UDC 519.7

On parallel composition of Finite State Machines with timed guards

Tvardovskii A.S. (Tomsk State University),

Yevtushenko N.V. (Ivannikov Institute for System Programming of the RAS)

Finite State Machines (FSMs) are widely used for analysis and synthesis of digital components of control systems. In order to take into account time aspects, timed FSMs are considered. In this paper, we address the problem of deriving a parallel composition of FSMs with timed guards and output delays (output timeouts). When the parallel composition is considered, component FSMs work in the dialog mode and the composition produces an external output when interaction between components is terminated. In this work, we formally introduce the parallel composition operator for FSMs with timed guards (TFSM) and show that unlike classical FSMs, a "slow environment" and the absence of live-locks are not enough for describing the behavior of a composition of deterministic TFSMs by a deterministic FSM with a single clock. Although the set of deterministic FSMs with timed guards is not closed under the parallel composition operator, some classes of deterministic TFSMs are still closed under this operator.

Keywords: *Timed Finite State Machines, output timeouts, timed guards, parallel composition*

1. Introduction

Finite State Machines (FSMs) are widely used for analysis and synthesis of discrete systems [1, 2], for example, for test derivation [3], verification [4] and optimization [1]. An FSM models a discrete event system that moves from state to state producing an output when an input is applied and is appropriate when reactive systems are considered. When the behavior of a complex system is described, a hierarchical approach is usually applied, i.e., a complex system is represented as a system of interacting components which usually are ‘simpler’ than the whole system. In the context of the FSM theory, a number of composition operators are considered and the notions of the parallel and synchronous compositions are defined [2]. A parallel composition describes the behavior of component machines which interact in a dialogue mode and is known to be appropriate when describing the joint functioning of protocol implementations and/or software components of telecommunication systems under the assumption of ‘one message in transit’. The parallel composition operator is well known for classical FSMs and the set of deterministic FSMs is closed under this operator when a so-called slow environment is considered. The slow environment means that the next input can be applied to the FSM composition only after the composition has produced an external output to the previous input.

Nowadays, a number of reactive systems take into account time aspects and thus, FSM based methods have to be extended to FSMs with time aspects such as timed guards, input and output timeouts [5, 6, 7]. A timed FSM can be considered as a good compromise between the expressive power and simplicity when considering timed automata [4, 8]. An FSM with timeouts can spontaneously change the current state if no input is applied before the input timeout expires. An FSM with time guards can have time restrictions at a state and thus, the behavior depends on a time instance when an external input is applied at the current state. Output timeouts in both cases describe the number of time units needed for processing an applied input.

In order to describe the joint behavior of communicating machines with time aspects, the composition operator has to be defined for the collection of Timed FSMs (TFSM). The parallel composition operator for FSMs with timeouts was proposed in [9] and is based on the intersection of corresponding underlying automata derived for TFSM components.

In this paper, we propose an algorithm for deriving a composed machine for a system of communicating FSMs with timed guards, namely, we define the parallel composition operator for a pair of such complete deterministic FSMs. It is shown that unlike classical FSMs, the set of deterministic FSMs with timed guards is not closed under the parallel composition operator in the context of slow environment assumption, i.e., the parallel composition of two deterministic TFSMs can have the nondeterministic behavior. Some examples can be found in [10]; however, in that paper, only the proof sketches were proposed and it was difficult to say whether more such situations can occur. We also consider some TFSM classes for which it is not the case, i.e., TFSM classes which are closed under the parallel composition. The definition of the parallel composition operator can also be used for checking whether there are live-locks in the composition of deterministic TFSMs, i.e., can be used for the verification of TFSM compositions.

The rest of the paper is organized as follows. Section 2 contains the preliminaries for classical and timed FSMs. In Section 3, the parallel composition operator is defined. In section 4, TFSM classes closed under the parallel composition are considered. Section 5 concludes the paper..

2. Preliminaries

Finite State Machines [1, 2] are widely used for modeling reactive systems that move from state to state under input stimuli and produce a predefined output response. Formally, an initialized FSM is a 5-tuple $\mathsf{S} = (S, s_0, I, O, h_S)$ where S is a finite non-empty set of states with the designated initial state s_0, I and O are input and output alphabets, and $h_S \subseteq (S \times I \times O \times S)$ is the transition (behavior) relation. A transition (s, i, o, s') describes the situation when FSM S moves from current state s to state s' when an input i is applied producing the output (response) o. The FSM S is *nondeterministic* if for some pair $(s, i) \in S \times I$, there exist several pairs $(o, s') \in O \times S$ such that $(s, i, o, s') \in h_S$; otherwise, the FSM is *deterministic*. The FSM S is *complete* if for each pair $(s, i) \in S \times I$ there exists $(o, s') \in O \times S$ such that $(s, i, o, s') \in h_S$; otherwise, the FSM is *partial*. FSM S is *observable* if 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 timed FSM can be annotated with a timed variable, timed guards and output delays [5, 6, 7]. In this paper, an initialized TFSM is a 5-tuple $\mathsf{S} = (S, s_0, I, O, \lambda_S)$ where S is a finite non-empty set of states with the designated initial state s_0, I and O are input and output alphabets, $\lambda_S \subseteq S \times I \times O \times S \times \Pi \times Z$ is the *transition relation* where the set Π is a set of *input timed guards* and Z is the set of output delays which are non-negative integers. An input timed guard $g \in \Pi$ describes the time domain when a transition can be executed and is given as an interval <*min*, *max*> from $[0; \infty)$, where $< \in \{(, [\}$, $> \in \{),]\}$. The transition $(s, i, o, s', g, d) \in S \times I \times O \times S \times \Pi \times Z$ 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 produced the last output; the clock then is set to zero and S produces output o after d time units. After producing the output the clock also is set to zero.

The TFSM S is a *complete* if the union of all input timed guards at any state s under every input i equals $[0; \infty)$. The TFSM S is a *deterministic* TFSM if for each two transitions $(s, i, o_1, s_1, g_1, d_1)$, $(s, i, o_2, s_2, g_2, d_2) \in h_S$, $s_1 \neq s_2$, $d_1 \neq d_2$ or $o_1 \neq o_2$, it holds that $g_1 \cap g_2 = \varnothing$, otherwise, TFSM S is *nondeterministic*. The TFSM S is *observable* if for every two transitions (s, i, o, s_1, g_1, d), $(s, i, o, s_2, g_2, d) \in \lambda_S$ such that $g_1 \cap g_2 \neq \varnothing$ it holds that $s_1 = s_2$. In this paper, we consider a system of interacting complete deterministic TFSM components and check whether the parallel composition can have live-locks and/or the nondeterministic behavior. 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. A timed output is a pair (o, d) where $o \in O$ and d is the output delay for producing the output o after an input was applied. A sequence of timed inputs $\alpha = (i_1, t_1) \ldots (i_n, t_n)$ is a *timed input sequence*, a sequence of timed outputs $\gamma = (o_1, d_1) \ldots (o_n, d_n)$ is a *timed output sequence*.

In this work, we also use the notion of a *Finite Automaton* that is a 4-tuple $\mathsf{S} = (S, s_0, A, h_S)$, where S is a finite non-empty set of states with the designated initial state s_0, A is a finite non-empty set of actions and $h_S \subseteq (S \times A \times S)$ is the transition (behavior) relation. A transition $(s, a,$

s′) describes the situation when automaton S moves from state *s* to state *s*′ under action *a*. A trace of automaton S at state *s* is a sequence of actions which takes S from state *s* to state *s*′.

3. Defining the parallel composition operator

In this section, we propose an algorithm for deriving the parallel composition of complete deterministic TFSMs. The parallel composition operator takes two TFSMs; the input (output) alphabet of each of them is divided into two subsets of external and internal inputs (outputs). We further assume that I_S and V (I_P and U) are external and internal input alphabets of component TFSM S (or P correspondingly) while O_S and U (O_P and V) are external and internal output alphabets of component TFSM S (or P correspondingly). When an external timed input is applied to the one of components, for example, to the component TFSM S, the component produces a corresponding output response that can be an internal or an external output after appropriate number of time units. If an external output is produced then the composition is ready to accept the next external input, while the value of a time variable of another component P is increased by the time that was needed for S to communicate with the environment. If S produces an internal output then the P processes it according to its description and produces an external output to the environment or an internal input that is applied to the S. The dialog holds until an external output is produced by one of the components. If the component TFSMs fall into the infinite dialogue then the composition is said to have a *live-lock* and the behavior of the composition is not defined for a corresponding input. As usual, we assume that the system of interaction TFSMs works in the slow environment, i.e., the next external input can be applied to the composition only when the latter has produced an external output to the previous external input.

Consider the composition of TFSMs S and P in Figure 1 and an external input sequence (i_1, 0.4).(i_2, 0.5) where i_1 and i_2 are external inputs of component TFSMs S and P, respectively. Input (i_1, 0.4) is applied to component TFSM S which produces the external output (o_{11}, 1) without any dialog with component TFSM P and resets the value of its timed variable to zero. At the same time, the value of the timed variable of component TFSM P is increased to 1.4, i.e., to the time value when component TFSM S was communicating with the environment, and reaches the value 1.9 when the next external input i_2 is applied. Correspondingly, the P produces an internal output (v, 2) to the component S and the latter produces an external output (o_{12}, 3). Therefore, for the next input (i_2, 0.5) the composition produces an external output (o_{12}, 5) where five is the sum of time units needed to execute both transitions. By direct inspection a reader can assure that if the second external input i_2 were applied at time 0.9 instead 0.5 then P would produce an external output (o_{22}, 1) without a dialog with component TFSM S.

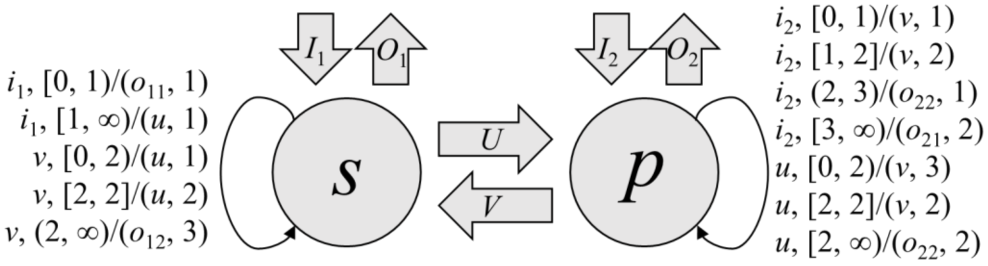

Fig. 1. Parallel composition of TFSMs S *and* P

When talking about the parallel composition over TFSMs, we would like to describe the behavior using the same model, i.e., the model of an FSM with timed guards. As a minimal timed guard has the duration one, the above example clearly shows that the parallel composition of deterministic TFSMs can have the nondeterministic behavior.

Another problem when describing the behavior of interacting deterministic FSMs which communicate in a dialogue mode, are live-locks when component FSMs fall into an infinite dialogue without producing an external output. Therefore, even under the assumption of the slow environment the parallel composition of two deterministic complete TFSMs can be partial and nondeterministic. Below we propose an algorithm for deriving the parallel composition of two FSMs with timed guards. In general case, this composition can have live-locks and the nondeterministic behavior; both features can be detected during the parallel composition construction.

Parallel composition operator:

Given a positive integer B, we use the following notations. The set G_B of intervals is the set $\{[a, a], (a, a + 1), [B, B], (B, \infty): a \in N$ and $a < B\}$, where N is the set of non-negative integers. Given two intervals $g, g' \in G_B$, we define the composition $g \circ g'$ of these intervals. If $g = [a, a]$ and $g' = [b, b]$ then $g \circ g' = [a + b, a + b]$; if $g = (a, b)$ and $g' = (a', b')$ then $g \circ g' = (a + a', b + b')$ assuming that $a + \infty = \infty$ for any integer a.

For each interval g, we specify a corresponding set $F_B(g)$ of G_B items. $F_B(g) = \{g' : g' \in G_B, g' \cap g \neq \varnothing\}$.

The behavior of a TFSM can be adequately described using a classical FSM that is called the *FSM abstraction* of the TFSM [11]. Given a complete observable possibly deterministic TFSM S = $(S, s_0, I, O, \lambda_S)$ with the largest finite boundary of timed guards B_S and the largest output delay D_S, the behavior of a corresponding FSM abstraction $\mathsf{A}_S(B) = (S, s_0, I_A, O_A, \lambda_{SA})$, where $B \geq B_S$, $I_A = I \times G_B$, $O_A = O \times \{0, 1, \ldots, D_S\}$, can be described as follows. There is a transition $(s, (i, g_i), (o, d), s') \in$

λ_{SA} if and only if there is a transition $(s, i, o, s', g, d) \in \lambda_S$ such that $g_i \subseteq g$. For deterministic TFSMs S and P in Figure 1 corresponding FSM abstractions are shown in Figure 2.

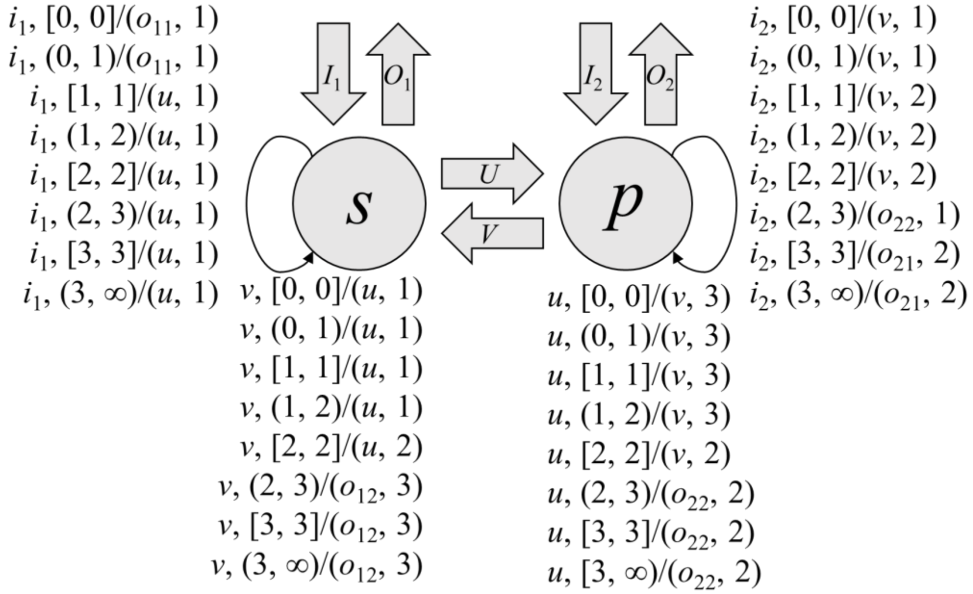

Fig. 2. FSM abstractions of TFSMs S and P

Algorithm for deriving the parallel composition of initialized deterministic complete FSMs with timed guards S and P

Input. Initialized deterministic complete FSMs with timed guards S = $(S, s_0, I_S = I_s \cup V, O_S = O_s \cup U, h_S)$ and P = $(P, p_0, I_P = I_p \cup U, O_P = O_s \cup V, h_P)$, where I_s and O_s (I_p and O_p) are external input and output alphabets of the component TFSMs S (P) while V and U (U and V) are internal input and output alphabets of the component FSM S (P); the alphabets are pairwise disjoint

Output. Parallel composition of S and P that is a FSM T with timed guards or one of messages «The system has a live-lock» or «The behavior of the system is nondeterministic»

Step 1. Derive the global automaton Q = $(Q, \{(i, g)/(r, d): i \in I_s \cup I_p, o \in O_s \cup O_p, g \in G_B, d \in \mathrm{Z}\}, q_0, \lambda_Q)$, where B is maximum of B_S and B_P and each state $q \in Q$ that is a *stable* state, is a 4-tuple $(s, g_s, p, g_p) \in S \times G_B \times P \times G_B$; each state $q \in Q$ that is an *unstable* state, is a 5-tuple $(s, g_s, p, g_p, (a, g))$ where $(s, g_s, p, g_p) \in S \times G_B \times P \times G_B$ and $a \in U \cup V$, $g \in G_B$. By definition, the initial state $q_0 = (s_0, [0, 0], p_0, [0, 0])$ is stable. A state is stable if and only if every incoming transition to a state is labelled by an action $(r, g)/(o, d)$ where o is the external output of S or P. At each stable state every transition under an action $(i, g)/(r, d)$, where i is the external input of S or P, is defined. At an unstable state that is a 5-tuple $(s, g_s, p, g_p, (a, g))$, the only defined transition is under an action $(a, g_s)/(o, d)$, $r \in O_S \cup O_P$.

Step 2. Derive FSM abstractions $A_S(B)$ and $A_P(B)$.

```
  For each external input $i \in I_s$, interval $g \in G_B$ and stable state $(s, [0,0], p, g_p)$ of the automaton Q and transition $(s, (i, g), (r, d), s')$ of the FSM abstraction $\mathsf{A}_S(B)$
    If $r$ is internal then add to Q transitions $((s, [0,0], p, g_p), (i, g)/(r, d), (s', [0,0], p, g_p', (r, [d, d]))$ where $g_p' \in F_B(g \circ g_p \circ [d, d])$;
    Else add to Q transitions $((s, [0,0], p, g_p), (i, g)/(r, d), (s', [0,0], p, g_p'))$, where $g_p' \in F_B(g \circ g_p \circ [d, d])$;
  For each external input $i \in I_p$, interval $g \in G_B$ and stable state $q = (s, g_s, p, [0,0])$ of the automaton Q and transition $(p, (i, g), (r, d), p')$ of the FSM abstraction $\mathsf{A}_P(B)$
    If $r$ is internal then add to Q transitions $((s, g_s, p, [0,0]), (i, g)/(r, d), (s, g_s', p', [0,0], (r, [d, d]))$ where $g_s' \in F_B(g \circ g_s \circ [d, d])$;
    Else add to Q transitions $((s, g_s, p, [0,0]), (i, g_i)/(r, d), (s, g_s', p', [0,0])$, where $g_s' \in F_B(g \circ g_s \circ [d, d]$;
  For each unstable state $(s, [0,0], p, g_p, (r, [d, d])$ of global automaton Q with an incoming transition labeled by $(i, g)/(r, [d, d])$ determine a transition $(p, (r, g_p), (m, k), p')$ of the FSM abstraction $\mathsf{A}_P(B)$;
    If $m$ is internal then add to Q a transition $((s, [0,0], p, g_p, (r, [d, d])), (r, [d, d])/(m, k), (s, [k, k], p', [0, 0], (m, [k, k]))$;
    Else add to Q a transition $((s, [0,0], p, g_p, (r, [d, d])), (r, [d, d])/(m, k), (s, [k, k], p', [0, 0]))$
  For each unstable state $(s, g_s, p, [0,0], (r, [d, d]))$ of global automaton Q with an incoming transition labeled by $(i, g)/(r, [d, d])$ determine a transition $(s, (r, g_s), (m, k), s')$ of the FSM abstraction $\mathsf{A}_P(B)$;
    If $m$ is internal then add to Q a transition $(s, g_s, p, [0,0], (r, [d, d])), (r, [d, d])/(m, k), (s', [0, 0], p, [k, k], (m, [k, k]))$
    Else add to Q a transition $((s, g_s, p, [0,0], (r, [d, d])), (r, [d, d])/(m, k), (s', [0, 0], p, [k, k]))$
Step 3. Derive the parallel composition of S and P in the form of FSM abstraction $\mathsf{A}_T$
  If the global automaton Q has a cycle where all transitions are labelled with internal actions then return the message «The composition has a live-lock»;
  Else Derive an FSM $\mathsf{A}_T$:
  -  states of $\mathsf{A}_T$ correspond to stable states of the global automaton Q;
  -  for each trace $\gamma = (i_1, g_1)/(o_1, d_1), \ldots, (i_n, g_n)/(o_n, d_n)$ which takes Q from a stable state $q_1$ to stable state $q_2$, add to $\mathsf{A}_T$ a transition $(q_1, (i_1, g_1), (o_n, d), q_2)$ where $d = d_1 + \ldots + d_n$ is the sum of all output delays in trace $\gamma$;
  -  derive the observable form of FSM $\mathsf{A}_T$ [12];
```

If A_T is nondeterministic **then** return the message «The composition behavior cannot be described by a deterministic TFSM».

Else the FSM abstraction A_T is translated into TFSM T.

Consider as an example of parallel composition of TFSMs S and P in Figure 1. A fragment of the global automaton derived at Step 2 of the above algorithm and a fragment of the corresponding FSM A_T are shown in Figure 3.

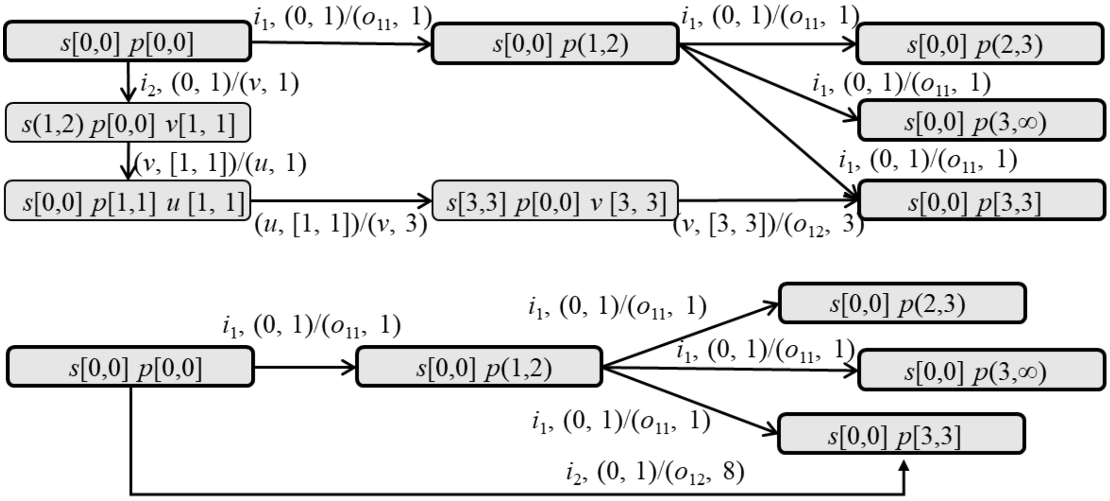

Fig. 3. A fragment of the global automaton Q for TFSMs S and P and a fragment of the corresponding FSM A_T

4. Selected TFSM classes closed under the parallel composition operator

When analyzing Algorithm 1, we conclude that the derived composed machine can be nondeterministic when at least one stable state of global automaton has interval (a, b). In this case, the response of TFSM components to input $(i, (a', b')$ cannot be uniquely specified. In this section, we consider special classes of TFSM composition for which the composition behavior is deterministic.

We first note that the global automaton Q and the corresponding composed machine are deterministic if $G_B = \{[0, 0], [1, 1], \ldots, [B, B], (B, \infty)\}$. Another case of deterministic behavior occurs when for each external input every component produces an internal output, as the automaton Q is then taken to an unstable state from which there the only deterministic transition is defined.

Below we formally specify deterministic TFSM classes which are closed under the paper composition operator. Differently from [10], the proofs to Propositions 1 and 2 are direct corollaries to the definition of the parallel composition operator.

Proposition 1 [10]. Let S and P are deterministic complete FSMs with timed guards. Parallel composition of TFSMs S and P is a deterministic TFSM if for each transition $(q, i, a, q', g, d) \in h_S \cup h_P$, where i is an external input, it holds that a is an internal output.

Proposition 2. Let S and P are deterministic complete FSMs with timed guards. Parallel composition of TFSMs S and P is deterministic TFSM if external inputs can be applied to TFSM composition only at integer time instances.

In both above cases, the value of the clock variable of each component is an integer when an external output is produced and $G_B = \{[0, 0], [1, 1], \ldots, [B, B], (B, \infty)\}$.

Proposition 3. Let S and P be deterministic complete FSMs with timed guards. The parallel composition of TFSMs S and P is a deterministic TFSM if the minimum output delay in TFSM S and P is bigger than the maximum of B_S and B_P.

In these cases, the value of the clock variable of each component is 0 or (B, ∞) when an external output is produced, respectively $G_B = \{[0, 0], (B, \infty)\}$.

Similar to parallel composition operator over classical FSMs [2], the parallel composition of FSMs with timed guards is a complete TFSM if the system of communicating TFSMs has no live-locks.

5. Conclusion

In this paper, we propose an algorithm for deriving a composed machine for a system of communicating FSMs with timed guards, namely, we define the parallel composition operator for a pair of complete deterministic FSMs with timed guards. Similar to classical finite state machines, we assume the slow environment, i.e., the next external input is applied to the system only when the system has produced an external output to the previous external input. However, differently from classical finite state machines, the set of deterministic TFSMs is not closed under the parallel composition operator. When deriving the parallel composition of two communicating FSMs with timed guards, we also detect live-locks in the system (if any) and the nondeterministic behavior if the latter occurs. We also specify some TFSM classes such that there is no nondeterministic behavior when TFSM components of these classes are composed.

Acknowledgement

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

References

1. Villa T., Kam T., Brayton R.K., Sandgiovanni-Vincentelli A. Synthesis of Finite State machines: Logic Optimization, 1997. 520 p.
2. Villa T., Yevtushenko N., Brayton R.K., Mishchenko A., Petrenko A., Sangiovanni-Vincentelli A. The Unknown Component Problem. Theory and Applications. Springer, 2012. 312 p.
3. Chow T.S., Test design modeled by finite-state machines // IEEE Trans. Software Eng. 1978. Vol. 4. No. 3. P. 178–187.
4. Alur R., Dill D.L. A theory of timed automata // Theoretical Computer Science. 1994. Vol. 126. P. 183–235.
5. Krichen M., Tripakis S., Conformance testing for real-time systems // Formal Methods Syst. Des. 2009. Vol. 34. No. 3, P. 238–304.
6. Merayo M.G., Nunez M., Rodriguez, I. Formal testing from timed finite state machines // Comput. Networks: Int. J. Comput. Telecommun. Networking. 2008. Vol. 52. No. 2. P. 432–460.
7. Bresolin D., El-Fakih K., Villa T., Yevtushenko N. Deterministic timed finite state machines: Equivalence checking and expressive power // Int. Conf. GANDALF. 2014. P. 203–216.
8. Springintveld J., Vaandrager F., D’Argenio P. Testing timed automata // Theor. Comput. Sci. 2001. Vol. 254. No. 1–2, P. 225–257.
9. Kondratyeva O., Yevtushenko N., Cavalli A. Solving parallel equations for Finite State Machines with Timeouts // Trudy ISP RAN/Proc. ISP RAS,2014. Vol. 26, No. 6. P. 85–98.
10. Tvardovskii A.S., Laputenko A.V. On the possibilities of FSM description of parallel composition of timed Finite State Machines // Trudy ISP RAN/Proc. ISP RAS, 2018. Vol. 30. No. 1. P. 25-40 (in Russian). DOI: 10.15514/ISPRAS-2018-30(1)-2
11. Bresolin D., Tvardovskii A., Yevtushenko N., Villa T., Gromov M. Minimizing Deterministic Timed Finite State Machines // IFAC-PapersOnline. 2018. Vol. 51. No. 7. P. 486–492.
12. Starke P. Abstract Automata / P. Starke // American Elsevier, 1972. 419 p.