ИЗВЕСТИЯ ВЫСШИХ УЧЕБНЫХ ЗАВЕДЕНИЙ
№ 10/3 ФИЗИКА 2015

УДК 519.7

O. KONDRATYEVA[1,2], *N. YEVTUSHENKO*[1], *A. CAVALLI*[2]

CONFORMANCE RELATIONS FOR FINITE STATE MACHINES WITH TIMEOUTS

In the paper, we study the conformance relations for complete and partial FSMs with Timeouts with real timer variable. The conformance relations are defined based on relations between sets of timed traces of compared machines for either real or integer values of timer variable. We show that as far as timeouts and boundaries of output delays intervals are integers, it is sufficient to compare TFSMs on the sets of timed traces with integer values of timer variable.

Key words: *finite state machines with timeouts, conformance relations, partial machine, integer-valued timer.*

The problem of describing the behavior of discrete event systems transforming input sequences of actions into output sequences arises in a number of applications, often requiring taking into account timed aspects of their behavior, and hence, developing appropriate models. One of the core questions while modeling time in discrete systems is defining the value domain for the time variable and range of timed functions, i.e., deciding whether to model time on continuous or discrete scale. The former seems to be more physically-realistic interpretation, while the latter has the benefits of more efficient and precise treatment in software tools. In this paper, we consider a timed extension to Finite State Machine (FSM) model that is widely used for discrete event system synthesis and analysis, namely, the Finite State Machine with Timeouts (TFSM) whose time functions are usually defined within integer range (further denote the set of integers as $\mathbf{N}$) and therefore becoming one of the main criticizing points while comparing the TFSM to other timed finite state models like timed or hybrid automata [1]. We first provide the definition of TFSM generalizing the range of its time functions to real values (denoting the set of non-negative real as $\mathbf{R}^+$), and then show that the restriction of timer variable to integer values preserves essential conformance relations between TFSMs, either for complete or partial cases.

Definition 1. A *Finite State Machine with Timeouts* (TFSM) is a 7-tuple $S = (S, I, O, \lambda_S, s_0, \Delta_S, \sigma_S)$, where the 5-tuple $(S, I, O, \lambda_S, s_0)$ is a classical FSM [2] augmented with a *timeout function* $\Delta_S: S \to S \times (\mathbf{N} \cup \{\infty\})$ and an *output delays* function $\sigma_S: \lambda_S \to Time$, where the range of the output delays function *Time*, in general, does not coincide with the set of integers $\mathbf{N}$. The *timeout function* $\Delta_S(s) = (s_T, T)$ prescribes for each state $s \in S$ the maximal time $T \in (\mathbf{N} \cup \{\infty\})$ (timeout) of the idle waiting at the state s for an input to be applied; and the next state $s_T \in S$ which the machine moves to if no input has been applied before the timeout expires. By definition, if $\Delta_S(s) = (s_T, \infty)$ then $s_T = s$, i.e., the machine can stay waiting for an input at state s infinitely long. The *output delay* $\sigma_S: \lambda_S \to Time$ function defines for each transition $tr = (s_1, i, s_2, o) \in \lambda_S$ the set of timed intervals $\{ [l; r) \mid l < r \wedge l, r \in (\mathbf{N} \cup \{0, \infty\}) \}$ within which the machine can process the applied input, execute the transition and produce the output [3-5]. Each TFSM has an internal timed variable – *timer* –indicating how much time has passed since the TFSM reached its current state or received the input that is being currently processed (and hence being reset after each transition).

The behavior of the TFSM is characterized by the set of (timed) traces it accepts. Denote $\mathbf{X} \in \{\mathbf{N}, \mathbf{R}^+\}$ the value domain for the timer variable. A sequence $(i_1, t_1) \dots (i_m, t_m)$ of timed inputs $(i_k, t_k) \in I \times \mathbf{X}$ is a *timed input* sequence, indicating for all $1 \le k \le m$ that an input i_k is applied to the TFSM when the timer has value $t_k \in \mathbf{X}$, and a sequence $(o_1, k_1) \dots (o_m, k_m)$ of timed outputs $(o_k, t_k) \in O \times \mathbf{X}$ is a *timed output* sequence, indicating for all $1 \le k \le m$ that an output o_k is produced by the TFSM exactly at the moment of time t_k after the input was applied. Similar to [5], in order to extend the transition function to timed input and output sequences, we define the function $time_S: S \times \mathbf{X} \to S$ which for a given state s and time value t computes the state to which TFSM moves according to timeout function t time units after reaching the state s. The value of $time_S(s, t)$ for the state s with $\Delta_S(s) = (s_p, T)$ is calculated iteratively: 1) if $t < T$ then $time_S(s, t) = s$. In particular, if $\Delta_S(s) = (s, \infty)$ then $time_S(s, t) = s$ for any value of t; 2) if $t = T$ then $time_S(s, t) = s_p$; 3) if $t > T$ then $time_S(s, t) = time_S(s_p, t - T)$.

Response of the TFSM in state s to a timed input (i, t) is calculated as a response to input i in the state $time_S(s, t)$, i.e., the transition relation λ_S is extended with transition $(s, (i, t), s', o)$ if there is a transition $(time_S(s, t), i, s', o)$ and output delay function is defined as $\sigma_S((s, (i, t), s', o)) = \sigma_S((time_S(s, t), i, s', o))$.

In this case, the corresponding timed output (o, t') where $t' \in \sigma_S((s, (i, t), s', o))$ is a possible response of the TFSM to the timed input (i, t) applied at the state s.

Consider the timed input sequence $\alpha = (i_1, t_1)(i_2, t_2)\ldots(i_n, t_n)$. Timed input sequence α is called *acceptable* by TFSM S in state s if there exists such timed output sequence $\beta = (o_1, k_1)(o_2, k_2)\ldots(o_n, k_n)$ and a chain of states $s_1s_2\ldots s_n$ such that λ_S contains $(s, (i_1, t_1), s_1, o_1)$, $(s_1, (i_2, t_2), s_2, o_2)$, … $(s_{n-1}, (i_n, t_n), s_n, o_n)$ and $k_1 \in \sigma_S((s, (i_1, t_1), s_1, o_1))$, $k_2 \in \sigma_S((s_1, (i_2, t_2), s_2, o_2))$, … $k_n \in \sigma_S((s_{n-1}, (i_n, t_n), s_n, o_n))$. Then, α/β is called the *timed trace* of the TFSM S in state s. Denote the set of all acceptable input sequences of S in state s as $in^{X}_S(s)$, all the traces of S in state s as $trace^{X}_S(s)$ and the set of all timed output sequences β, such that α/β is a timed trace, as $out^{X}_S(s, \alpha)$. Note, that we use super-index $\mathbf{X} \in \{\mathbf{N}, \mathbf{R}^+\}$ to mark whether timed traces are considered with integer or real values of timer variable. In terms of traces, notion of completeness is defined as follows: the TFSM S is *complete* if any timed input sequence α is acceptable in the initial state of S, i.e., if $in^{X}_S(s) = (I \times \mathbf{X})^*$, otherwise the TFSM S is *partial*.

In order to design and analyze different interactive systems there should exist formal relations between two systems which allow comparing their behaviors. Since the specifications for real systems are often incomplete, we consider not only complete conformance relations, implying for compared machines to be defined on exactly the same sets of input sequences, but also quasi conformance relations. For FSMs, such relations are well defined [2, 6], and we modify these relations for TFSMs.

Definition 2. Given TFSMs S and P over the same input and output alphabets, we consider the following conformance relations. These definitions are the same for cases of integer or real values of timer variable in the sets of timed traces, and the choice of value domain is denoted with the index $\mathbf{X} \in \{\mathbf{N}, \mathbf{R}^+\}$.

1. *Equivalence*, written $S \cong_X P$: the TFSMs S and P are *equivalent*, if the sets of their traces coincide, i.e., it holds that $trace^{X}_S(s_0) = trace^{X}_P(p_0)$.
2. *Reduction*, written $S \leq_X P$: the TFSM S is called a *reduction* of the TFSM P if $trace^{X}_S(s_0) \subseteq trace^{X}_P(p_0)$, i.e., the behavior of S is contained in the behavior of P.
3. *Quasi-equivalence*, written $S \sqsupseteq_X P$: the TFSM S is called *quasi-equivalent* to the TFSM P if $in^{X}_S(s_0) \supseteq in^{X}_P(p_0)$ and for all $\alpha \in in^{X}_P(p_0)$ it holds that $out^{X}_S(s_0, \alpha) = out^{X}_P(p_0, \alpha)$, i.e. S and P have the same output responses to all the input sequences acceptable by P.
4. *Quasi-reduction*, written $S \lesssim_X P$: the TFSM S is called a *quasi-reduction* of the TFSM P, if $in^{X}_S(s_0) \supseteq in^{X}_P(p_0)$ and for all $\alpha \in in^{X}_P(p_0)$ it holds $out^{X}_S(s_0, \alpha) \subseteq out^{X}_P(p_0, \alpha)$, i.e. for all input sequences accepted by the TFSM P, the TFSM S can produces some of the output responses produced by P.

In case of complete TFSMs, by definition, the quasi-equivalence and quasi-reduction coincide with equivalence and reduction relations correspondingly. The equivalence and reduction relations are defined regardless of whether compared TFSMs are complete and deterministic or partial and nondeterministic. For partial TFSMs the equivalence means that behaviors of both compared TFSMs should be defined on exactly the same sets of input sequences. For real systems this requirement might become too strong in case of under- or partially-specified systems, when the implementations should be able to support specified input sequences but allowed to have additional functionality for underspecified ones [6].

The problem is that in general case inputs are applied and outputs are produced at any real moments of time, i.e., $X = R^+$, and in this case all these relations are defined over sets of timed sequences for real time instances and, hence, uncountable sets of traces. But since all the timeouts in the TFSMs and boundaries of intervals for output delays are integers and timer variable is reset after each transition, we can restrict the sets of traces to integer time instances preserving the above conformance relations between TFSMs.

The restriction to integer valued timer in timed traces description of TFSM behavior is possible due to following properties.

Property 1: for any state s it holds that $time_S(s, n + \delta) = time_S(s, n)$ for all $n \in \mathbf{N}$ and $\delta \in [0,1)$.

Proof. 1) if $\Delta_S(s) = (s', T)$ and $n < T$ or $\Delta_S(s) = (s,\infty)$ then $n + \delta < T$ and $time_S(s, n + \delta) = time_S(s, n) = s$; 2) if $\Delta_S(s) = (s', T)$ and $n = T$ then $time_S(s, n + \delta) = time_S(s', \delta) = s'$ since $\delta < 1$; 3) if $\Delta_S(s) = (s', T)$ and $n > T$ then $time_S(s, n + \delta) = time_S(s', n - T + \delta)$ which iteratively is reduced to the previous clause. □

As a corollary, $(s, (i, n + \delta), o, s_n) \in \lambda_S$ if and only if $(s, (i, n), o, s_n) \in \lambda_S$. (**Property 2**).

Consider a timed (input or output) sequence $\alpha = (a_1, t_1)(a_2, t_2)\ldots(a_k, t_k)$ where $t_j = n_j + \delta_j$, $\delta_j \in [0,1)$ and $n_j \in \mathbf{N}$ for all $1 \leq j \leq k$. We denote $\alpha^{N} = (a_1, n_1)(a_2, n_2)\ldots(a_k, n_k)$.

Proposition 1. For any state s it holds that $\alpha \in in^{\mathrm{R}}{}_S(s)$ if and only if $\alpha^{\mathrm{N}} \in in^{\mathrm{N}}{}_S(s)$.

Proof. Consider two input sequences $\alpha = (i, n+\delta)\alpha'$ and $\alpha_1 = (i, n)\alpha'$. It holds that $\alpha \in in^{\mathrm{R}}{}_S(s)$ if and only if there exist s_1 and o such that $(time_S(s, n + \delta), i, o, s_1) \in \lambda_S$, and $\alpha' \in in^{\mathrm{R}}{}_S(s_1)$. Due to Property 1 and Property 2, it holds $(time_S(s, n), i, o, s_1) \in \lambda_S$, and since $\alpha' \in in^{\mathrm{R}}{}_S(s_1)$ then $\alpha_1 \in in^{\mathrm{R}}{}_S(s_1)$. The proposition then can be proved by induction on the length of sequence α'.□

As a corollary, the following proposition holds:

Proposition 2. For any state s and any input sequence α, it holds $out^{\mathrm{R}}{}_S(s, \alpha) = out^{\mathrm{R}}{}_S(s, \alpha^{\mathrm{N}})$.

To restrict timed output sequences to integer values of time, we apply the assumption that the time is measured by a global discrete timer, then if an output o is produced at some time instance $n+\delta$, with $n \in \mathbf{N}$ and $\delta \in [0,1)$, we observe the timer value n and denote this timed output as (o, n). In other words, we consider any timed output (o, n) as an output o produced any moment of time between $[n, n+1)$ after a corresponding input was applied, and two timed outputs (o, n) and $(o, n+\delta)$ are considered equivalently observed for all $n \in \mathbf{N}$ and $\delta \in [0,1)$.

Proposition 3: Given TFSMs S and P defined over the same input and output alphabets; and $@ \in \{\cong, \leq, \sqsupseteq, \lesssim\}$ being a conformance relation. Then; $S\ @_{\mathrm{R}}\ P$ if and only if $S\ @_{\mathrm{N}}\ P$.

The *proof* directly follows from Proposition 1, Proposition 2 and Definition 2. □

In conclusion, in this paper we considered the conformance relations for the Finite State Machines with Timeouts for both complete and partial machines, and justified the usage of discrete timer variable when describing machine behavior with the set of accepted timed traces. Obtained results allow simulating the TFSM behaviors with real timers by the corresponding finite automata, similar to what we have done for TFSMs with integer timers in our previous works [e.g., 7], which brings all the power of regular languages and classical automata theory to be adapted for solving synthesis and analysis problems for timed models.

REFERENCES

1. Henzinger T.A.. The theory of hybrid automata. // Verification of Digital and Hybrid Systems. – SpringerVerlag, Berlin, 2000 – pp. 265–292.
2. Hopcroft J., Motwani E. R., Ullman J.D. Introduction to Automata Theory, Languages, and Computation . – 2nd edition. – Addison-Wesley, 2007. – 535 p.
3. Merayo M.G., Nunez M., Rodriguez I. Formal Testing from Timed Finite State Machines // Computer Networks. – 2008. – Vol. 52 №2. – pp.432-460.
4. Bresolin D., El-Fakih K., Villa T., Yevtushenko N. Deterministic Timed Finite State Machines: Equivalence Checking and Expressive Power // Proceedings GandALF 2014. – EPTCS 161, 2014. – pp. 203-216.
5. Zhigulin M., Yevtushenko N., Maag S., Cavalli A.R. FSM-Based Test Derivation Strategies for Systems with Time-Outs // Proceedings of the international conference QSIC 2011. – pp. 141-149.
6. Kushik N., Yevtushenko N., Cavalli A. On Testing against Partial Non-observable Specifications // Proceedings of 9th International Conference on the Quality of Information and Communications Technology (QUATIC). – 2014. – pp. 230-233.
7. Kondratyeva O., Yevtushenko N., Cavalli A. Solving parallel equations for Finite State Machines with Timeouts // Proceedings of the Institute for System Programming. – Vol. 26 (Issue 6). 2014. – pp. 85-98.

[1]National Research Tomsk State University, Tomsk, Russia
[2]Telecom SudParis, Evry, France
E-mail: kondratyeva.olga.vic@gmail.com, yevtushenko@sibmail.com, ana.cavalli@telecom-sudparis.eu

Kondratyeva Olga, doctorant;
Yevtushenko Nina, Dr.Sc., professor;
Cavalli Ana, PhD, professor.

О.В. КОНДРАТЬЕВА[1,2], Н.В. ЕВТУШЕНКО[1], А.КАВАЛЛИ[2]

ОТНОШЕНИЯ КОНФОРМНОСТИ ДЛЯ ВРЕМЕННЫХ АВТОМАТОВ С ТАЙМАУТАМИ

В статье рассматриваются отношения конформности для полностью определенных и частичных автоматов с таймаутами. В общем случае, отношения конформности определяются как для целых, так и для действительных значений временной переменной автомата. В статье показано, что так как таймауты и границы временных интервалов для функции задержки выхода задаются целыми числами, то при сравнении временных автоматов относительно рассмотренных отношений достаточно рассматривать случай целочисленных значений временной переменной.

Ключевые слова: *временной автомат с таймаутами, отношения конформности, целочисленная временная переменная.*

СПИСОК ЛИТЕРАТУРЫ

1. Henzinger T.A.. The theory of hybrid automata. // Verification of Digital and Hybrid Systems. – SpringerVerlag, Berlin, 2000 – pp. 265–292.
2. Hopcroft J., Motwani E. R., Ullman J.D. Introduction to Automata Theory, Languages, and Computation . – 2nd edition. – Addison-Wesley, 2007. – 535 p.
3. Merayo M.G., Nunez M., Rodriguez I. Formal Testing from Timed Finite State Machines // Computer Networks. – 2008. – Vol. 52 №2. – pp.432-460.
4. Bresolin D., El-Fakih K., Villa T., Yevtushenko N. Deterministic Timed Finite State Machines: Equivalence Checking and Expressive Power // Proceedings GandALF 2014. – EPTCS 161, 2014. – pp. 203-216.
5. Zhigulin M., Yevtushenko N., Maag S., Cavalli A.R. FSM-Based Test Derivation Strategies for Systems with Time-Outs // Proceedings of the international conference QSIC 2011. – pp. 141-149.
6. Kushik N., Yevtushenko N., Cavalli A. On Testing against Partial Non-observable Specifications // Proceedings of 9th International Conference on the Quality of Information and Communications Technology (QUATIC). – 2014. – pp. 230-233.
7. Кондратьева О.В., Евтушенко Н.В., Кавалли А.Р. Решение автоматных уравнений для временных автоматов относительно параллельной композиции // Труды Института системного программирования РАН. – Том 26. Выпуск 6. – 2014. – стр. 85-98.

[1]Национальный исследовательский Томский государственный университет, г. Томск, Россия
[2]Университет Телеком Южный Париж, г.Эври, Франция
E-mail: kondratyeva.olga.vic@gmail.com, yevtushenko@sibmail.com, ana.cavalli@telecom-sudparis.eu

Кондратьева Ольга Викторовна, аспирант;
Евтушенко Нина Владимировна, д.т.н., профессор;
Кавалли Ана, профессор.