

# Deterministic Timed Finite State Machines: Equivalence Checking and Expressive Power

Davide Bresolin

University of Bologna  
Bologna, Italy

davide.bresolin@unibo.it

Khaled El-Fakih

American University of Sharjah  
Sharjah, United Arab Emirates

kelfakih@aus.edu

Tiziano Villa

University of Verona  
Verona, Italy

tiziano.villa@univr.it

Nina Yevtushenko

Tomsk State University  
Tomsk, Russia

yevtushenko@sibmail.com

There has been a growing interest in defining models of automata enriched with time. For instance, timed automata were introduced as automata extended with clocks. In this paper, we study models of timed finite state machines (TFSMs), i.e., FSMs enriched with time, which accept timed input words and generate timed output words. Here we discuss some models of TFSMs with a single clock: TFSMs with timed guards, TFSMs with timeouts, and TFSMs with both timed guards and timeouts. We solve the problem of equivalence checking for all three models, and we compare their expressive power, characterizing subclasses of TFSMs with timed guards and of TFSMs with timeouts that are equivalent to each other.

## 1 Introduction

Finite automata (FA) and finite state machines (FSMs) are formal models widely used in the practice of engineering and science, e.g., in application domains ranging from sequential circuits, communication protocols, embedded and reactive systems, to biological modelling.

Since the 90s, the standard classes of FA have been enriched with the introduction of time constraints to represent more accurately the behaviour of systems in discrete or continuous time. Timed automata (TA) are such an example: they are finite automata augmented with a number of resettable real-time clocks, whose transitions are triggered by predicates involving clock values [1].

More recently, timed models of FSMs have been proposed in the literature by the introduction of time constraints such as timed guards or timeouts. Timed guards restrict the input/output transitions to happen within given time intervals. In particular, the timed FSM proposed in [5, 3, 4] features: one clock variable, time constraints to limit the time elapsed at a state, and clock reset when a transition is executed.

The timed FSM proposed in [7, 6] features: one clock variable, time constraints to limit the time elapsed when an output has to be produced after an input has been applied to the FSM, clock reset when an output is produced, timeouts. The meaning of timeouts is the following: if no input is applied at a current state for some timeout period, the timed FSM moves from the current state to another state using a timeout function; e.g., timeouts are common in telecommunication protocols and systems.

TA and TFSMs are also used when deriving tests for discrete event systems. However, methods for deriving complete finite test suites with a guaranteed fault coverage exist only for TFSMs, therefore TFSMs are preferred over TA and other models, when the derivation of complete tests is required.

In this paper, we investigate some models of TFSMs with a single clock: TFSMs with only timed guards, TFSMs with only timeouts, and TFSMs with both timed guards and timeouts. We solve the

problem of equivalence checking for all three models, we compare their expressive power, and we characterize subclasses of TFSMs with timed guards and of TFSMs with timeouts that are equivalent to each other. These results are obtained by introducing relations of bisimulation that define untimed finite state machines whose states include information on the clock regions. This is reminiscent of the region graph construction used to prove that in timed automata the verification questions (e.g., expressed by safety properties) have the same answer for all the clock valuations in the same clock region [1]. In our case, we are able to prove a stronger result: the timed behaviours of two timed FSMs are equivalent if and only if the behaviours of the companion untimed FSMs are equivalent. So our models of timed FSMs strike a good balance between expressivity and computational complexity.

## 2 Timed FSM Models

Let  $A$  be a finite alphabet, and let  $\mathbb{R}^+$  be the set of non-negative reals. A *timed symbol* is a pair  $(a, t)$  where  $t \in \mathbb{R}^+$  is called the *timestamp* of the symbol  $a \in A$ . A timed word is then defined as a finite sequence  $(a_1, t_1)(a_2, t_2)(a_3, t_3) \dots$  of timed symbols where the sequence of timestamps  $t_1 t_2 t_3 \dots$  is increasing. All the timed models considered in this paper are input/output machines that operate by reading a *timed input word*  $(i_1, t_1)(i_2, t_2) \dots (i_k, t_k)$  defined on some *input alphabet*  $I$ , and producing a corresponding *timed output word*  $(o_1, t_1)(o_2, t_2) \dots (o_k, t_k)$  on some *output alphabet*  $O$ . The production of outputs is assumed to be instantaneous: the timestamp of the  $j$ -th output  $o_j$  is the same of the  $j$ -th input  $i_j$ . Models where there is a delay between reading an input and producing the related output are possible but not considered in this paper. Given a timed word  $(a_1, t_1)(a_2, t_2) \dots (a_k, t_k)$ ,  $\text{Untime}((a_1, t_1)(a_2, t_2) \dots (a_k, t_k)) = a_1 a_2 \dots a_k$  denotes the word obtained when deleting the timestamps.

A timed possibly non-deterministic and partial FSM (TFSM) is an FSM augmented with a clock. The clock is a real number that measures the time delay at a state, and its value is reset to zero when a transition is executed. In this section we first introduce the TFSM model with timed guards given in [3, 5] and the TFSM model with timeouts given in [7, 9]. Then, we define a TFSM model with both timed guards and timeouts that subsumes the other two. In addition, we study the equivalence problem for each of the three TFSM models.

### 2.1 TFSM with timed guards

A timed guard defines the time interval when a transition can be executed. Intuitively, a TFSM in the present state  $s$  and accepting input  $i$  at a time  $t$  satisfying the timed guard responds with output  $o$  and moves to the next state  $s'$ , while the clock is reset to 0 and restarts advancing in state  $s'$ .

**Definition 1** (TFSM with Timed Guards [3, 5]). *A TFSM with timed guards is a tuple  $M = (S, I, O, \lambda_S, s_0)$  where  $S$ ,  $I$ , and  $O$  are finite disjoint non-empty sets of states, inputs and outputs, respectively,  $s_0$  is the initial state,  $\lambda_S \subseteq S \times I \times \Pi \times O \times S$  is a transition relation where  $\Pi$  is the set of input timed guards. Each guard in  $\Pi$  is an interval  $g = \langle t_{\min}, t_{\max} \rangle$  where  $t_{\min}$  is a nonnegative integer, while  $t_{\max}$  is either a nonnegative integer or  $\infty$ ,  $t_{\min} \leq t_{\max}$ , and  $\langle \in \{ (, [ \}$  while  $\rangle \in \{ ), ] \}$ .*

The *timed state* of a TFSM is a pair  $(s, x)$  such that  $s \in S$  is a state of  $M$  and  $x \in \mathbb{R}^+$  is the current value of the clock. *Transitions* between timed states can be of two types:

- *timed transitions* of the form  $(s, x) \xrightarrow{t} (s, x + t)$  where  $t \in \mathbb{R}^+$ , representing the fact that a delay of  $t$  time units has elapsed without receiving any input;

- *input/output transitions* of the form  $(s, x) \xrightarrow{i, o} (s', 0)$ , representing reception of the input symbol  $i \in I$ , production of the output  $o \in O$  and reset of the clock. An input/output transition can be activated only if there exists a tuple  $(s, i, \langle t_{min}, t_{max} \rangle, o, s') \in \lambda_S$  such that  $x \in \langle t_{min}, t_{max} \rangle$ .

A *timed run* of a TFSM with timed guards  $M$  interleaves timed transitions with input/output transitions. Given a timed input word  $v = (i_1, t_1)(i_2, t_2) \dots (i_k, t_k)$ , a timed run of  $M$  over  $v$  is a finite sequence  $\rho = (s_0, 0) \xrightarrow{t_1} (s_0, t_1) \xrightarrow{i_1, o_1} (s_1, 0) \xrightarrow{t_2 - t_1} (s_1, t_2 - t_1) \xrightarrow{i_2, o_2} (s_2, 0) \xrightarrow{t_3 - t_2} \dots \xrightarrow{i_k, o_k} (s_k, 0)$  such that  $s_0$  is the initial state of  $M$ , and for every  $j \geq 0$   $(s_j, 0) \xrightarrow{t_{j+1} - t_j} (s_j, t_{j+1} - t_j) \xrightarrow{i_{j+1}, o_{j+1}} (s_{j+1}, 0)$  is a valid sequence of transitions of  $M$ . The timed run  $\rho$  is said to *accept* the timed input word  $v = (i_1, t_1)(i_2, t_2) \dots (i_k, t_k)$  and to *produce* the timed output word  $u = (o_1, t_1)(o_2, t_2) \dots (o_k, t_k)$ . The behavior of  $M$  is defined in terms of the input/output words accepted and produced by the machine.

**Definition 2.** *The behavior of a TFSM  $M$  is a partial mapping  $B_M : (I \times \mathbb{R})^* \mapsto 2^{(O \times \mathbb{R})^*}$  that associates every input word  $w$  accepted by  $M$  with the set of output words  $B_M(w)$  produced by  $M$  under input  $w$ . When  $M$  is an untimed FSM the behavior is defined as a partial mapping  $B_M : I^* \mapsto 2^{O^*}$ .*

*Two machines  $M$  and  $M'$  with the same input and output alphabets are equivalent if and only if they have same behavior, i.e.  $B_M = B_{M'}$ .*

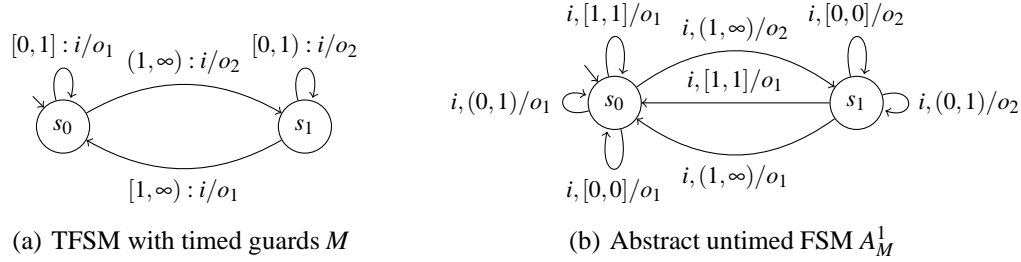
**Complete and deterministic machines.** The usual definitions for FSMs of deterministic and non-deterministic, submachine, etc., can be extended to all timed FSMs models considered here. In particular, a TFSM is *complete* if for each state  $s$ , input  $i$  and value of the clock  $x$  there exists at least one transition  $(s, x) \xrightarrow{i, o} (s', 0)$ , otherwise the machine is *partial*. A TFSM is *deterministic* if for each state  $s$ , input  $i$  and value of the clock  $x$  there exists at most one input/output transition, otherwise is *non-deterministic*.

For the sake of simplicity, from now on we consider only *complete and deterministic* machines, leaving the treatment of partial and non-deterministic TFSM to future work. When a machine  $M$  is deterministic and complete, we have that  $B_M(w)$  is a singleton set for every input word  $w$ . Hence, we can redefine the behavior  $B_M$  as a function  $B_M : (I \times \mathbb{R})^* \mapsto (O \times \mathbb{R})^*$  that associates every input word  $w = (i_1, t_1)(i_2, t_2) \dots (o_k, t_k)$  with the unique output word  $B_M(w)$  produced by  $M$  under input  $w$ .

Moreover, we can consider the transition relation of the machine as a complete function  $\lambda_S : S \times I \times \mathbb{R}^+ \mapsto S \times O$  that takes as input the current state  $s$ , the delay  $t$  and the input symbol  $i$  and produces the (unique) next state and output symbol  $\lambda_S(s, t, i) = (s', o)$  such that  $(s, 0) \xrightarrow{t} (s, t) \xrightarrow{i, o} (s', 0)$ . With a slight abuse of the notation, we can extend it to a function  $\lambda_S : S \times (I \times \mathbb{R}^+)^* \mapsto S \times O^*$  that takes as inputs the initial state  $s$  and a timed word  $w$ , and returns the state reached by the machine after reading  $w$  and the generated output word. We will use  $s \xrightarrow{w, u} s'$  as a shorthand for  $\lambda_S(s, w) = (s', u)$ .

**Equivalence checking of TFSM with timed guards.** In this section we show how to solve the equivalence problem of TFSM with guards by reducing it to the equivalence problem of untimed FSM. We proceed in three steps: first, we show how to build an “abstract” FSM from a TFSM with guards; then we define an appropriate notion of bisimulation to compare TFSM with guards with untimed FSM; finally, from the properties of the bisimulation relation, we conclude that two TFSM with guards are equivalent if and only if their abstractions are equivalent.

Now, let  $M$  be a TFSM with guards. We define  $\max(M)$  as the greatest integer constant (different from  $\infty$ ) appearing in the guards of  $\lambda_S$ . For any natural number  $N \geq \max(M)$ , we define  $\mathbb{I}_N$  as the set of intervals  $\mathbb{I}_N = \{[n, n] \mid n \leq N\} \cup \{(n, n+1) \mid 0 \leq n < N\} \cup \{(N, \infty)\}$ . The discrete abstraction of a TFSM with guards will take as inputs pairs of the form  $(i, \langle n, n' \rangle)$  where  $i$  is the actual input and  $\langle n, n' \rangle \in \mathbb{I}_N$  an interval representing the time delay.

Figure 1:  $\mathbb{I}_N$ -abstraction of TFSM with timed guards.

**Definition 3.** Given a TFSM with timed guards  $M = (S, I, O, \lambda_S, s_0)$  and a natural number  $N \geq \max(M)$ , we define the abstract FSM  $A_M^N = (S, I \times \mathbb{I}_N, O, \lambda_A, s_0)$  as the untimed FSM such that  $(s, (i, \langle n, n' \rangle), o, s') \in \lambda_A$  if and only if  $(s, i, \langle t, t' \rangle, o, s') \in \lambda_S$  for some guard  $\langle t, t' \rangle$  such that  $\langle n, n' \rangle \subseteq \langle t, t' \rangle$ .

Figure 1 shows an example of a simple TFSM with timed guards and of the corresponding  $\mathbb{I}_N$ -abstraction (for  $N = 1$ ). We cannot directly compare the behavior of a timed FSM with the behavior of its untimed abstraction, since the former accepts timed input words on  $I$  and the latter accepts untimed input words on  $I \times \mathbb{I}_N$ . For this reason, we need to introduce the notion of abstraction of a timed word.

**Definition 4.** Given a finite alphabet  $A$ , a finite timed word  $v = (a_1, t_1) (a_2, t_2) (a_3, t_3) \dots (a_m, t_m)$ , an integer  $N$  and the set of intervals  $\mathbb{I}_N$ , we define its  $\mathbb{I}_N$ -abstraction as the finite word  $\mathbb{I}_N(v) = (a_1, \langle n_1, n'_1 \rangle) (a_2, \langle n_2, n'_2 \rangle) (a_3, \langle n_3, n'_3 \rangle) \dots (a_m, \langle n_m, n'_m \rangle)$  such that  $t_i - t_{i-1} \in \langle n_i, n'_i \rangle$  for every  $1 \leq i \leq m$ .

$\mathbb{I}_N$ -bisimulation connects states of a timed FSM with states of an untimed FSM.

**Definition 5.** Given a TFSM with timed guards  $T = (S, I, O, \lambda_S, s_0)$ , an untimed FSM  $U = (R, I \times \mathbb{I}_N, O, \lambda_R, r_0)$ , an integer  $N \geq \max(T)$  and the set of intervals  $\mathbb{I}_N$ , an  $\mathbb{I}_N$ -bisimulation is a relation  $\sim \subseteq S \times R$  that respects the following conditions:

1. for every pair of states  $s \in S$  and  $r \in R$  such that  $s \sim r$ , if  $(s, i, \langle t, t' \rangle, o, s') \in \lambda_S$ , then for every  $\langle n, n' \rangle \in \mathbb{I}_N$  such that  $\langle n, n' \rangle \subseteq \langle t, t' \rangle$  there exists a transition  $(r, (i, \langle n, n' \rangle), o, r') \in \lambda_R$  such that  $r' \sim s'$ ;
2. for every pair of states  $s \in S$  and  $r \in R$  such that  $s \sim r$ , if  $(r, (i, \langle n, n' \rangle), o, r') \in \lambda_R$  then there exists a transition  $(s, i, \langle t, t' \rangle, o, s') \in \lambda_S$  such that  $\langle n, n' \rangle \subseteq \langle t, t' \rangle$  and  $r' \sim s'$ .

$T$  and  $U$  are  $\mathbb{I}_N$ -bisimilar if there exists an  $\mathbb{I}_N$ -bisimulation  $\sim \subseteq S \times R$  such that  $s_0 \sim r_0$ .

$\mathbb{I}_N$ -bisimilar machines have the same behavior, as formally proved by the following lemma.

**Lemma 1.** Given a TFSM with timed guards  $T = (S, I, O, \lambda_S, s_0)$  and an untimed FSM  $U = (R, I \times \mathbb{I}_N, O, \lambda_R, r_0)$ , if there exists an  $\mathbb{I}_N$ -bisimulation  $\sim$  such that  $s_0 \sim r_0$  then for every timed input word  $v = (i_1, t_1) (i_2, t_2) (i_3, t_3) \dots (i_m, t_m)$  we have that  $\text{Untime}(B_T(v)) = B_U(\mathbb{I}_N(v))$ .

*Proof.* We prove the lemma by showing that “for every pair of states  $s \sim r$  and timed word  $v$ ,  $\lambda_S(s, v) = (s', w)$  if and only if  $\lambda_R(r, \mathbb{I}_N(v)) = (r', \text{Untime}(w))$  with  $s' \sim r'$ ”.

We prove the claim by induction on the length  $m$  of the input word  $v$ . Suppose  $m = 1, v = (i_1, t_1)$  and  $\lambda_S(s, (i_1, t_1)) = (s_1, (o_1, t_1)) = (s_1, w)$ . By the definition of TFSM with timed guards, we have that there exists a transition  $(s, i, \langle t, t' \rangle, o_1, s_1) \in \lambda_S$  such that  $t_1 \in \langle t, t' \rangle$ . Since  $s \sim r$ , by the definition of  $\mathbb{I}_N$ -bisimulation we have that there exists a transition  $(r, (i, \langle n_1, n'_1 \rangle), o_1, r_1) \in \lambda_R$  with  $t_1 \in \langle n_1, n'_1 \rangle$  and  $s_1 \sim r_1$ . Since  $\mathbb{I}_N(v) = (i_1, \langle n_1, n'_1 \rangle)$ , we have that  $\lambda_R(r, (i_1, \langle n_1, n'_1 \rangle)) = (r_1, o_1)$  and the claim is proved.

Now, suppose the claim holds for all natural numbers up to  $m - 1$ , and let  $v = (i_1, t_1) \dots (i_{m-1}, t_{m-1}) (i_m, t_m) = v' (i_m, t_m)$ . Suppose that  $\lambda_S(s, v') = (s_{m-1}, w')$  and that  $\lambda_S(s_{m-1}, (i_m, t_m - t_{m-1})) = (s_m, (o_m, t_m -$

$t_{m-1}$ ): by inductive hypothesis, we have that  $\lambda_R(r, \mathbb{I}_N(v')) = (r_{m-1}, \text{Untime}(w'))$  and that  $\lambda_R(r_{m-1}, \mathbb{I}_N(i_m, t_m - t_{m-1})) = (r_m, o_m)$  with  $s_{m-1} \sim r_{m-1}$  and  $s_m \sim r_m$ . This implies that  $\lambda_S(s, v) = \lambda_S(s, v'(i_m, t_m)) = (s_m, w'(o_m, t_m)) = (s_m, w)$  and  $\lambda_R(r, \mathbb{I}_N(v)) = \lambda_R(r, \mathbb{I}_N(v'(i_m, t_m))) = (r_m, \text{Untime}(w')o_m) = (r_m, \text{Untime}(w))$  with  $s_m \sim r_m$ , and thus that the claim holds also for  $m$ .

To conclude the proof of the Lemma it is sufficient to recall that from the definition of behaviour we have that  $B_T(v) = w$  if and only if  $\lambda_S(s_0, v) = (s_m, w)$  for some state  $s_m \in S$ . From  $s_0 \sim r_0$  we can conclude that  $\lambda_R(r_0, \mathbb{I}_N(v)) = (r_m, \text{Untime}(w))$  and thus that  $B_U(\mathbb{I}_N(v)) = \text{Untime}(w) = \text{Untime}(B_T(v))$ .  $\square$

**Lemma 2.** *A TFSM with timed guards  $M$  is  $\mathbb{I}_N$ -bisimilar to the abstract FSM  $A_M^N$ .*

*Proof.* The identity relation on  $S$  is an  $\mathbb{I}_N$ -bisimulation for  $M$  and  $A_M^N$ .  $\square$

**Theorem 1.** *Let  $M$  and  $M'$  be two TFSM with timed guards and let  $N$  be such that  $N \geq \max(M)$  and  $N \geq \max(M')$ . Then  $M$  and  $M'$  are equivalent if and only if the two abstract FSM  $A_M^N$  and  $A_{M'}^N$  are equivalent.*

*Proof.* By Lemma 2 we have that  $M$  is  $\mathbb{I}_N$ -bisimilar to  $A_M^N$  and that  $M'$  is  $\mathbb{I}_N$ -bisimilar to  $A_{M'}^N$ . We first show that if  $M$  and  $M'$  are equivalent, then  $A_M^N$  and  $A_{M'}^N$  are equivalent. Under the assumption that  $M$  and  $M'$  are complete and deterministic, we have that  $M$  and  $M'$  are equivalent if for every timed input word  $v$ ,  $B_M(v) = B_{M'}(v)$ . By Lemma 1 we have that  $\text{Untime}(B_M(v)) = B_{A_M^N}(\mathbb{I}(v))$  and that  $\text{Untime}(B_{M'}(v)) = B_{A_{M'}^N}(\mathbb{I}(v))$ . This implies that  $B_{A_M^N}(\mathbb{I}(v)) = B_{A_{M'}^N}(\mathbb{I}(v))$ . Since for every untimed word  $w$  over the alphabet  $I \times \mathbb{I}_N$  it is possible to find a timed word  $v$  over the alphabet  $I$  such that  $w = \mathbb{I}_N(v)$ , we can conclude that  $A_M^N$  and  $A_{M'}^N$  are equivalent.

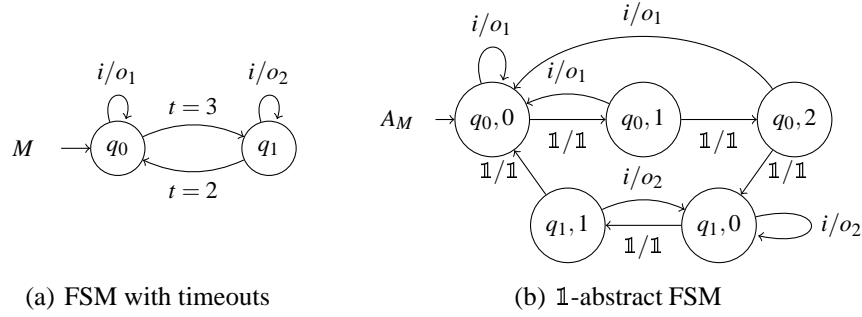
For the converse implication, suppose by contradiction that  $A_M^N$  and  $A_{M'}^N$  are equivalent but  $M$  and  $M'$  are not. This means that there exists a timed input word  $v = (i_1, t_1) \dots (i_m, t_m)$  such that  $B_M(v) \neq B_{M'}(v)$ . Let  $B_M(v) = w = (o_1, t_1) \dots (o_m, t_m)$  and  $B_{M'}(v) = w' = (o'_1, t_1) \dots (o'_m, t_m)$ . Notice that, by the definition of TFSM with timed guards,  $w$  and  $w'$  must have the same timestamps (outputs are produced instantaneously), and thus they must differ on the produced output symbols. This implies that there exists at least one index  $1 \leq j \leq m$  such that  $o_j \neq o'_j$  and hence that  $\text{Untime}(w) \neq \text{Untime}(w')$ . By Lemma 1 we have that  $B_{A_M^N}(\mathbb{I}(v)) = \text{Untime}(w)$  and that  $B_{A_{M'}^N}(\mathbb{I}(v)) = \text{Untime}(w')$ . Hence,  $B_{A_M^N}(\mathbb{I}(v)) \neq B_{A_{M'}^N}(\mathbb{I}(v))$ , in contradiction with the hypothesis that  $A_M^N$  and  $A_{M'}^N$  are equivalent.  $\square$

## 2.2 TFSM with timeouts

**Definition 6** (TFSM with Timeouts [7, 9]). *A TFSM with timeouts is a 6-tuple  $M = (S, I, O, \lambda_S, s_0, \Delta_S)$  where  $S$ ,  $I$ , and  $O$  are finite disjoint non-empty sets of states, inputs and outputs, respectively,  $s_0$  is the initial state,  $\lambda_S \subseteq S \times I \times O \times S$  is a transition relation and  $\Delta_S : S \rightarrow S \times (\mathbb{N}_{\geq 1} \cup \{\infty\})$  is a timeout function.*

Transitions of TFSM with timeouts can be triggered not only by the reception of an input, but also by timeouts. When the machine enters a state  $s$  it resets the clock to 0. If an input  $i$  is received before the timeout  $\Delta_S(s)_{\downarrow \mathbb{N}}$  expires and a transition  $(s, i, o, s') \in S \times I \times O \times S$  exists, then the machine produces  $o$ , moves to state  $s'$  while resetting the clock at  $s'$  to 0. If no input is received before the timeout  $\Delta_S(s)_{\downarrow \mathbb{N}}$  expires, then the TFSM will move to the state specified by the timeout function  $\Delta_S(s)_{\downarrow S}$  and reset the clock to 0. If  $\Delta_S(s)_{\downarrow \mathbb{N}} = \infty$ , then the machine can stay at state  $s$  infinitely long waiting for an input.

A timed state of a TFSM with timeouts is a pair  $(s, x) \in S \times \mathbb{R}^+$  with the additional constraint that  $x < \Delta_S(s)_{\downarrow \mathbb{N}}$  (the value of the clock cannot exceed the timeout). Timed and input/output transitions are defined as follows.

Figure 2: Example of  $\mathbb{1}$ -abstraction of a TFSM with timeouts.

- The *timed transition* relation  $\xrightarrow{t}$  is the smallest relation closed under the following properties:
  - for every timed state  $(s, x)$  and delay  $t \geq 0$ , if  $x + t < \Delta_S(s)_{\downarrow \mathbb{N}}$ , then  $(s, x) \xrightarrow{t} (s, x + t)$ ;
  - for every timed state  $(s, x)$  and delay  $t \geq 0$ , if  $x + t = \Delta_S(s)_{\downarrow \mathbb{N}}$ , then  $(s, x) \xrightarrow{t} (s', 0)$  with  $s' = \Delta_S(s)_{\downarrow S}$ ;
  - if  $(s, x) \xrightarrow{t_1} (s', x')$  and  $(s', x') \xrightarrow{t_2} (s'', x'')$  then  $(s, x) \xrightarrow{t_1+t_2} (s'', x'')$ .
- The *input/output transition* relation  $\xrightarrow{i,o}$  is such that  $(s, x) \xrightarrow{i,o} (s', 0)$  if and only if  $(s, i, o, s') \in \lambda_S$  and  $x < \Delta_S(s)_{\downarrow \mathbb{N}}$ .

The definition of timed run, behavior, complete and deterministic machine given in Section 2.1 for TFSM with timed guards can be extended to TFSM with timeouts.

**Equivalence checking of TFSM with timeouts.** We solve the equivalence problem for TFSM with timeouts using the same approach we used in Section 2.1 for TFSM with timed guards: we reduce the problem to the equivalence of standard FSM by an appropriate notion of “abstract” untimed FSM.

In the case of a TFSM with timeouts  $M$ , the constant  $\max(M)$  is defined as the greatest timeout value of the function  $\Delta_S$  different from  $\infty$ . States of the abstract FSM will be pairs  $(s, n)$  where  $s$  is a state of  $M$  and  $n$  is a natural number ranging from 0 to  $\max(M) - 1$  abstracting the clock value. Transitions can be either standard input/output transitions labelled with pairs from  $I \times O$  or “time elapsing” transitions labelled with the special pair  $(\mathbb{1}, \mathbb{1})$  representing a one time-unit delay without inputs (see also [9]).

**Definition 7.** Given a TFSM with timeouts  $M = (S, I, O, \lambda_S, s_0, \Delta_S)$ , let  $N = \max(M) - 1$ . We define the  $\mathbb{1}$ -abstract FSM  $A_M = (S \times \{0, \dots, N\}, I \cup \{\mathbb{1}\}, O \cup \{\mathbb{1}\}, \lambda_A, (s_0, 0))$  as the untimed FSM such that:

- $(s, n) \xrightarrow{\mathbb{1}, \mathbb{1}} (s, n + 1)$  if and only if  $n + 1 < \Delta_S(s)_{\downarrow \mathbb{N}} < \infty$ ;
- $(s, n) \xrightarrow{\mathbb{1}, \mathbb{1}} (s', 0)$  if and only if  $\Delta_S(s) = (s', n + 1)$ ;
- $(s, 0) \xrightarrow{\mathbb{1}, \mathbb{1}} (s, 0)$  if and only if  $\Delta_S(s)_{\downarrow \mathbb{N}} = \infty$ ;
- $(s, n) \xrightarrow{i,o} (s', 0)$  if and only if  $(s, i, o, s') \in \lambda_S$ .

Figure 2 shows an example of a TFSM with timeouts and its  $\mathbb{1}$ -abstraction. In this case the untimed abstraction accepts untimed input words on  $I \cup \{\mathbb{1}\}$  where the delay is implicitly represented by sequences of the special input symbol  $\mathbb{1}$  interleaved with the occurrences of the original input symbols from  $I$ . A sequence of  $n$   $\mathbb{1}$  in a row represents a delay  $t$  included in the right-open interval  $[n, n + 1)$ .

To compare timed words with untimed ones, we need to introduce the following notion of abstraction of a timed word. Given a real number  $t \in \mathbb{R}$ , we denote with  $\lfloor t \rfloor$  the integer part of  $t$ , and with  $\mathbb{1}^{\lfloor t \rfloor}$  a (possibly empty) sequence of  $\lfloor t \rfloor$  delay symbols  $\mathbb{1}$ .

**Definition 8.** Given a finite alphabet  $A$  and a finite timed word  $v = (a_1, t_1) (a_2, t_2) (a_3, t_3) \dots (a_m, t_m)$ , we define its  $\mathbb{1}$ -abstraction as the finite word  $\mathbb{1}(v) = \mathbb{1}^{\lfloor t_1 \rfloor} a_1 \mathbb{1}^{\lfloor t_2 - t_1 \rfloor} \dots \mathbb{1}^{\lfloor t_j - t_{j-1} \rfloor} a_j \mathbb{1}^{\lfloor t_{j+1} - t_j \rfloor} \dots \mathbb{1}^{\lfloor t_m - t_{m-1} \rfloor} a_m$ .

$\mathbb{1}$ -bisimulation connects *timed states*  $(s, x)$  of a timed FSM with states of an untimed FSM.

**Definition 9.** Given a TFSM with timeouts  $T = (S, I, O, \lambda_S, s_0, \Delta_S)$  and an untimed FSM  $U = (R, I \cup \{\mathbb{1}\}, O \cup \{\mathbb{1}\}, \lambda_R, r_0)$ , a  $\mathbb{1}$ -bisimulation is a relation  $\sim \subseteq (S \times \mathbb{R}^+) \times R$  that respects the following conditions for every pair of states  $(s, x) \in S \times \mathbb{R}^+$  and  $r \in R$  such that  $(s, x) \sim r$ :

1. if  $(s, x) \xrightarrow{t} (s', x')$  with  $\lfloor x+t \rfloor = \lfloor x+1 \rfloor$  then there exists  $r' \in R$  such that  $r \xrightarrow{\mathbb{1}, \mathbb{1}} r'$  and  $(s', x') \sim r'$ ;
2. if  $r \xrightarrow{\mathbb{1}, \mathbb{1}} r'$  then for every  $t$  such that  $\lfloor x+t \rfloor = \lfloor x+1 \rfloor$  there exists  $(s', x') \in S \times \mathbb{R}^+$  such that  $(s, x) \xrightarrow{t} (s', x')$  and  $(s', x') \sim r'$ ;
3. if  $(s, x) \xrightarrow{t} (s, x+t) \xrightarrow{i, o} (s', 0)$  with  $\lfloor x \rfloor = \lfloor x+t \rfloor$  then there exists  $r' \in R$  such that  $r \xrightarrow{i, o} r'$  and  $(s', 0) \sim r'$ ;
4. if  $r \xrightarrow{i, o} r'$  then for every  $t \in \mathbb{R}$  such that  $\lfloor x \rfloor = \lfloor x+t \rfloor$  there exists  $(s', 0) \in S \times \mathbb{R}^+$  such that  $(s, x) \xrightarrow{t} (s, x+t) \xrightarrow{i, o} (s', 0)$  and  $(s', 0) \sim r'$ .

$T$  and  $U$  are  $\mathbb{1}$ -bisimilar if there exists a  $\mathbb{1}$ -bisimulation  $\sim \subseteq S \times R$  such that  $(s_0, 0) \sim r_0$ .

To understand the previous conditions, notice that  $\lfloor x+t \rfloor = \lfloor x+1 \rfloor$  implies  $1 \leq t < 2$  and that  $\lfloor x \rfloor = \lfloor x+t \rfloor$  implies  $0 \leq t < 1$ . Therefore condition 1. refers to a timed transition in  $T$  of length  $1 \leq t < 2$ , which corresponds to the existence of a  $\mathbb{1}$ ,  $\mathbb{1}$ -transition in  $U$ . Similarly, condition 3. refers in  $T$  to a timed transition of length  $0 \leq t < 1$  followed by an input-output transition, which corresponds to an input-output transition in  $U$ . Finally, condition 2. refers to a  $\mathbb{1}$ ,  $\mathbb{1}$ -transition in  $U$ , which corresponds to timed transitions in  $T$  with  $1 \leq t < 2$ ; condition 4. refers to an input-output transition in  $U$ , which corresponds in  $U$  to timed transitions with  $0 \leq t < 1$  followed by an input-output transition. The timed transitions for  $t \geq 2$  are handled by induction in the following Lemma 3.

**Lemma 3.** Given a TFSM with timeouts  $T = (S, I, O, \lambda_S, s_0, \Delta_S)$  and an untimed FSM  $U = (R, I \cup \{\mathbb{1}\}, O \cup \{\mathbb{1}\}, \lambda_R, r_0)$ , every  $\mathbb{1}$ -bisimulation relation  $\sim \subseteq (S \times \mathbb{R}^+) \times R$  respects the following properties for every  $(s, 0) \sim r$  and  $t \geq 1$ :

- (i) if  $(s, 0) \xrightarrow{t} (s', x')$  then there exists  $r'$  such that  $(s', x') \sim r'$  and  $r \xrightarrow{\mathbb{1}, \mathbb{1}^{\lfloor t \rfloor}} r'$ ;
- (ii) if  $r \xrightarrow{\mathbb{1}, \mathbb{1}^{\lfloor t \rfloor}} r'$  then there exists  $(s', x') \sim r'$  such that  $(s, 0) \xrightarrow{t} (s', x')$ .

*Proof.* The proof is by induction on  $\lfloor t \rfloor$ . For the basis of the induction, suppose  $\lfloor t \rfloor = 1$  ( $1 \leq t < 2$ ) and let  $(s, 0) \sim r$ . The two properties are a direct consequence of the definition of  $\mathbb{1}$ -bisimulation. By condition 1 of Definition 9, we have that for every  $1 \leq t < 2$ ,  $(s, 0) \xrightarrow{t} (s', x')$  implies that there exists  $r'$  such that  $(s', x') \sim r'$  and  $r \xrightarrow{\mathbb{1}, \mathbb{1}} r'$ . By condition 2 of Definition 9, we have that for every  $1 \leq t < 2$ ,  $r \xrightarrow{\mathbb{1}, \mathbb{1}} r'$  implies that there exists  $(s', x') \sim r'$  such that  $(s, 0) \xrightarrow{t} (s', x')$ .

For the inductive case, suppose that  $\lfloor t \rfloor \geq 2$  and that the Lemma holds for  $\lfloor t-1 \rfloor \geq 1$ . Now, let  $(s, 0) \xrightarrow{t} (s', x')$  and consider the timed state  $(s'', x'')$  such that  $(s, 0) \xrightarrow{t-1} (s'', x'') \xrightarrow{1} (s', x')$ . By inductive hypothesis we have that there exists  $r''$  such that  $(s'', x'') \sim r''$  and  $r \xrightarrow{\mathbb{1}, \mathbb{1}^{\lfloor t-1 \rfloor}} r''$ . By condition 1 of Definition 9, we have that there exists  $r'$  such that  $(s', x') \sim r'$  and  $r'' \xrightarrow{\mathbb{1}, \mathbb{1}} r'$  and thus that  $r \xrightarrow{\mathbb{1}, \mathbb{1}^{\lfloor t \rfloor}} r'$ . To prove property (ii), suppose  $r \xrightarrow{\mathbb{1}, \mathbb{1}^{\lfloor t \rfloor}} r'$  and consider the state  $r''$  such that  $r \xrightarrow{\mathbb{1}, \mathbb{1}^{\lfloor t-1 \rfloor}} r'' \xrightarrow{\mathbb{1}, \mathbb{1}} r'$ . By inductive hypothesis we have that there exists  $(s'', x'') \sim r''$  such that  $(s, 0) \xrightarrow{t-1} (s'', x'')$ . By condition 2 of Definition 9 it is possible to find a state  $(s', x')$  such that  $(s', x') \sim r'$  and  $(s'', x'') \xrightarrow{1} (s', x')$ . This shows that  $(s, 0) \xrightarrow{t} (s', x')$  and concludes the proof.  $\square$

$\mathbb{1}$ -bisimilar machines have the same behavior, as formally proved in the following.

**Lemma 4.** *Given a TFSM with timeouts  $T = (S, I, O, \lambda_S, s_0, \Delta_S)$  and an untimed FSM  $U = (R, I \cup \{\mathbb{1}\}, O \cup \{\mathbb{1}\}, \lambda_R, r_0)$ , if there exists a  $\mathbb{1}$ -bisimulation  $\sim$  such that  $(s_0, 0) \sim r_0$  then for every timed input word  $v = (i_1, t_1) \dots (i_m, t_m)$  we have that  $\mathbb{1}(B_T(v)) = B_U(\mathbb{1}(v))$ .*

*Proof.* We prove the lemma by showing that “for every pair of states  $s \in S$  and  $r \in R$  such that  $(s, 0) \sim r$  and timed word  $v$ ,  $\lambda_S(s, v) = (s', w)$  if and only if  $\lambda_R(r, \mathbb{1}(v)) = (r', \mathbb{1}(w))$  with  $(s', 0) \sim r'$ ”.

We prove the claim by induction on the length  $m$  of the input word. Suppose  $m = 1$ ,  $v = (i_1, t_1)$  and  $w = (o_1, t_1)$ . By the definition of TFSM with timeouts we have that  $\lambda_S(s, (i_1, t_1)) = (s_1, (o_1, t_1))$  if and only if there exists a timed state  $(s', x')$  such that  $(s, 0) \xrightarrow{t_1} (s', x') \xrightarrow{i_1, o_1} (s_1, 0)$ . To prove the direct implication, suppose  $\lambda_S(s, v) = (s', w)$ . We distinguish between two cases depending on the value of  $t_1$ .

- If  $t_1 < 1$ , by condition 3. of the definition of  $\mathbb{1}$ -bisimulation, there exists  $r_1 \in R$  such that  $r \xrightarrow{i_1, o_1} r_1$ . Hence,  $\lambda_R(r, \mathbb{1}(i_1, t_1)) = (r_1, \mathbb{1}(o_1, t_1))$ .
- If  $t_1 \geq 1$ , by Lemma 3 (i), there exists  $r'$  such that  $r \xrightarrow{\mathbb{1}, \mathbb{1}^{\lfloor t_1 \rfloor}} r'$  and  $(s', x') \sim r'$ . By condition 3. of the definition of  $\mathbb{1}$ -bisimulation, we have that it is possible to find a state  $r_1 \in R$  such that  $r' \xrightarrow{i_1, o_1} r_1$ . This implies that under input  $\mathbb{1}^{\lfloor t_1 \rfloor} i_1 = \mathbb{1}(i_1, t_1)$  the FSM  $U$  produces the output word  $\mathbb{1}^{\lfloor t_1 \rfloor} o_1 = \mathbb{1}(o_1, t_1)$ , and thus we can conclude that  $\lambda_R(r, \mathbb{1}(i_1, t_1)) = (r_1, \mathbb{1}(o_1, t_1))$ .

To prove the converse implication, suppose  $\lambda_R(r, \mathbb{1}(i_1, t_1)) = (r_1, o_1) = (r_1, \mathbb{1}(o_1, t_1))$ . We distinguish between two cases depending on the value of  $t_1$ .

- If  $t_1 < 1$ , by condition 4. of the definition of  $\mathbb{1}$ -bisimulation (applied with  $t = 0$ ), there exists  $(s_1, 0) \in S \times \mathbb{R}$  such that  $(s, 0) \xrightarrow{t_1} (s, t_1) \xrightarrow{i_1, o_1} (s_1, 0)$ . Hence,  $\lambda_S(s, (i_1, t_1)) = (s_1, (o_1, t_1))$ .
- If  $t_1 \geq 1$ , then by the assumption  $\lambda_R(r, \mathbb{1}(i_1, t_1)) = (r_1, o_1) = (r_1, \mathbb{1}(o_1, t_1))$  there exists  $r' \in R$  such that  $r \xrightarrow{\mathbb{1}, \mathbb{1}^{\lfloor t_1 \rfloor}} r' \xrightarrow{i_1, o_1} r_1$ . By Lemma 3(ii), there exists  $(s', x') \in S \times \mathbb{R}$  such that  $(s, 0) \xrightarrow{t_1} (s', x')$  and  $(s', x') \sim r'$ . By condition 4. of the definition of  $\mathbb{1}$ -bisimulation (applied with  $t = 0$ ), we have that there exists a timed state  $(s_1, 0)$  such that  $(s', x') \xrightarrow{i_1, o_1} (s_1, 0)$ . This implies that under input  $(i_1, t_1)$  the TFSM  $T$  produces the timed output word  $(o_1, t_1)$ , and thus we can conclude that  $\lambda_S(s, (i_1, t_1)) = (s_1, (o_1, t_1))$ .

To prove the inductive case, suppose  $m > 1$ ,  $v = (i_1, t_1) \dots (i_m, t_m)$  and  $w = (i_1, t_1) \dots (i_m, t_m)$ . Now, let  $v' = (i_1, t_1) \dots (i_{m-1}, t_{m-1})$  and  $w' = (o_1, t_1) \dots (o_{m-1}, t_{m-1})$ . By inductive hypothesis, we have that  $\lambda_S(s, v') = (s_{m-1}, w')$  if and only if  $\lambda_R(r, \mathbb{1}(v')) = (r_{m-1}, \mathbb{1}(w'))$  and that  $\lambda_S(s_{m-1}, (i_m, t_m - t_{m-1})) = (s_m, (o_m, t_m - t_{m-1}))$  if and only if  $\lambda_R(r_{m-1}, \mathbb{1}(i_m, t_m - t_{m-1})) = (r_m, \mathbb{1}(o_m, t_m - t_{m-1}))$  for some  $(s_{m-1}, 0) \sim r_{m-1}$  and  $(s_m, 0) \sim r_m$ . This implies that  $\lambda_S(s, v'(i_m, t_m)) = (s_m, w'(o_m, t_m))$  if and only if  $\lambda_R(r, \mathbb{1}(v)) = \lambda_R(r, \mathbb{1}(v'(i_m, t_m))) = (r_m, \mathbb{1}(w') \mathbb{1}(o_m, t_m - t_{m-1})) = (r_m, \mathbb{1}(w))$ , and thus that the claim holds also for  $m$ .

To conclude the proof of the Lemma it is sufficient to recall that from the definition of behaviour we have that  $B_T(v) = w$  if and only if  $\lambda_S(s_0, v) = (s_m, w)$  for some state  $s_m \in S$ . From  $(s_0, 0) \sim r_0$  we can conclude that  $\lambda_R(r_0, \mathbb{1}(v)) = (r_m, \mathbb{1}(w))$  and thus that  $B_U(\mathbb{1}(v)) = \mathbb{1}(w) = \mathbb{1}(B_T(v))$ .  $\square$

**Lemma 5.** *A TFSM with timeouts  $M$  is  $\mathbb{1}$ -bisimilar to the abstract FSM  $A_M$ .*

*Proof.* The relation  $\sim = \{((s, x), (s, n)) \mid \Delta_S(s)_{\downarrow \mathbb{N}} < \infty \text{ and } \lfloor x \rfloor = n\} \cup \{((s, x), (s, 0)) \mid \Delta_S(s)_{\downarrow \mathbb{N}} = \infty\}$  is a  $\mathbb{1}$ -bisimulation for  $M$  and  $A_M$ .  $\square$

**Theorem 2.** *Let  $M$  and  $M'$  be two TFSM with timeouts. Then  $M$  and  $M'$  are equivalent if and only if the two abstract FSM  $A_M$  and  $A_{M'}$  are equivalent.*



### 2.3 TFMSM with timeouts and timed guards

In this paper we define a new timed FSM model that incorporates both guards and timeouts. Informally, each state of the machine has a timeout (possibly  $\infty$ ) and all outgoing transitions of the state have timed guards with upper bounds less than the state timeout. As in the other models described above, time is set to zero when executing a transition.

**Definition 10** (Timed FSM). *A timed FSM  $\mathcal{S}$  is a finite state machine augmented with timed guards and timeouts. Formally, a timed FSM (TFMSM) is a 6-tuple  $(S, I, O, \lambda_S, s_0, \Delta_S)$  where  $S, I,$  and  $O$  are finite disjoint non-empty sets of states, inputs and outputs, respectively,  $s_0$  is the initial state,  $\lambda_S \subseteq S \times (I \times \Pi) \times O \times S$  is a transition relation where  $\Pi$  is the set of input timed guards, and  $\Delta_S : S \rightarrow S \times (N \cup \{\infty\})$  is a timeout function.*

Similarly to FSMs with timeouts, if no input is applied at a current state  $s$  before the timeout  $\Delta_S(s)_{\downarrow N}$  expires, then the TFMSM will move to another state  $\Delta_S(s)_{\downarrow S}$  as prescribed by the timeout function. If  $\Delta_S(s)_{\downarrow N} = \infty$ , then the TFMSM can stay at state  $s$  infinitely long waiting for an input. Similarly to FSMs with timed guards, an input/output transition can be triggered only if the value of the clock is inside the guard  $\langle t_{min}, t_{max} \rangle$  labeling the transition. Timed transitions are thus defined as for TFMSM with timeouts, while input/output transitions are defined as for TFMSM with timed guards:

- The *timed transition* relation  $\xrightarrow{t}$  is the smallest relation closed under the following properties:
  - for every timed state  $(s, x)$  and delay  $t \geq 0$ , if  $x + t < \Delta_S(s)_{\downarrow N}$ , then  $(s, x) \xrightarrow{t} (s, x + t)$ ;
  - for every timed state  $(s, x)$  and delay  $t \geq 0$ , if  $x + t = \Delta_S(s)_{\downarrow N}$ , then  $(s, x) \xrightarrow{t} (s', 0)$  with  $s' = \Delta_S(s)_{\downarrow S}$ ;
  - if  $(s, x) \xrightarrow{t_1} (s', x')$  and  $(s', x') \xrightarrow{t_2} (s'', x'')$  then  $(s, x) \xrightarrow{t_1+t_2} (s'', x'')$ .
- The *input/output transition* relation  $\xrightarrow{i,o}$  is such that  $(s, x) \xrightarrow{i,o} (s', 0)$  if and only if there exists  $(s, i, \langle t_{min}, t_{max} \rangle, o, s') \in \lambda_S$  such that  $x \in \langle t_{min}, t_{max} \rangle$ .

**Equivalence checking of TFMSM with timeouts and timed guards.** We can solve the equivalence problem for TFMSM with timeouts and timed guards by combining the techniques used for TFMSM with timeouts (Section 2.2) and for TFMSM with timed guards (Section 2.1). To incorporate the effect of guards in the abstract untimed FSM, we have to use a finer granularity for the “time elapsing” transitions, which are now labelled with the special pair  $(\mathfrak{t}, \mathfrak{t})$ , which intuitively represents a time delay  $0 < t^* < 1$  without inputs. In the case of a TFMSM with timeouts and timed guards  $M$ , the constant  $\max(M)$  is defined as the maximum between the greatest timeout value of the function  $\Delta_S$  (different from  $\infty$ ) and the greatest integer constant (different from  $\infty$ ) appearing in the guards of  $\lambda_S$ . States of the abstract FSM will be pairs  $(s, \langle n, n' \rangle)$  where  $s$  is a state of  $M$  and  $\langle n, n' \rangle$  is either a point-interval  $[n, n]$  or an open interval  $(n, n + 1)$  from the set  $\mathbb{I}_N$  defined in 2.1 for the abstraction of TFMSM with timed guards.

**Definition 11.** *Given a TFMSM with timeouts and timed guards  $M = (S, I, O, \lambda_S, s_0, \Delta_S)$ , let  $N = \max(M)$ . We define the  $\mathfrak{t}$ -abstract FSM  $A_M = (S \times \mathbb{I}_N, I \cup \{\mathfrak{t}\}, O \cup \{\mathfrak{t}\}, \lambda_A, (s_0, [0, 0]))$  as the untimed FSM such that:*

- $(s, [n, n]) \xrightarrow{\mathfrak{t}, \mathfrak{t}} (s, (n, n + 1))$  if and only if  $n + 1 \leq \Delta_S(s)_{\downarrow N}$ ;
- $(s, (n, n + 1)) \xrightarrow{\mathfrak{t}, \mathfrak{t}} (s, [n + 1, n + 1])$  if and only if  $n + 1 < \Delta_S(s)_{\downarrow N}$ ;
- $(s, (n, n + 1)) \xrightarrow{\mathfrak{t}, \mathfrak{t}} (s', [0, 0])$  if and only if  $\Delta_S(s) = (s', n + 1)$ ;
- $(s, (N, \infty)) \xrightarrow{\mathfrak{t}, \mathfrak{t}} (s, (N, \infty))$  if and only if  $\Delta_S(s)_{\downarrow N} = \infty$ ;
- $(s, \langle n, n' \rangle) \xrightarrow{i,o} (s', [0, 0])$  if and only if there exists  $(s, i, \langle t, t' \rangle, o, s') \in \lambda_S$  such that  $\langle n, n' \rangle \subseteq \langle t, t' \rangle$ .

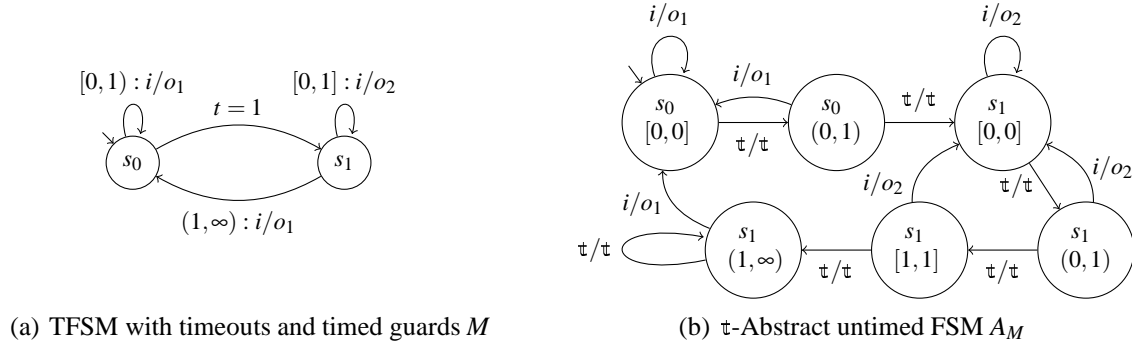
Figure 3:  $\tau$ -abstraction of TFSM with timeout and timed guards.

Figure 3 shows an example of a TFSM with timeouts and its  $\tau$ -abstraction. In this case the untimed abstraction accepts untimed input words on  $I \cup \{\tau\}$ . As for TFSM with timeouts, the delay is implicitly represented by sequences of the special input symbol  $\tau$  interleaving the occurrences of the real input symbols from  $I$ . In this case the representation of delays is more involved:

- an *even number*  $2n$  of  $\tau$  symbols represents a delay of *exactly*  $n$  time units;
- an *odd number*  $2n + 1$  of  $\tau$  symbols represents a delay  $t$  included in the open interval  $(n, n + 1)$ .

The notion of abstraction of a timed word captures the above intuition.

**Definition 12.** Let  $\tau(t)$  be a function mapping a delay  $t \in \mathbb{R}$  to a sequence of  $\tau$  as follows:  $\tau(t) = \tau^{2t}$  if  $\lfloor t \rfloor = t$ ,  $\tau(t) = \tau^{2\lfloor t \rfloor + 1}$  otherwise. Given a finite alphabet  $A$  and a finite timed word  $v = (a_1, t_1)(a_2, t_2)(a_3, t_3) \dots (a_m, t_m)$ , we define its  $\tau$ -abstraction as the finite word  $\tau(v) = \tau(t_1)a_1\tau(t_2 - t_1) \dots \tau(t_j - t_{j-1})a_j\tau(t_{j+1} - t_j) \dots \tau(t_{m-1} - t_m)a_m$ .

The definition of  $\tau$ -bisimulation is similar to the one of  $\mathbb{1}$ -bisimulation. As in Definition 9, conditions 1. and 2. formalize the connection between timed transitions and the special symbol  $\tau$ . The finer granularity of the time delays allows us to simplify conditions 3. and 4.: differently from Definition 9, we do not need to consider timed transitions  $(s, x) \xrightarrow{t} (s, x + t)$  before the actual input/output transition.

**Definition 13.** Given a TFSM with timed guards and timeouts  $T = (S, I, O, \lambda_S, s_0, \Delta_S)$  and an untimed FSM  $U = (R, I \cup \{\tau\}, O \cup \{\tau\}, \lambda_R, r_0)$ , a  $\tau$ -bisimulation is a relation  $\sim \subseteq (S \times \mathbb{R}^+) \times R$  that respects the following conditions for every pair of states  $(s, x) \in S \times \mathbb{R}^+$  and  $r \in R$  such that  $(s, x) \sim r$ :

1. if  $(s, x) \xrightarrow{t} (s', x')$  with  $0 < t < 1$  and either  $x \in \mathbb{N}$  or  $x + t \in \mathbb{N}$  then there exists  $r' \in R$  such that  $r \xrightarrow{\tau, \tau} r'$  and  $(s', x') \sim r'$ ;
2. if  $r \xrightarrow{\tau, \tau} r'$  then for every  $0 < t < 1$  such that either  $x \in \mathbb{N}$  or  $x + t \in \mathbb{N}$  there exists  $(s', x') \in S \times \mathbb{R}^+$  such that  $(s, x) \xrightarrow{t} (s', x')$  and  $(s', x') \sim r'$ ;
3. if  $(s, x) \xrightarrow{i, o} (s', 0)$  then there exists  $r' \in R$  such that  $r \xrightarrow{i, o} r'$  and  $(s', 0) \sim r'$ ;
4. if  $r \xrightarrow{i, o} r'$  then there exists  $(s', 0) \in S \times \mathbb{R}^+$  such that  $(s, x) \xrightarrow{i, o} (s', 0)$  and  $(s', 0) \sim r'$ .

$T$  and  $U$  are  $\tau$ -bisimilar if there exists a  $\tau$ -bisimulation  $\sim \subseteq S \times R$  such that  $(s_0, 0) \sim r_0$ .

The following lemma proves that  $\tau$ -bisimilar machines have the same behavior.

**Lemma 6.** Given a TFSM with timeouts  $T = (S, I, O, \lambda_S, s_0, \Delta_S)$  and an untimed FSM  $U = (R, I \cup \{\tau\}, O \cup \{\tau\}, \lambda_R, r_0)$ , if there exists a  $\tau$ -bisimulation  $\sim$  such that  $(s_0, 0) \sim r_0$  then for every timed input word  $v = (i_1, t_1) \dots (i_m, t_m)$  we have that  $\tau(B_T(v)) = B_U(\tau(v))$ .

*Proof.* The claim can be proved using the same argument of Lemma 4. See [2] for details.  $\square$

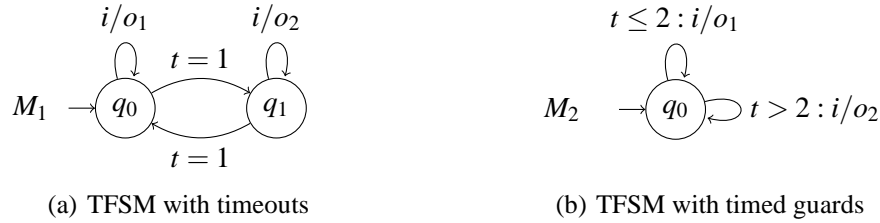


Figure 4: Expressivity counterexamples.

**Lemma 7.** *A TFSM with timeouts and timed guards  $M$  is  $\tau$ -bisimilar to the abstract FSM  $A_M$ .*

*Proof.* The relation  $\sim = \{((s, x), (s, \langle n, n' \rangle)) \mid x \in \langle n, n' \rangle\}$  is a  $\tau$ -bisimulation for  $M$  and  $A_M$ . □

**Theorem 3.** *Let  $M$  and  $M'$  be two TFSM with timeouts and timed guards. Then  $M$  and  $M'$  are equivalent if and only if the two abstract FSM  $A_M$  and  $A_{M'}$  are equivalent.*

### 3 Comparison of TFSM models

In this section we compare the TFSM models considered in this paper with respect to their expressivity. It is easy to see that the class of TFSM with timeouts and timed guards includes both TFSM with timeouts and TFSM with timed guards. We start our comparison by showing that TFSM with timed guards and TFSM with timeouts are incomparable. To this end, consider the two TFSM of Figure 4: the following lemmas prove that there is no TFSM with timed guards equivalent to the TFSM with timeouts  $M_1$ , and no TFSM with timeouts equivalent to the TFSM with timed guards  $M_2$ .

**Proposition 1.** *Let  $I = \{i\}$ ,  $O = \{o_1, o_2\}$  and consider the complete TFSM with timeouts  $M_1$  depicted in Figure 4(a). The behavior of  $M_1$  cannot be described by any TFSM with timed guards.*

*Proof.* It is easy to see that  $M_1$ , under input  $i$ , produces the output  $o_1$  only at those time instants  $t$  such that  $2n \leq t < 2n + 1$  for some natural  $n$ , while for time instants  $t'$  such that  $2n + 1 \leq t' < 2n + 2$  the machine produces the output  $o_2$ . Suppose that there exists a complete TFSM with timed guards  $M'_1 = (S, I, O, \lambda_S, s_0, \Delta_S)$  that is equivalent to  $M_1$ . Let  $t_{max}$  be the maximum value that appears on the guards of the transitions exiting the initial state  $s_0$  of  $M'_1$  that are labelled with  $i/o_1$ . Two cases may arise.

- $t_{max} < +\infty$ . In this case, let  $n$  be such that  $t_{max} < 2n + \frac{1}{2}$ , and let  $w$  be the single-letter timed input word  $(i, 2n + \frac{1}{2})$ .  $M_1$  accepts  $w$  and produces the corresponding timed output word  $(o_1, 2n + \frac{1}{2})$ . Consider now the TFSM with timed guards  $M'_1$ . At time  $2n + \frac{1}{2}$  the TFSM  $M'_1$  is still in the initial state  $s_0$ . However, no active transition exiting  $s_0$  at time  $2n + \frac{1}{2}$  is labelled with  $i/o_1$ , since  $t_{max} < 2n + \frac{1}{2}$ . Since  $M'_1$  is complete, there must exist an active transition from  $s_0$  at time  $2n + \frac{1}{2}$ , but this transition must be labelled with  $i/o_2$ . Hence, under input  $w$ ,  $M'_1$  must produce the timed word  $(o_2, 2n + \frac{1}{2})$ , in contradiction with the hypothesis that  $M_1$  and  $M'_1$  are equivalent.
- $t_{max} = +\infty$ . In this case, there is a transition exiting  $s_0$  labelled with  $i/o_1$  and with an interval  $\langle t_f, +\infty \rangle$ , for some  $t_f < +\infty$ . Let  $n$  be such that  $t_f < 2n + 1 + \frac{1}{2}$ , and let  $w$  be the single-letter timed input word  $(i, 2n + 1 + \frac{1}{2})$ .  $M_1$  accepts  $w$  and produces the output word  $(o_2, 2n + 1 + \frac{1}{2})$ . Consider now the TFSM with timed guards  $M'_1$ . Since no input is received before, at time  $2n + 1 + \frac{1}{2}$  the TFSM  $M'_1$  is still in the initial state  $s_0$ . However, since  $t_f < 2n + 1 + \frac{1}{2} < +\infty$ , a transition labelled

---

**Algorithm 1:** Transform a loop-free TFSM with timeouts into a TFSM with timed guards

---

**Input:** A loop-free TFSM with timeouts  $M = (S, I, O, \lambda_S, s_0, \Delta_S)$

```

1 while there exists  $s_j \in S$  such that  $\Delta_S(s_j)_{\downarrow \mathbb{N}} < \infty$  do
2   let  $s_j \in S$  such that  $\Delta_S(s_j)_{\downarrow \mathbb{N}} < \infty$ ;
3   let  $(s_k, n) = \Delta_S(s_j)$ ;
4   foreach  $(s_k, i, [t_1, t_2], o, s_h) \in \lambda_S$  do  $\lambda_S = \lambda_S \cup \{(s_j, i, [t_1 + n, t_2 + n], o, s_h)\}$ ;
5   if  $\Delta_S(s_k)_{\downarrow \mathbb{N}} < \infty$  then  $\Delta_S(s_j) = (\Delta_S(s_k)_{\downarrow S}, \Delta_S(s_k)_{\downarrow \mathbb{N}} + n)$  else  $\Delta_S(s_j) = (s_j, \infty)$ ;
6 end
7 return  $(S, I, O, \lambda_S, s_0)$ ;

```

---

with  $i/o_1$  is active at time  $2n + 1 + \frac{1}{2}$ . Hence,  $M'_1$  accepts  $w$  and produces the timed output word  $(o_1, 2n + 1 + \frac{1}{2})$ . This is a contradiction with the hypothesis that  $M_1$  and  $M'_1$  are equivalent.  $\square$

**Proposition 2.** Let  $I = \{i\}$ ,  $O = \{o_1, o_2\}$  and consider the TFSM with timed guards  $M_2$  depicted in Figure 4(b). The behavior of  $M_2$  cannot be described by any TFSM with timeouts.

*Proof.* It is easy to see that  $M_2$  (under input  $i$ ) produces the output  $o_1$  only at those time instants  $t$  such that  $t \leq 2$ , while for time instants strictly greater than 2 the machine produces the output  $o_2$ . Suppose by contradiction that there exists a TFSM with timeouts  $M'_2 = (S, I, O, \lambda_S, s_0, \Delta_S)$  that is equivalent to  $M_2$ , and consider the single-letter timed input word  $w = (i, 2)$ . We have that  $M_2$  accepts  $w$  and produces the output  $o_1$  at time 2. Since  $M'_2$  is equivalent to  $M_2$ , we have that  $w$  must be accepted by  $M'_2$  as well. Let  $(s, x)$  be a timed state of  $M'_2$  such that  $(s_0, 0) \xrightarrow{2} (s, x)$ . Since  $w$  is accepted, we have that there must exist a transition  $(s, i, o_1, s') \in \lambda_S$  for some  $s' \in S$ . Now, let  $w'$  be the timed input word  $(i, 2 + \varepsilon)$ , for some  $\varepsilon < 1 \leq \Delta_S(s)_{\downarrow \mathbb{N}}$ . By the definition of TFSM with timeouts, we have that  $(s_0, 0) \xrightarrow{2+\varepsilon} (s, x + \varepsilon)$ ; thus  $w'$  is accepted by  $M'_2$ , which produces the output  $o_1$  at time  $2 + \varepsilon$ , since the input/output transition  $(s, i, o_1, s') \in \lambda_S$  can be triggered from the timed state  $(s, x + \varepsilon)$ . However,  $M_2$  produces the output  $o_2$  with input  $w'$ , in contradiction with the hypothesis that  $M_2$  and  $M'_2$  are equivalent.  $\square$

The above examples show that, in general, TFSM with timeouts cannot be transformed into TFSM with timed guards, and that TFSM with timed guards cannot be transformed into TFSM with timeouts. We will now study the restrictions under which this transformation is possible,

Consider the TFSM with timeouts  $M_1$  of Figure 4(a): it shows that that cycles of timeout transitions cannot be captured by a TFSM with timed guards. However, when there are no such cycles, we can use Algorithm 1 to transform a TFSM with timeouts into a TFSM with timed guards.

**Proposition 3.** Given a TFSM with timeouts  $M$  without loops of timeout transitions, Algorithm 1 terminates and builds a TFSM with timed guards that is equivalent to  $M$ .

*Proof.* See [2].  $\square$

Consider now the TFSM with timed guards  $M_2$  of Figure 4(b): it suggests that transitions with left-open guards  $(t_1, t_2)$  or with right-closed guards  $\langle t_1, t_2 \rangle$  cannot be captured by TFSM with timeouts. The following proposition shows that if we force all guards to be left-closed and right-open intervals  $[t_1, t_2)$ , then the translation into a TFSM with timeouts is possible.

**Proposition 4.** The behavior of a TFSM with timed guards can be described by a TFSM with timeouts if all guards are left-closed and right-open.

*Proof.* Let  $M = (S, I, O, \lambda_S, s_0)$  be a TFSM with timed guards where every transition  $(s, i, g, o, s') \in \lambda_S$  is such that  $g = [t_{min}, t_{max})$  is a left-closed and right-open interval. We build a TFSM with timeouts  $M' = (S', I, O, \lambda'_S, s'_0, \Delta'_S)$  equivalent to  $M$  as follows.

1. Given an enumeration of the states  $S = \{s_0, s_1, \dots, s_n\}$  of  $M$ , for each state  $s_j \in S$  we define the corresponding set  $T_j = \{0 = t_0^j < t_1^j < \dots < t_m^j = \infty\}$  as the set of all values that appear in the guards of transitions departing from  $s_j$ . The states of  $M'$  are pairs  $(s_j, [t_k^j, t_{k+1}^j))$  where  $s_j \in S$  is a state of  $M$  and  $t_k^j, t_{k+1}^j \in T_j$ . Intuitively, a state  $(s_j, [t_k^j, t_{k+1}^j))$  of  $M'$  corresponds to the situation when  $M$  is in state  $s_j$  with clock value in the interval  $[t_k^j, t_{k+1}^j)$ .
2. The timeout transition is defined as  $\Delta'_S((s_j, [t_k^j, t_{k+1}^j))) = ((s_j, [t_{k+1}^j, t_{k+2}^j)), t_{k+1}^j - t_k^j)$  when  $t_{k+1}^j < \infty$ , and as  $\Delta'_S((s_j, [t_k^j, t_{k+1}^j))) = ((s_j, [t_k^j, t_{k+1}^j)), \infty)$  when  $t_{k+1}^j = \infty$ .
3. For every  $(s_j, i, [t_{min}, t_{max}), o, s_k) \in \lambda_S$  we put a transition  $((s_j, [t_h^j, t_{h+1}^j)), i, o, (s_k, [0, t_1^k))) \in \lambda'_S$  for every  $h$  such that  $t_{min} \leq t_h^j$  and  $t_{h+1}^j \leq t_{max}$ .
4. The initial state of  $M'$  is  $s'_0 = (s_0, [0, t_1^0))$ .

To prove that  $M$  and  $M'$  have the same behavior, we have to show that for every sequence of transitions  $(s_j, 0) \xrightarrow{t} (s_j, t) \xrightarrow{i, o} (s_k, 0)$  of  $M$  there is a corresponding sequence in  $M'$ , and vice-versa. Now, since  $(s_j, t) \xrightarrow{i, o} (s_k, 0)$  is an input/output transition of  $M$ , we have that there exists a transition  $(s_j, i, [t_{min}, t_{max}), o, s_k) \in \lambda_S$  such that  $t \in [t_{min}, t_{max})$ . By 2., we have that  $((s_j, [0, t_1^j)), 0) \xrightarrow{t} ((s_j, [t_h^j, t_{h+1}^j)), t')$  for some  $h$  such that  $t_h^j \leq t < t_{h+1}^j$ , and with  $t' = t - t_h^j$ . By 3., we have that  $((s_j, [t_h^j, t_{h+1}^j)), i, o, (s_k, [0, t_1^k))) \in \lambda'_S$ . Hence,  $((s_j, [0, t_1^j)), 0) \xrightarrow{t} ((s_j, [t_h^j, t_{h+1}^j)), t') \xrightarrow{i, o} ((s_k, [0, t_1^k)), 0)$  is a sequence of transitions of  $M'$ .

To conclude the proof, suppose that  $((s_j, [0, t_1^j)), 0) \xrightarrow{t} ((s_j, [t_h^j, t_{h+1}^j)), t') \xrightarrow{i, o} ((s_k, [0, t_1^k)), 0)$  is a sequence of transitions of  $M'$ . By 2., we have that  $h$  is such that  $t_h^j \leq t < t_{h+1}^j$ . Since  $((s_j, [t_h^j, t_{h+1}^j)), t') \xrightarrow{i, o} ((s_k, [0, t_1^k)), 0)$ , we have that there exists a transition  $((s_j, [t_h^j, t_{h+1}^j)), i, o, (s_k, [0, t_1^k))) \in \lambda'_S$ . By 3., there exists a transition  $(s_j, i, [t_{min}, t_{max}), o, s_k) \in \lambda_S$  such that  $t_{min} \leq t_h^j$  and  $t_{h+1}^j \leq t_{max}$ . Hence, we have proved that  $(s_j, 0) \xrightarrow{t} (s_j, t) \xrightarrow{i, o} (s_k, 0)$  is a valid sequence of transitions in  $M$ .  $\square$

The results obtained in this section are summarized in Figure 5: neither TFSM with timed guards nor TFSM with timeouts are sufficient to describe the behavior of each other, nor of TFSM with both timed guards and timeouts. Moreover, we would like to point out that Algorithm 1 builds a TFSM with LCRO guards from a loop-free TFSM with timeouts, and that the construction in the proof of Proposition 4 builds a loop-free TFSM with timeouts from a TFSM with LCRO guards. This allows us to conclude that the two classes are equivalent.

## 4 Conclusions

In this paper, we investigated some models of deterministic TFSMs with a single clock: TFSMs with only timed guards, TFSMs with only timeouts, and TFSMs with both timed guards and timeouts. We showed that the behaviours of the timed FSMs are equivalent if and only if the behaviours of the companion untimed FSMs obtained by time-abstracting bisimulations are equivalent. Moreover, we compared their expressive power, characterizing subclasses of TFSMs with timed guards and of TFSMs with timeouts that are equivalent to each other. These timed FSM models exhibit a good trade-off between expressive

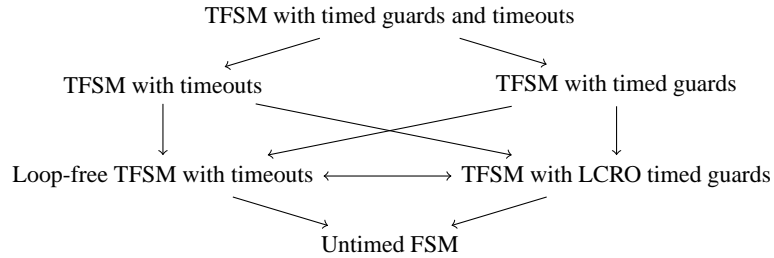


Figure 5: Comparison of TFSM models.

power and ease of analysis. We are currently generalizing these results to non-deterministic timed FSMs, and comparing our models with classical FSMs and special classes of timed automata (e.g., with a single clock). Future work includes deriving tests for a timed FSM with timed guards and timeouts, extending the derivation for a timed FSM with timed guards and for a timed FSM with timeouts, respectively, in [4] and in [9]. Finally, we will define the composition of timed FSMs and investigate the solution of equations over timed FSMs to synthesize unknown timed FSMs [8].

## References

- [1] Rajeev Alur & David L. Dill (1994): *A theory of timed automata*. *Theoretical Computer Science* 126(2), pp. 183 – 235, doi:10.1016/0304-3975(94)90010-8.
- [2] Davide Bresolin, Khaled El-Fakih, Tiziano Villa & Nina Yevtushenko (2014): *Deterministic timed finite state machines: equivalence checking and expressive power*. Research report RR 94/2014, Department of Computer Science, University of Verona, Verona, Italy.
- [3] K. El-Fakih, M. Gromov, N. Shabalina & N. Yevtushenko (2013): *Distinguishing Experiments for Timed Nondeterministic Finite State Machines*. *Acta Cybernetica* 212(2), pp. 205–222.
- [4] Khaled El-Fakih, Nina Yevtushenko & Adenilso Simao (2014): *A practical approach for testing timed deterministic finite state machines with single clock*. *Science of Computer Programming* 80, Part B(0), pp. 343 – 355, doi:10.1016/j.scico.2013.09.008.
- [5] Maxim Gromov, Khaled El-Fakih, Natalia Shabalina & Nina Yevtushenko (2009): *Distinguishing Non-deterministic Timed Finite State Machines*. In: *Formal Techniques for Distributed Systems, Lecture Notes in Computer Science* 5522, Springer Berlin Heidelberg, pp. 137–151, doi:10.1007/978-3-642-02138-1\_9.
- [6] Robert M. Hierons, Mercedes G. Merayo & Manuel Núñez (2009): *Testing from a stochastic timed system with a fault model*. *The Journal of Logic and Algebraic Programming* 78(2), pp. 98 – 115, doi:10.1016/j.jlap.2008.06.001.
- [7] Mercedes G. Merayo, Manuel Núñez & Ismael Rodríguez (2008): *Formal testing from timed finite state machines*. *Computer Networks* 52(2), pp. 432–460, doi:10.1016/j.comnet.2007.10.002.
- [8] T. Villa, N. Yevtushenko, R. Brayton, A. Mishchenko, A. Petrenko & A. Sangiovanni-Vincentelli (2012): *The unknown component problem: theory and applications*. Springer Verlag, doi:10.1007/978-0-387-68759-9.
- [9] M. Zhigulin, N. Yevtushenko, S. Maag & A. Cavalli (2011): *FSM-Based Test Derivation Strategies for Systems with Time-Outs*. In: *Proc. of the 11th Int. Conf. on Quality Software (QSIC 2011)*, pp. 141–149, doi:10.1109/QSIC.2011.30.