UDC 519.7

Deriving homing sequences for Finite State Machines with timed guards

Tvardovskii Aleksandr (National Research Tomsk State University),
Yevtushenko Nina (Ivannikov Institute for System Programming of the RAS, National Research University Higher School of Economics)

State identification is the well-known problem in the theory of Finite State Machines (FSM) where homing sequences (HS) are used for the identification of a current FSM state, and this fact is widely used in the area of software testing and verification. For various kinds of FSMs, there exist sufficient and necessary conditions for the existence of preset and adaptive HS and algorithms for their derivation. Nowadays timed aspects become very important for hardware and software systems. In this work, we address the problem of checking the existence and derivation of homing sequences for FSMs with timed guards. The investigation is based on the FSM abstraction of a Timed FSM.

Keywords: *Finite State Machine, Timed guards, FSM abstraction, Homing sequence.*

1. Introduction

Testing is an important part of the hardware and software life cycle and since the complexity of telecommunication and other control systems permanently increases, formal methods for deriving high quality tests are in a great demand [1-2]. When deriving tests with guaranteed fault coverage of reasonable length, state identification sequences for Finite State Machines (FSM) are widely utilized [3]. Homing sequences (HS) allow determining a current state of an FSM under test and can be efficiently used for reducing testing efforts in active and passive testing [4, 5]. Various approaches for deriving homing sequences are developed and these sequences can be preset or adaptive [7, 8, 9]. Preset input sequences are derived before starting the identification procedure based on a successor tree of an FSM under investigation [6, 7] and such techniques exist for deterministic and nondeterministic, partial and complete, weakly initialized and non-initialized FSMs [8].

Nowadays time aspects become very important for digital and hybrid systems, and, respectively, classical FSMs have been extended with time variables [11-15]. A timed FSM (TFSM) is an FSM annotated with a *clock* and extended by input/output timeouts [12, 14] and input/output timed

guards [11, 15]. Input timed guards describe the behavior at a given state for inputs which arrive during an appropriate time interval until the state timeout expires. As mentioned above, methods for deriving preset HS are well studied for classical FSMs and in this work, we consider the problem of the HS derivation for FSMs with timed guards.

The rest of the paper has the following structure. Section 2 contains preliminaries. In Section 3, the problem of deriving a HS for an FSM with timed guards is investigated. Section 4 contains optimization methods for solving a HS problem for FSMs with timed guards. Section 5 concludes the paper.

2. Preliminaries

In this section, we briefly remind the notions of classical and Timed Finite State Machines and discuss existing methods for representing a Timed FSM by the corresponding abstraction that is a classical FSM.

2.1. Finite State Machines

Finite State Machines (FSM) [3] or simply *machines* are used for describing the behavior of a system that moves from state to state under input stimuli and produces a prescribed output response. Formally, an FSM is a 4-tuple $\mathsf{S} = (S, I, O, h_S)$ where S is a finite non-empty set of states, I and O are input and output alphabets, and $h_S \subseteq (S \times I \times O \times S)$ is the transition (behavior) relation. Such FSMs are sometimes called non-initialized FSMs; an *initialized* FSM has the designated initial state s_0 and is a 5-tuple (S, s_0, I, O, h_S). A transition (s, i, o, s') describes the situation when an input i is applied to S at the current state s. In this case, the FSM moves to state s' and produces the output (response) o. In this work we consider complete observable machines, i.e., machines where for each pair $(s, i) \in S \times I$ there exists $(o, s') \in O \times S$ such that $(s, i, o, s') \in h_S$ and for every two transitions (s, i, o, s_1), $(s, i, o, s_2) \in h_S$ it holds that $s_1 = s_2$. A complete FSM S is *deterministic* if for each pair $(s, i) \in S \times I$ there exists exactly one $(o, s') \in O \times S$ such that $(s, i, o, s') \in h_S$.

A trace or an *Input/Output sequence* α/γ of the complete FSM S at state s is a sequence of consecutive input/output pairs starting at the state s, where α is the input sequence and γ is the corresponding output sequence. Given a complete observable FSM S, states s and p are *equivalent* if the sets of output responses at these states coincide for each input sequence. A complete deterministic FSM S is *reduced* if every two different states $s_1, s_2 \in S$ are not equivalent. FSM S is *strongly connected* if for each pair of states $s_1, s_2 \in S$ there exists a trace that takes the FSM from state s_1 to state s_2.

2.2. Timed Finite State Machines

In this paper, we consider FSMs with timed guards (TFSM), i.e., FSMs which are enriched with a clock variable and timed guards [11, 13, 15]. A non-initialized TFSM is a 4-tuple $\mathsf{S} = (I, S, O, h_{\mathsf{S}})$ where S is a finite non-empty set of states, I and O are input and output alphabets, Π is a set of *input timed guards* and $h_{\mathsf{S}} \subseteq S \times I \times O \times S \times \Pi$ is the *transition relation*. An initialized TFSM has the designated initial state s_0. An input timed guard $g \in \Pi$ describes the time domain when a transition can be executed and is given in the form of interval $<min, max>$ of $[0; \infty)$, where $< \in \{(, [\}, > \in \{),]\}$. We also denote B_{S} the largest finite boundary of timed guards of S. The transition $(s, i, o, s', g) \in S \times I \times O \times S \times \Pi$ means that TFSM S being at state s accepts an input i applied at time $t \in g$ measured from the initial moment or from the moment when TFSM S has produced the last output; the clock then is set to zero and S produces output o. Given TFSM S, S is a *complete* TFSM if every input is defined at every state and the union of all input timed guards at any state s under every input i equals $[0; \infty)$. A TFSM S is *deterministic* if for every two transitions (s, i, o_1, s_1, g_1), $(s, i, o_2, s_2, g_2) \in h_{\mathsf{S}}$, $s_1 \neq s_2$ or $o_1 \neq o_2$, it holds that $g_1 \cap g_2 = \varnothing$, otherwise, TFSM S is *nondeterministic*. A TFSM is *observable* if for every two transitions (s, i, o, s_1, g_1), $(s, i, o, s_2, g_2) \in h_{\mathsf{S}}$, where $g_1 \cap g_2 \neq \varnothing$, it holds that $s_1 = s_2$.

A *timed input* is a pair (i, t) where $i \in I$ and t is a real; a timed input (i, t) means that input i is applied to the TFSM at time instance t measured from the initial moment or from the moment when the last input was applied to TFSM S. A sequence of timed inputs $\alpha = (i_1, t_1) \ldots (i_n, t_n)$ is a *timed input sequence*. Given a timed input sequence $(i_1, t_1) \ldots (i_n, t_n)$, an input i_1 is applied when the clock value is equal to t_1; after applying the input, the machine produces a prescribed output and the clock is set to 0. The machine is then waiting for the next input i_2 that is applied when the clock value equals t_2. A sequence $\alpha/\gamma = (i_1, t_1)/o_1 \ldots (i_n, t_n)/o_n$ of consecutive pairs of timed inputs and outputs starting at the state s is a *timed trace* of TFSM S at state s. Similar to FSMs, α is an applied timed input sequence while γ is the corresponding output response of the TFSM to sequence α of applied inputs. For example, when the timed input $(i_1, 1.7)$ is applied to TFSM S (Figure 1) at state s_1 the TFSM moves to state s_3, produces output o_2, reset the clock and waits for the next input.

The notions of reduced and strongly connected TFSMs are similar to those of classical FSMs up to the replacement of an input sequence to a timed input sequence.

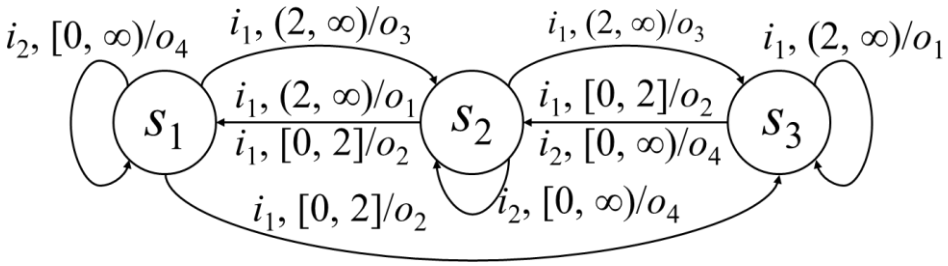

Fig. 1. FSM with timed guards S.

2.3. FSM Abstraction

In a number of cases, the behavior of a TFSM can be adequately described by a classical FSM that is called the *FSM abstraction* of the TFSM [13]. Given a complete observable possibly nondeterministic TFSM $\mathsf{S} = (S, I, O, h_\mathsf{S})$ with the largest finite boundary of timed guards B_S, a corresponding FSM abstraction $\mathsf{A_S} = (S, I_A, O, h_{\mathsf{AS}})$, where $I_A = \{(i, [0, 0]), (i, (0, 1)), \ldots, (i, (B_\mathsf{S} - 1, B_\mathsf{S})), (i, [B_\mathsf{S}, B_\mathsf{S}]), (i, (B_\mathsf{S}, \infty)): i \in I\}$, can be derived as follows. There is a transition $(s, (i, g_i), o, s') \in h_{\mathsf{AS}}$ if and only if there is a transition $(s, i, o, s', g) \in h_\mathsf{S}$ such that $g_i \subseteq g$. For the nondeterministic TFSM S in Figure 1, Table 1 represents the flow table of the corresponding FSM abstraction where rows correspond to inputs, columns correspond to states and a corresponding item for state s and input i contains the corresponding pairs of state s' and output o such that $(s, i, o, s') \in h_{\mathsf{AS}}$. A timed input sequence $\alpha = (i_1, t_1) \ldots (i_n, t_n)$ of the FSM with timed guards S can be transferred into a corresponding sequence of inputs $\alpha_{FSM} = (i_1, g_1) \ldots (i_n, g_n)$ for the FSM abstraction $\mathsf{A_S}$ and vice versa where $t_j \in g_j$, $j = 1\ldots n$. By direct inspection, a reader can assure that for every other input, a corresponding row coincides with one of three inputs colored in grey.

Table 1. Flow table of the FSM abstraction of the TFSM in Fig 1.

i/s	s_1	s_2	s_3
$(i_1, [0, 0])$	s_3/o_2	s_1/o_2	s_2/o_2
$(i_1, (0, 1))$	s_3/o_2	s_1/o_2	s_2/o_2
$(i_1, [1, 1])$	s_3/o_2	s_1/o_2	s_2/o_2
$(i_1, (1, 2))$	s_3/o_2	s_1/o_2	s_2/o_2
$(i_1, [2, 2])$	s_3/o_2	s_1/o_2	s_2/o_2
$(i_1, (2, \infty))$	s_2/o_3	s_1/o_1, s_3/o_3	s_3/o_1

i/s	s_1	s_2	s_3
$(i_2, [0, 0])$	s_1/o_4	s_2/o_4	s_2/o_4
$(i_2, (0, 1))$	s_1/o_4	s_2/o_4	s_2/o_4
$(i_2, [1, 1])$	s_1/o_4	s_2/o_4	s_2/o_4
$(i_2, (1, 2))$	s_1/o_4	s_2/o_4	s_2/o_4
$(i_2, [2, 2])$	s_1/o_4	s_2/o_4	s_2/o_4
$(i_2, (2, \infty))$	s_1/o_4	s_2/o_4	s_2/o_4

Note that the FSM abstraction of a complete deterministic (nondeterministic) TFSM is a complete deterministic (nondeterministic) FSM and similar to the statement proven in [13] for initialized TFSMs, the following statement holds for a non-initialized TFSM.

Proposition 1. There exists a timed trace α/γ at state s of a non-initialized TFSM S if and only if the FSM abstraction $\mathsf{A_S}$ has a trace α_{FSM}/γ at state s.

3. Homing sequences for TFSMs

A Homing Sequence (HS) allows to determine a state reached by an FSM after applying this input sequence and observing the produced outputs. In this section, we define a Homing Sequence for a complete FSM with timed guards and show how this sequence can be derived based on the FSM abstraction of the TFSM.

Given a trace α/γ of a complete observable possibly nondeterministic FSM, state s' is the α/γ-successor of state s in FSM S if S moves from state s to state s' by trace α/γ. The α/γ-successor of a subset S' of states is the union of α/γ-successors over all states $s \in S'$; note that the α/γ-successor of a state s as well as of a subset S' can be the empty set. In the same way the α/γ-successor is defined for a timed trace α/γ for a complete observable TFSM.

An input sequence α is a *Homing Sequence* (HS) for a non-initialized complete observable possibly nondeterministic FSM S if for each output sequence γ, the α/γ-successor of S is a singleton or does not exist. A HS for a complete observable FSM can be derived using an appropriate truncated successor tree [3, 8, 16].

When timed FSMs are considered, the value of the clock variable must be taken into account before starting a homing experiment. By definition, the value of the TFSM clock is always reset to zero when an input is applied and respectively a HS is derived under the same assumption. Thus, a timed input sequence α is a *homing sequence* for a non-initialized complete observable possibly nondeterministic FSM with timed guards S if and only if the α/γ-successor of S for each output sequence γ is a singleton or the empty set. Proposition 1 and the one-to-one correspondence between states of an FSM with timed guards and its FSM abstraction imply the following statement.

Proposition 2. A timed input sequence α is a HS for a complete non-initialized observable possibly nondeterministic FSM with timed guards S if and only if the input sequence α_{FSM} is a HS for the FSM abstraction $\mathsf{A_S}$.

Therefore, a HS for an FSM with timed guards can be constructed as a HS for its FSM abstraction. At the next step, a HS for the FSM abstraction should be transformed into a timed HS for the FSM with timed guards.

Here we notice that if a complete deterministic FSM is reduced and strongly connected then a homing sequence always exists and length of a shortest homing sequence does not exceed $n(n-1)/2$ where n is the number of FSM states. Since in this case, according to propositions below, the FSM abstraction of an FSM with timed guards possesses the same features, the same holds for TFSMs.

Proposition 3. A complete deterministic FSM with timed guards S is strongly connected if and only if the FSM abstraction A_S is a strongly connected FSM.

Proposition 4. A complete deterministic FSM with timed guards S is state reduced if and only if the FSM abstraction A_S is a reduced FSM.

Based on Propositions 2-4, the following statement holds.

Proposition 5. Given a complete deterministic reduced and strongly connected FSM with timed guards S, a homing sequence always exists for TFSM S and length of a shortest homing sequence does not exceed $n(n-1)/2$ where n is the number of TFSM states.

Thus, a HS always exists for a complete reduced and strongly connected deterministic FSM with timed guards. Moreover, according to Proposition 4 and the results in [17], the upper bound of HS length cannot be reduced.

Note that for a non-initialized complete observable nondeterministic FSM length of a shortest homing sequence can reach $2^{n-1}-1$ [16] and thus, the following statement holds due to Proposition 2.

Proposition 6. Given a non-initialized complete observable nondeterministic FSM with timed guards S, length of a shortest homing sequence can reach $2^{n-1}-1$ where n is the number of TFSM states.

Similar to classical FSMs, a HS does not always exist for a nondeterministic observable FSM with timed guards. Checking the existence and deriving a HS for TFSMs can be performed by a slightly modified algorithm for FSMs [16].

Algorithm 1 for checking the existence and deriving a HS for an FSM with timed guards

Input: A complete non-initialized observable possibly nondeterministic FSM with timed guards $\mathsf{S} = (S, I, O, h_\mathsf{S})$

Output: Message ‘There is no HS for S‘ or a HS α for TFSM S

Step 1. Derive the FSM abstraction $\mathsf{A}_\mathsf{S} = (S, I_A, O, h_{\mathsf{AS}})$ for TFSM S.

Step 2. Construct a truncated successor tree for the FSM A_S. The root of the tree is labeled by the set of all pairs of different states while the nodes of the successor tree are labeled by sets of pairs of different states from S or empty set; edges of the tree are labeled by inputs. There exists an edge labeled by an input i from a node labeled by the set P at level j, $j \geq 0$, to a node at level j+1 labeled

by the set Q where a pair $\{s_1, s_2\} \in Q$ if and only if this pair is an i/o-successor of some pair of P. Given a node labeled by the set P at the level k, $k \geq 0$, the node is *terminal* if P is empty (**Rule 1**) or P contains a set R that labels a node at a level j, $j < k$ (**Rule-2**). If the successor tree has no nodes labeled with the empty set, then there is no HS for FSM $\mathsf{A_S}$. An input sequence α_{FSM} that labels a path with minimal length to a node labeled with the empty set is transformed into a corresponding timed input sequence α that is a shortest HS for S.

Nevertheless, the number of successors for each node of a truncated successor tree when checking the HS existence equals the number of inputs of the FSM abstraction, i.e., reaches $(2(B_S + 1)*|I|)$ instead of $|I|$ for classical FSMs. In the next section, we consider classes of TFSMs for which the number of the FSM abstraction inputs can be optimized when checking the existence and deriving a HS for a timed FSM.

4. Optimizing the number of the FSM abstraction inputs when checking the existence and deriving a HS for TFSMs

Here we notice that when constructing a HS for the FSM abstraction, the number of inputs becomes larger than that for the initial TFSM. In [11], approaches for the FSM abstraction optimization have been proposed in the context of test derivation procedures and in this section, we propose to use some of them for solving the HS problem for TFSMs.

Consider the FSM abstraction (Table 1) of the TFSM in Figure 1. As we can see it can well happen that there exist two equal rows for different inputs of the FSM abstraction. For example, the rows of the table for inputs $(i_1, [0, 0])$ and $(i_1, (0, 1))$ coincide and thus, it is sufficient to consider only one of them when looking for a HS.

Proposition 7. Given a complete FSM $\mathsf{S} = (S, I, O, h_S)$ and two inputs i and i', let for each state s it holds that $(s, i, o, s') \in h_S$ implies $(s, i', o, s') \in h_S$. The FSM S has a HS α if and only if a sequence α' obtained from α by the replacement of each input i' in α by i is a HS for the FSM $\mathsf{S}' = (S, I \setminus \{i'\}, O, h_{S'})$ where $h_{S'}$ is obtained from h_S by the removing transitions under input i'.

Proof. Let there exist HS $\alpha = i_1 \dots i_l$ of FSM S. Therefore, for each trace $\alpha/\gamma = i_1/o_1 \dots i_l/o_l$ the α/γ-successor of S is a singleton or does not exist. According to the statement conditions, for a sequence α' obtained from α by the replacement of each input i' in α by i it holds that the α'/γ-successor of S is a subset of the α/γ-successor of S, i.e., is a singleton or does not exist. Thus, α' is a HS for S'.

Corollary. The FSM S has a HS α of length l if and only if the FSM $\mathsf{S}' = (S, I \setminus \{i'\}, O, h_{S'})$ a HS α' of length l.

In other words, if for a pair of inputs the rows of the transition table coincide then the latter can be deleted from the FSM without losing a homing sequence if such a sequence exists, i.e., it is a way to minimize the number of FSM inputs when solving the HS problem. For example, a transition table for the input reduced form of the FSM abstraction in Table 1 has three rows colored in grey and, respectively, the number of successors for each node of a successor tree is three (Figure 2). By direct inspection, one can assure that there is a HS (i_1, (2, ∞)), (i_1, (2, ∞)), (i_1, (2, ∞)) of length 3.

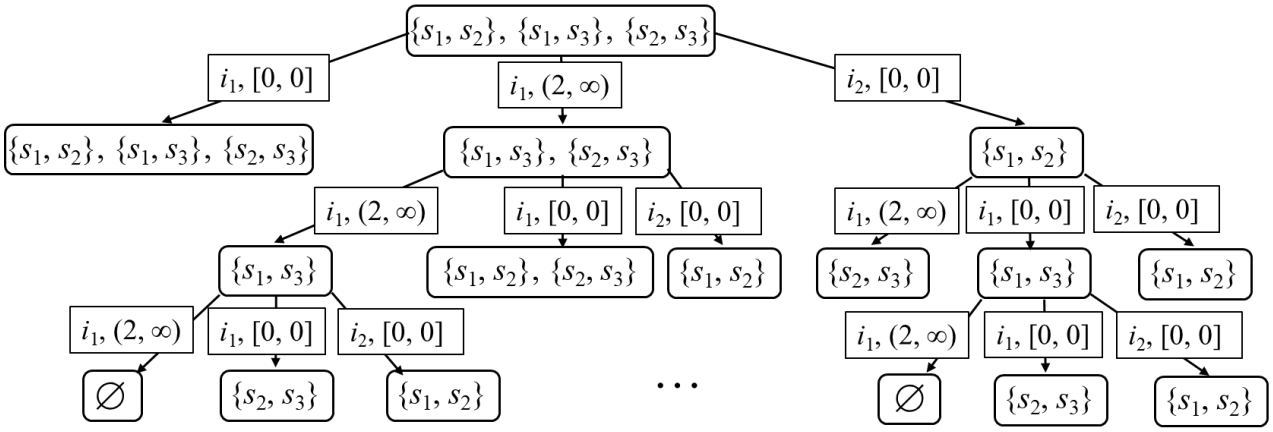

Fig. 2. The successor tree for the integer projection of FSM abstraction $\mathsf{A_S}$ (Table 1).

Let all the timed guards of a given TFSM S be closed on the left, i.e., each timed interval is of the form [*m*, *n*). In this case, the *integer projection* $\mathsf{A}^{int}_{\mathsf{S}}$= (*S*, I_A^{int}, *O*, h_{AS}) of the FSM abstraction where I_A^{int} = {(*i*, [*m*, *m*]) : *i* ∈ *I*, 0 ≤ *m* ≤ B_{S}} can be used for the HS derivation.

Proposition 8. Given a complete possibly nondeterministic TFSM S = (*S*, *I*, *O*, h_{S}) where each timed interval has the form [*m*, *n*), let $\mathsf{A}^{int}_{\mathsf{S}}$= (*S*, I_A^{int}, *O*, h^{int}_{AS}) where I_A^{int} = {(*i*, [*m*, *m*]) : *i* ∈ *I*, 0 ≤ *m* ≤ B_{S}} be the integer projection of the FSM abstraction. The FSM $\mathsf{A}^{int}_{\mathsf{S}}$ has a HS α_{FSM}' of length *l* if and only if TFSM S has a HS α of length *l*.

Proof. Let there exist a HS α of length *l* for FSM with timed guards S. By definition, α is a HS for TFSM S if and only if α_{FSM} is a HS for FSM abstraction $\mathsf{A_S}$. If all the timed guards of TFSM S are closed on the left, then for each input (*i*, (*a*, *b*)) of $\mathsf{A_S}$, there exists input (*i*, [*a*, *a*]) of $\mathsf{A}^{int}_{\mathsf{S}}$ such that (*s*, (*i*, (*a*, *b*)), *o*, *s*′) ∈ h_{AS} if and only if (*s*, (*i*, [*a*, *a*]), *o*, *s*′) ∈ h^{int}_{AS}. Respectively, by Proposition 7, the FSM abstraction $\mathsf{A_S}$ has a HS α_{FSM} of length *l* if and only if FSM $\mathsf{A}^{int}_{\mathsf{S}}$ has a HS α'_{FSM} of length *l*.

Thus, the above proposition claims that when deriving a HS for a machine with timed guards, in some situations, the number of inputs of the FSM abstraction can be twice minimized without losing a solution.

5. Conclusions

In this work, a method for deriving a homing sequence for an FSM with timed guards is proposed based on its FSM abstraction. We show that similar to classical FSMs, length of a shortest HS is polynomial with respect to the number of states for a deterministic reduced FSM with timed guards and can reach an exponential value for the nondeterministic case. However, the FSM abstraction has more inputs than the initial TFSM and for this reason, we discuss how the number of inputs of the FSM abstraction can be optimized when solving the HS problem.

Acknowledgements

This work is partly supported by RFBR Project No. 19-07-00327.

References

1. Bochmann, G., Petrenko A.: Protocol testing: review of methods and relevance for software testing. In Proc. of International Symposium on Software Testing and Analysis, Seattle, 1994, pp. 109–123.
2. Lee, D., Yannakakis, M.: Testing finite state machines: state identification and verification. IEEE Trans. on Computers 43(3), 1994, pp. 306–320.
3. Gill, A.: Introduction to the Theory of Finite-State Machines. McGraw-Hill, 1962.
4. Jourdan, G.-V., Ural, H., and Yenigun, H.: Reduced checking sequences using unreliable reset. Inf. Process. Lett. 115(5), 2015, pp. 532–535.
5. H. Ural, F. Zhang, and J.C. Zhang.: Effects of overlapping subsequences in constructing checking sequences. Journal of Advances in Information Sciences 1(1), 2013, pp. 59–73.
6. Kushik N., López J., Cavalli A., Yevtushenko N.: Improving Protocol Passive Testing through "Gedanken" Experiments with Finite State Machines. In Proceedings of QRS, 2016, pp. 315–322.
7. Hung-En Wang, Kuan-Hua Tu, Jie-Hong R. Jiang, Natalia Kushik: Homing Sequence Derivation with Quantified Boolean Satisfiability. Lecture Notes in Computer Science (LNCS), № 10533, 2017, pp. 230–242.
8. Yenigun, H. Yevtushenko, N. Kushik, N. López J.: The effect of partiality and adaptivity on the complexity of FSM state identification problems. Trudy ISP RAN/Proc. ISP RAS 30 (1), 2018, pp. 7–24.
9. Petrenko, A., Yevtushenko N.: Adaptive Testing of Deterministic Implementations Specified by Nondeterministic FSMs. LNCS № 7019, 2011, pp. 162–178.
10. Krichen M. and Tripakis S. Conformance testing for real-time systems. Formal Methods Syst. Des. 34 (3), 2009, pp. 238–304.

11. El-Fakih K., Yevtushenko N., and Fouchal H.: Testing timed finite state machines with guaranteed fault coverage. Proc. of the 21st IFIP WG 6.1 Int. Conf. on Testing of Software and Communication Systems and 9th Int. FATES Workshop, 2009, pp. 66–80.
12. Merayo M.G., Nunez M., and Rodriguez I.: Formal testing from timed finite state machines. Comput. Networks: Int. J. Comput. Telecom. Networking 52 (2), 2008, pp. 432–460.
13. Bresolin D., El-Fakih K., Villa T., and Yevtushenko N.: Deterministic timed finite state machines: Equivalence checking and expressive power. Int. Conf. GANDALF, 2014, pp. 203–216.
14. Zhigulin M., Yevtushenko N., Maag S., Cavalli A.: FSM-Based Test Derivation Strategies for Systems with Time-Outs. Int. Conf. On Quality Software, Madrid, 2011, pp. 141–150.
15. Gromov, M., El-Fakih, K., Shabaldina, N., and Yevtushenko, N.: Distinguishing non-deterministic timed finite state machines. In Formal Techniques for Distributed Systems. Springer Berlin Heidelberg. Lecture Notes in Computer Science 5522, 2009, pp.137–151.
16. Kushik, N., Yevtushenko, N.: On the Length of Homing Sequences for Nondeterministic Finite State Machines. Lecture Notes in Computer Science, № 7982, 2013, pp. 220–231.
17. Hibbard T. N.: Lest upper bounds on minimal terminal state experiments of two classes of sequential machines. Journal of the ACM 8(4), 1961, pp. 601–612.