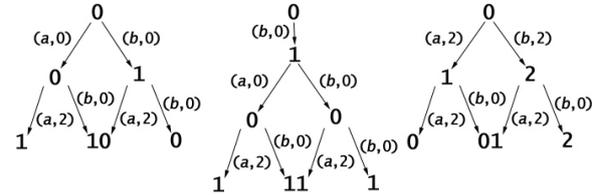
Как обычно, из множества  $TS$  можно удалить последовательности, которые являются начальными отрезками других последовательностей.

Построим  $(n, 3)$  – полный тест для автомата  $A$  на рис. 2.

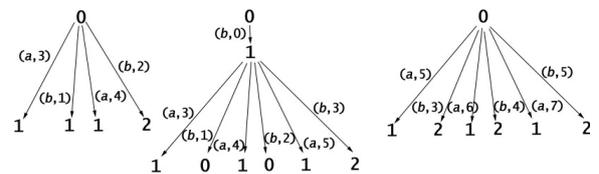


Множество достижимости  $V = \{\varepsilon, (b,0)\}$ , множество различимости  $W = \{(a,2), (b,0)\}$ . Соответственно,  $TS_1 = \{(a,2), (b,0), (b,0) \cdot (a,2), (b,0) \cdot (b,0)\}$ . Состояние 2 достижимо из начального состояния только по задержке 2, т. е.  $TS_2 = \{(a,4), (b,2)\}$ .

Проверяем переходы из состояний 0, 1 и 2 и получаем  $TS_3$ :



Проверяем переходы по задержкам из состояний 0, 1 и 2 и получаем  $TS_4$ :



Таким образом, после удаления из множества  $TS$  последовательностей, которые являются начальными отрезками других последовательностей, получим  $2: TS = \{(a,4), (b,0) \cdot (a, 0) \cdot W, (b,0) \cdot (b,0) \cdot W, (a,2) \cdot W, (b,2) \cdot W, (a,3), (b,1), (b,0) \cdot (a,3), (b,0) \cdot (b,1), (b,0) \cdot (a,4), (b,0) \cdot (b,2), (b,0) \cdot (a,5), (b,0) \cdot (b,3), (a,5), (b,3), (a,6), (b,4), (a,7), (b,5)\}$ .

Пусть выходная реакция проверяемой системы на некоторую временную входную последовательность отличается от выходной реакции автомата на рис. 2, т. е. проверяемая система является неисправной. Если поведение проверяемой системы описывается автоматом, который имеет не более трех состояний и максимальную конечную задержку не более 3, то такая неисправность будет обнаружена построенным тестом.

### 3.2. Синтез теста для проверки времени обработки входных последовательностей

Под *временем обработки входного символа*  $(i,t)$  автоматом  $A$  в состоянии  $s$  будем понимать время  $z$ , затраченное на обработку временного входного сигнала  $(i,t)$  в этом состоянии и вывод соответствующего выходного сигнала, т. е.  $z(s, (i,t)) = \sigma_A(T(s,t), i)$ . Время  $z(s, \alpha)$ , затраченное автоматом в состоянии  $s$  на обработку временной входной последовательности  $\alpha$ , назовем *временем обработки входной последовательности*  $\alpha$ , и это время равно сумме времени обработки каждого временного входного символа в последовательности.

Пусть заданы временные автоматы  $A = (S, s_0, I, O, \lambda_A, \Delta_A, \sigma_A)$  и  $B = (P, p_0, I, O, \lambda_B, \Delta_B, \sigma_B)$ . В этом разделе мы сравниваем эти автоматы по времени обработки временных входных последовательно-

стей. Поскольку такое сравнение имеет смысл только для  $f$ -эквивалентных автоматов, то мы полагаем, что автоматы  $A$  и  $B$  являются  $f$ -эквивалентными.

Автомат  $B$  строго эквивалентен (по времени обработки) автомату  $A$  (обозначение:  $A \cong_{\tau} B$ ), если автоматы  $A$  и  $B$   $f$ -эквивалентны и для любой временной входной последовательности  $z(s_0, \alpha) = z(p_0, \alpha)$ .

Автомат  $B$  не медленнее автомата  $A$  (по времени обработки), если автоматы  $A$  и  $B$   $f$ -эквивалентны и для любой временной входной последовательности  $z(s_0, \alpha) \geq z(p_0, \alpha)$ .

Автомат  $B$  не быстрее автомата  $A$  (по времени обработки), если автоматы  $A$  и  $B$   $f$ -эквивалентны и для любой временной входной последовательности  $z(s_0, \alpha) \leq z(p_0, \alpha)$ .

В данный момент мы, вообще говоря, не знаем, как можно полностью охарактеризовать отношения «не быстрее» и «не медленнее», поэтому мы вводим более строгие отношения для времени обработки входных сигналов.

Пусть связные приведенные автоматы  $A$  и  $B$  являются  $f$ -эквивалентными. Согласно утверждению 3, существует взаимно однозначное отображение  $h: S \rightarrow P$  такое, что  $h(s_0) = p_0$  и для любого состояния  $s \in S$ , для каждого перехода  $s \xrightarrow{t} s'$  и каждого перехода  $s \xrightarrow{i/o} s'$  автомат  $B$  имеет переходы  $h(s) \xrightarrow{t} h(s')$  и  $h(s) \xrightarrow{i/o} h(s')$ .

Автомат  $B$  строго не медленнее (по времени обработки) автомата  $A$  (обозначение:  $A \succ_{\tau} B$ ), если автоматы  $A$  и  $B$  являются  $f$ -эквивалентными, и для любого состояния  $s$  автомата  $A$  и любого входного символа  $i$  имеет место:  $z(h(s), (i, 0)) \leq z(s, (i, 0))$ .

Автомат  $B$  строго не быстрее (по времени обработки) автомата  $A$  (обозначение:  $A \prec_{\tau} B$ ), если автоматы  $A$  и  $B$  являются  $f$ -эквивалентными, и для любого состояния  $s$  автомата  $A$  и любого входного символа  $i$  имеет место:  $z(h(s), (i, 0)) \geq z(s, (i, 0))$ .

Согласно утверждению 3, если временная входная последовательность  $\alpha$  переводит автомат  $A$  из состояния  $s$  в  $s'$ , то автомат  $B$  перейдет из состояния  $p = h(s)$  в состояние  $p' = h(s')$ . Поэтому если  $B$  строго не медленнее (строго не быстрее) временного автомата  $A$ , то  $B$  не медленнее (не быстрее) временного автомата  $A$ .

Для проверки, является автомат  $B$  строго не медленнее (строго не быстрее) временного автомата  $A$ , достаточно построить множество входных последовательностей, которые являются обходом графа переходов автомата  $A$ .

Для каждого  $ir$ -состояния  $s$  автомата  $A$  строим подмножество последовательностей  $\alpha_s: (i, 0)$ ,  $\alpha_s \in V$ ,  $i \in I$ . Объединяем все последовательности во множество  $TTS_1$ .

Для каждого  $iur$ -состояния  $s'$  находим  $ir$ -состояние  $s$  такое, что существует переход  $s \xrightarrow{\tau} s'$ , где  $\tau$  представляет последовательность переходов по

задержкам, и последовательность  $\alpha_s \in V$ . Строим подмножество последовательностей  $\alpha_s: (i, \tau)$ ,  $\alpha_s \in V$ ,  $i \in I$ . Объединяем все последовательности в множество  $TTS_2$ .

Объединение множеств  $TTS_1$  и  $TTS_2$  есть полный тест для проверки, является ли автомат  $B$  эквивалентным (строго не медленнее, строго не быстрее) автомату  $A$  по времени обработки временных входных последовательностей.

**Утверждение 5.** Если автомат  $B \in \mathfrak{S}_n(\theta_A)$  является  $f$ -эквивалентным приведенному связному автомату  $A$ , то автомат  $B$  строго эквивалентен (по времени обработки временных входных последовательностей) автомату  $A$ , если и только если время обработки автоматом  $B$  каждого временного входного символа из последовательностей множества  $TTS_1 \cup TTS_2$  совпадает со временем обработки этого символа автоматом  $A$ .

Аналогичные утверждения можно доказать для отношений «строго не медленнее» и «строго не быстрее».

**Утверждение 6.** Если автомат  $B \in \mathfrak{S}_n(\theta_A)$  является  $f$ -эквивалентным приведенному связному автомату  $A$ , то автомат  $B$  строго не медленнее (строго не быстрее) автомата  $A$ , если и только если время обработки автоматом  $B$  каждого временного входного символа из последовательностей множества  $TTS_1 \cup TTS_2$  не больше (соответственно не меньше) времени обработки этого символа автоматом  $A$ .

Заметим, что поскольку полный тест, построенный в предыдущем разделе для проверки  $f$ -эквивалентности, включает обход графа переходов автомата  $A$ , то на основе этого теста можно также проверить, соответствует ли предъявленный автомат спецификации и по скорости обработки временных входных символов.

### Заключение

Рассмотрена модель временного автомата, предложен алгоритм построения полного проверяющего теста для временного автомата относительно модели «черного ящика». При построении теста для идентификации состояний использован метод Василевского ( $W$ -метод), однако длину теста можно сократить за счет более эффективного использования различающих последовательностей (HSI и H методы). Предложенный подход можно использовать также для случая, когда спецификация является частичным детерминированным автоматом. В этом случае достаточно предположить, что если переход в некотором состоянии не определен для заданного входного воздействия, то автомат остается в этом состоянии с выходным символом *ignore* или *null*.

Работа частично поддержана проектами ФЦП (государственный контракт № 02.514.12.402 и № П1157).

### СПИСОК ЛИТЕРАТУРЫ

1. Merayo M.G., Nunez M., Rodriguez I. Formal testing from timed finite state machines // Computer Networks. – 2008. – V. 52. – № 2. – P. 432–460.
2. Gromov M.L., Popov D.D., Yevtushenko N.V. Deriving test suites for timed Finite State Machines // Proc. of IEEE East-West Design & Test Symposium'08. – Kharkov, Ukraine: SPD FL Stepanov V.V., 2008. – P. 339–343.
3. Гилл А. Введение в теорию конечных автоматов. – М.: Наука, 1962. – 272 с.
4. Hennie F.C. Fault-detecting experiments for sequential circuits // Proc. of 5<sup>th</sup> Annual Symp. on Switching Theory and Logical Design. – USA, Nov. 1964. – P. 95–110.
5. Василевский М.П. О распознавании неисправности автоматов // Кибернетика. – 1973. – № 4. – С. 93–108.

Поступила 11.02.2010 г.

УДК 519.876.5

## ОРГАНИЗАЦИЯ И ПРИНЦИП РАБОТЫ МНОГОФУНКЦИОНАЛЬНЫХ СХЕМ МОДЕЛИРОВАНИЯ

Г.П. Цапко, С.Г. Цапко, Д.В. Тараканов

Томский политехнический университет  
\*Сургутский государственный университет  
E-mail: serg@aics.ru

*Предложены механизмы для реализации многофункциональных схем моделирования. Показана реализация схемы моделирования с разомкнутой структурой, рассмотрены временные режимы работы последовательной многофункциональной схемы. Приведена реализация циклической многофункциональной схемы моделирования и ее реализация в терминах E-сети. Представлена реализация разработанной замкнутой многофункциональной схемы моделирования, обеспечивающей многократное селективное обслуживание группы разнородных функциональных процессов с минимальным количеством E-сетевых моделей.*

### Ключевые слова:

Многофункциональная схема, имитационное моделирование, фишка, переход, позиция.

### Key words:

Multifunctional circuit simulation, chip, shift, position.

При практической реализации динамических моделей приходится достаточно часто решать типовые вычислительные и алгоритмические задачи, в том числе с помощью алгоритмических схем моделирования. Данные задачи сводятся к реализации типовых логических  $\{Alg\}$ , вычислительных  $\{\rho\}$  операций и операторов задержки продвижения фишек  $\{\Delta t\}$  по E-сети. Для уменьшения общего количества EN (E-net) элементов (позиций, переходов) в динамической модели (ДМ) предлагается использовать многофункциональные схемы моделирования (МСМ), которые работают по принципу «одна EN схема – несколько моделей». Это позволит значительно сократить не только размер модели, но и уменьшить трудозатраты на создание ДМ и автоматизировать процесс моделирования. Для решения проблемы реализации многофункциональности желательно отказаться от традиционного принципа программирования E-сетевых схем, когда функционирование элементов EN структуры строго фиксировано в рамках программного кода. Для введения принципа автоматической адаптации EN схемы предлагается использовать следующие механизмы:

1) генерация профилировщиком заданной архитектуры модели согласно цели функционирования

и выбор подмножества E-сетевых подсистем, необходимых для формирования текущей реализации модели;

- 2) транспортировка совокупностью фишек по E-сети модели служебных данных  $Atr.Data_C$ , необходимых для работы E-сетевых переходов.  $Atr.Data_C$  включает в себя следующие элементы: идентификатор, операторы преобразования атрибутов, операторы задержки, управляющие операторы;
- 3) настройка требуемых переходов EN модели путем записи служебных данных в буфер перехода;
- 4) селективный процесс работы переходов модели согласно идентификационным параметрам входного потока данных (фишек).

Организация процесса работы системы профилирования подробно изложена в [1]. В процессе функционирования система профилирования генерирует служебные фишки  $V_{in}^c(\{atr_i\})$  которые последовательно проходят через E-сетевую схему динамической модели и при активизации переходов происходит считывание данных в буфер перехода  $BUF_{Ci}$  (рис. 1).

После записи служебных данных  $Data_C$  в буфер перехода происходит изменение флага  $F_{ar}$  фишки