DISTINGUING NON-DETERMINISTIC TIMED FINITE STATE MACHINES

Maxim Gromov[1], Khaled El-Fakih[2], Natalia Shabaldina[1], Nina Yevtushenko[1]

[1]Tomsk State University, 36 Lenin Str.. Tomsk, 634050, Russia
gromov@sibmail.com, snv@kitidis.tsu.ru, ninayevtushenko@yahoo.com
[2]American University of Sharjah, PO Box 26666, UAE
kelfakih@aus.edu

Abstract: Conformance testing with the guaranteed fault coverage is based on distinguishing faulty system implementations from the corresponding system specification. We consider timed systems modeled by timed possibly non-deterministic finite state machines (TFSMs) and propose algorithms for distinguishing two TFSMs. In particular, we present a preset algorithm for separating two separable TFSMs and an adaptive algorithm for *r*-distinguishing two possibly non-separable TFSMs. The proposed techniques extend existing methods for untimed non-deterministic FSMs by dealing with the fact that unlike untimed FSMs in general, a TFSM has an infinite number of timed inputs. Correspondingly we state that the upper bounds on the length of distinguishing sequences are the same as for untimed FSMs.

1. Introduction

Timed systems are used in various application areas such as telecommunication systems, plant and traffic controllers and others. A number of formal models have been proposed for testing and verification of timed systems (see, for example, [1], [5], [22]) including systems modeled as timed Finite State Machines (FSMs) [9], [15], [16]. FSMs are widely used in many application areas; in particular, they are used as the underlying models for formal description techniques, such as SDL and UML State Diagrams, and many conformance test derivation methods are based on a specification given in the form of a finite state machine. For surveys see [3], [11] and for some related experiments see [4]. Most of the past work on FSM-based conformance testing has been done for deriving tests for deterministic FSMs w.r.t. the equivalence relation. In addition, there also exist methods for deriving tests for non-deterministic FSMs w.r.t. a number of conformance relations, such as the equivalence, reduction, and the non-separability relations [6], [7], [8], [12], [17], [18], [21]. Two FSMs are equivalent if they have the same input/output behavior and an FSM *P* is a reduction of FSM *S* if the behavior of *P* is contained in the behavior of *S*. Moreover, two FSMs are non-separable [23] if the sets of output responses of these machines to each input sequence intersect. If there exists an input sequence, called a *separating sequence*, such that the output responses of the two FSMs to the sequence are disjoint then the machines are separable. Two complete FSMs are *r*-

distinguishable if they have no common complete reduction. This fact can be checked by a finite set of sequences which is called an *r*-distinguishing set of the two FSMs. In this paper, we say that two FSMs are distinguishable if they are separable or *r*-distinguishable. Experiments that distinguish two FSMs can be classified as adaptive and preset [10]. In an adaptive experiment the next input of an experiment depends on the outputs to previous input sequences and in a preset experiment the whole input sequence is predetermined independently of the intermediate outcome of an experiment. Separating two FSMs can be done in a preset experiment; however, two non-separable FSMs can be still distinguished by an adaptive experiment using the *r*-distinguishability relation.

Testing based on timed FSM models is a difficult task since it requires checking the time constraints of the system in addition to input and output behavior. In the past few years some work has been carried out on deriving test suites based on timed automata. For example, Springintveld et al. [22] proposed a rigorous method that derives test suites with the guaranteed fault coverage w.r.t. the equivalence relation when the system specification and an Implementation Under Test (IUT) are deterministic. The results were extended in [5] to non-deterministic timed automata w.r.t. the equivalence relation under the assumption of "*all weather conditions*" [13], [14], also called *complete testing* assumption in [12]. According to this assumption, if an input sequence (a test case) is applied a number of times to a non-deterministic IUT, then all possible output sequences of the IUT to this test case are observed while testing. Similar to FSM-based methods, the methods in [5], [22] use so-called distinguishing sequences in test derivation; however, these sequences are derived for the equivalence relation. Recently, Merayo et al. [15], [16] considered a timed possibly non-deterministic FSM model. Time constrains limit a time elapsed when an output has to be produced after an input has been applied to the FSM. When an output is produced the clock variable is reset to zero. The model also takes into account time-outs; if no input is applied at a current state for some time-out period, the (timed) FSM moves from current state to another state using a time-out function. Various conformance relations are introduced for such a timed FSM model; however, the problem of deriving distinguishing sequences w.r.t. the proposed relations is not tackled in the papers. A timed model of a stochastic FSM is considered in [9] where the authors propose a method for deriving a complete test suite for the considered model w.r.t. the reduction relation. Distinguishing sets used for deriving a complete test suite extend corresponding sets for untimed FSMs based on related random variables.

When an IUT has a limited controllability, as happens, for instance, in remote testing, the complete testing assumption cannot be satisfied. In this case, the only relation that can be used for the preset testing with the guaranteed fault coverage is the separability relation [19], defined by Starke in [23], and the only relation that can be used for the adaptive testing with the guaranteed fault coverage is the *r*-distinguihability relation. Derivation methods and upper bounds on length of distinguishing sequences for untimed non-deterministic FSMs based on the separability relation can be found in [2], [20] and derivation methods based on the *r*-distinguishability relation can be found in [8], [17], [18]. However, methods given for the derivation of distinguishing sequence for untimed FSMs cannot be directly applied to timed FSMs, since in timed FSMs, in general, the number of timed inputs

is infinite; thus, the extension of these methods is not a trivial problem. Accordingly, in this paper, we propose algorithms for distinguishing timed non-deterministic FSMs (TFSMs) w.r.t. the separability and *r*-distinguishability relations. In particular, given two TFSMs, we present a preset algorithm for deriving a shortest (timed) sequence that separates the two machines, when such a sequence exists. For two non-separable but *r*-distinguishable TFSMs, we present an adaptive algorithm for deriving sequences that *r*-distinguish these machines. We also state that upper bounds on the length of such distinguishing sequences coincide with those of untimed FSMs and similar to untimed FSMs those bounds are reachable. As usual, the algorithms presented in this paper can be used as well for fault diagnosis of timed FSMs.

We note that the TFSM model considered in this paper is somehow similar to that given in [15], [16]. In particular, as in [15], [16], we consider non-deterministic timed FSMs where time constraints are used to limit time elapsed at states and we also use one clock variable that is reset at every transition; however, unlike [15], [16], we do not consider time-outs at states. According to this fact, more complex time constraints can be described by the model in [15], [16]. Another timed model that is used as basis for test derivation is given in [5], [22]. This model is very close to the popular automaton based model presented by Alur and Dill [1]. However, we recall that the work in [22] considers only deterministic input/output behaviors of a timed I/O automaton while the authors in [5] consider non-deterministic behaviors only w.r.t. the equivalence relation under "all weather conditions" assumption. In comparison to the models used for test derivation in [5], [22], the models presented in this paper and in [15], [16] have less modeling capability since one clock is used and the clock is reset at every transition. However, unlike the timed model used in [5], [22], the timed models of this paper and in [15], [16] consider non-determinism and have an FSM as the underlying model. Correspondingly, for such TFSMs, FSM-based methods can be adapted for deriving distinguishing sequences as well as for deriving test suites with the guaranteed fault coverage.

This paper is organized as follows. Section 2 includes preliminaries. Sections 3 and 4 include algorithms, propositions and examples related to the derivation of separating and *r*-distinguishing sequences for timed non-deterministic FSMs. Section 5 concludes the paper.

2. Preliminaries

In this section, we introduce a timed non-deterministic Finite State Machine (TFSM) with some related notions and definitions.

Definition 1. An FSM S is a 5-tuple (S, I, O, λ_S, s_0), where S, I, and O are finite sets of states, inputs and outputs, respectively, s_0 is the initial state and $\lambda_S \subseteq S \times I \times O \times S$ is a behavior relation. □

A timed possibly non-deterministic and partial FSM (TFSM) is an FSM annotated with a *clock*, a time reset operation and time guards associated with transitions. The clock t is a real number that measures the time delay at a state and the time reset operation resets the value of the clock t to zero at the execution of a transition. A time guard g_i describes the time domain when a transition can be executed and is given in

the form $\lceil min, max \rceil$, where $\lceil \in \{(, [\}, \rceil \in \{),]\}$ and *min* and *max* are non-negative rationales such that $min \leq max$. When $min = max$ we consider the only interval $[min, min] = \{min\}$. An output delay describes the time domain when an output has to be produced after an input is applied and is also given in the form $\lceil min, max \rceil$ over rational bounds *min* and *max* where $min \leq max$. Here we assume that the time reset operation is specified at every transition of a given TFSM.

Definition 2. A timed FSM (TFSM) $\mathcal{S}$ often called simply *a machine* throughout the paper, is a 5-tuple $(S, I, O, \lambda_S, s_0)$; the transition relation $\lambda_S \subseteq S \times I \times O \times S \times \Pi \times \aleph$ where Π is the set of time guards over $[0, \infty)$ and $\aleph$ is the set of output delay intervals over $[0, \infty)$.□

The behavior of a TFSM $\mathcal{S}$ can be described as follows. If $(s, i, o, s', g_i = \lceil min, max \rceil, g_o = \lceil min', max' \rceil) \in S \times I \times O \times S \times \Pi \times \aleph$, we say that TFSM $\mathcal{S}$ when being at state *s* and accepting input *i* at time *t* satisfying the time guard $t \in \lceil min, max \rceil$, responds (after the input *i* has been applied) with output *o* within the time delay specified in g_o and moves to the state s'. The clock is reset to zero and starts advancing at s'.

A zero output delay, i.e. $g_o = [0, 0]$, indicates that the output is produced instantly at the time when the input is applied. For simplicity, for a transition with $g_o = [0, 0]$ and input guard g_i over $[0, \infty)$, we omit g_o and g_i from the description of the transition. Thus, a transition (s, i, o, s') indicates that being at state *s* and accepting input *i* at any time, $\mathcal{S}$ responds with output *o* instantly when *i* is applied. In this paper, we consider only functional distinguishability [15], [16] between TFSMs and thus, we do not consider output delays. In other words, the transition relation is a 5-tuple, $\lambda_S \subseteq S \times I \times O \times S \times \Pi$.

TFSM $\mathcal{S}$ is *well-defined* if for each two transitions $(s, i, o, s', \lceil min_1, max_1 \rceil), (s, i, o', s'', \lceil min_2, max_2 \rceil) \in \lambda_S$ s.t. $min_2 \in \lceil min_1, max_1 \rceil$ or $min_1 \in \lceil min_2, max_2 \rceil$ it holds that $o \neq o'$ or $s' \neq s''$. In this paper, we consider only well-defined TFSMs. In this case, we cannot merge two guards, out of the same state and under the same input, without changing the behavior of the TFSM.

A TFSM $\mathcal{S}$ is *observable* if for each two transitions $(s, i, o, s', \lceil min_1, max_1 \rceil), (s, i, o', s'', \lceil min_2, max_2 \rceil) \in \lambda_S$ it holds that if $\lceil min_1, max_1 \rceil \cap \lceil min_1, max_1 \rceil \neq \emptyset$ then $o' = o$ implies $s' = s''$.

The machine $\mathcal{S}$ is (time) *deterministic* if for each two transitions $(s, i, o, s, \lceil min_1, max_1 \rceil), (s, i, o', s', \lceil min_2, max_2 \rceil) \in \lambda_S$, it holds that $\lceil min_1, max_1 \rceil \cap \lceil min_2, max_2 \rceil = \emptyset$; otherwise, the machine $\mathcal{S}$ is (time) *non-deterministic*. Each deterministic TFSM is observable.

The TFSM $\mathcal{S}$ is *input enabled* if the underlying FSM is complete, i.e., if for each pair $(s, i) \in S \times I$, λ_S has a transition $(s, i, o, s', \lceil min, max \rceil)$.

The TFSM $\mathcal{S}$ is *complete* if the underlying FSM is complete and for each pair $(s, i) \in S \times I$ of TFSM $\mathcal{S}$, the union of time guards over all transitions $(s, i, o, s', \lceil min, max \rceil) \in \lambda_S$ equals to $[0, \infty)$; otherwise, the machine is called *partial*. Given a complete TFSM, the behavior of the TFSM is defined at each state for each input that can be applied at any time instance in $[0, \infty)$.

Definition 3. Given a TFSM $\mathcal{S} = (S, I, O, \lambda_S, s_0)$, a pair (i, t), $i \in I$, t is a nonnegative rational, is a *timed input* that states that an input *i* is applied at time *t*.

Given a state s, there is a *clocked transition* $(s, (i, t), o, s')$ in $\mathcal{S}$ if there exists a transition $(s, i, o, s', \lceil min, max \rceil) \in \lambda_S$ with $t \in \lceil min, max \rceil$. □

A TFSM $\mathcal{S} = (S, I, O, \lambda_S, s_0)$ is a *submachine* of TFSM $\mathcal{P} = (P, I, O, \lambda_P, p_0)$ if $S \subseteq P$, $s_0 = p_0$ and each clocked transition $(s, (i, t), o, s')$ of $\mathcal{S}$ is a clocked transition of $\mathcal{P}$.

Definition 4. Given TFSM $\mathcal{S} = (S, I, O, \lambda_S, s_0)$, state s and a (time) guard $g = \lceil min, max \rceil$, state s' is an (i, g)-*successor* of state s if there exists $t \in g$ s.t. $(s, (i, t), o, s')$ is a clocked transition of $\mathcal{S}$. Generally, the set of (i, g)-successors of state s can be empty as well as can have several states. Given a set of states $M \subseteq S$ and a timed guard $g = \lceil min, max \rceil$, the set M' of states is an (i, g)-*successor* of the set M if M' is the union of the sets of (i, g)-successors over all states of the set M. □

Given a TFSM $\mathcal{S} = (S, I, O, \lambda_S, s_0)$ and a pair $(s, i) \in S \times I$, let $G = \{j_1 = 0, j_2, \ldots, j_m\}$, $j_a < j_{a+1}$, $a = 1, \ldots, m - 1$, be the finite ordered set of boundaries of guards over all transitions $(s, i, o, s', g_j) \in \lambda_S$. We denote $\Pi_{(s, i)}$ the (finite) set $\{(j_1, j_2), \ldots, (j_{m-1}, j_m), (j_m, \infty), \{j_1\}, \{j_2\}, \{j_3\}, ..\{j_m\}\}$, i.e., the set $\Pi_{(s, i)}$ has singletons for all boundaries and all (infinite) domains with consecutive boundaries of the set G. By definition, the set $\Pi_{(s, i)}$ is finite and items of the set are very close to regions of the region graph in [1]. Each item of the set $\Pi_{(s, i)}$ describes a time domain (or region) where the TFSM has the same behavior for the pair (s, i). If there is no transition $(s, i, o, s', \lceil min, max \rceil) \in \lambda_S$ then, by definition, $\Pi_{(s, i)}$ is the empty set. By definition of the set $\Pi_{(s, i)}$, the following statement holds.

Proposition 1. Given a TFSM $\mathcal{S} = (S, I, O, \lambda_S, s_0)$, a pair $(s, i) \in S \times I$ s.t. the set $\Pi_{(s, i)}$ is not empty, $g \in \Pi_{(s, i)}$ and $t_1, t_2 \in g$, the sets of (i, t_1)- and (i, t_2)-successors of state s coincide. □

We note that a TFSM can have the same behavior for the pair (s, i) in different domains of the set $\Pi_{(s,i)}$. For example, suppose that λ_S has transitions $(s, i, o_1, s_1, [0, 2))$, $(s, i, o_2, s_2, [2, \infty))$, $(s, i, o_3, s_1, [0, 3))$, $(s, i, o_2, s_1, [3, \infty))$ for (s, i). The set $\Pi_{(s, i)} = \{(0, 2), (2, 3), (3, \infty), \{0\}, \{2\}, \{3\}\}$. The set of $(i, 1)$-successors of state s coincides with the set of $(i, 0.5)$-successors. Moreover, the TFSM at state s has the same behavior for timed inputs $(i, 0)$ and $(i, 1)$ despite of the fact that time instances 0 and 1 belong to different domains of the set $\Pi_{(s, i)}$.

Definition 5. Given a TFSM $\mathcal{S}$, a sequence over the input (output) alphabet is called an *input* (*output*) sequence. A sequence $(i_1, t_1) \ldots (i_l, t_l)$ of timed inputs is a *timed input* sequence. The set of all timed sequences is denoted I_t^*. We also introduce the function $out_{\mathcal{S}}$ that maps the set $S \times I_t^*$ into the set of output sequences. Given state s and a timed input sequence $\alpha = (i_1, t_1) \ldots (i_l, t_l)$, an output sequence $o_1 \ldots o_l \in out_{\mathcal{S}}(s, \alpha)$ if there exist states $s_1 = s, \ldots, s_{l+1}$ s.t. for each $j \in \{1, \ldots, l\}$ the TFSM $\mathcal{S}$ has a clocked transition $(s_j, (i_j, t_j), o_j, s_{j+1})$ and as usual, we say that the pair $(\alpha, out_{\mathcal{S}}(s, \alpha))$ *can take* the machine $\mathcal{S}$ from state s to state s_{l+1}. A pair "timed_input_sequence_α/output_sequence_β" is a *timed I/O sequence* or a *timed trace* of $\mathcal{S}$ at state s if $\beta = out_{\mathcal{S}}(s, \alpha)$.

If TFSM $\mathcal{S}$ is deterministic then for each state s and each timed input sequence α, the set $out_{\mathcal{S}}(s, \alpha)$ has at most one item. If TFSM $\mathcal{S}$ is complete then the set $out_{\mathcal{S}}(s, \alpha)$ is not empty.

The set of all timed traces of $\mathcal{S}$ at state s is denoted $TTr_{\mathcal{S}}(s)$, also denoted $TTr_{\mathcal{S}}$ for short if s is the initial state of $\mathcal{S}$. As usual, the TFSM $\mathcal{S}$ is *initially connected* if for

each state *s*, there exists a timed trace that can take the machine from the initial state to state *s*.

As usual, the behavior of two TFSMs can be compared using their intersection. The intersection of two TFSMs S and P is not defined at state (*s*,*p*) for a timed input (*i*, *t*) when S and P at states *s* and *p* produce disjoint sets of outputs to this timed input.

Definition 6. Given TFSMs S and P, the *intersection* $S \cap P$ is the largest connected submachine of the TFSM ($S \times P$, I, O, $\lambda_{S \cap P}$, (s_0,p_0)) where $((s,p), i, o, (s',p'), \lceil min_1, max_1 \rceil) \in \lambda_{S \cap P}$ if there are transitions $(s, i, o, s', \lceil min_2, max_2 \rceil) \in \lambda_S$ and $(p, i, o, p', \lceil min_3, max_3 \rceil) \in \lambda_P$ s.t. $\lceil min_2, max_2 \rceil \cap \lceil min_3, max_3 \rceil \neq \emptyset$ and $\lceil min_1, max_1 \rceil = \lceil min_2, max_2 \rceil \cap \lceil min_3, max_3 \rceil$. □

Similar to untimed FSMs [18], a number of compatibility and distinguishability relations can be defined between two complete non-deterministic timed FSMs. The only difference is that these relations are defined w.r.t. timed input sequences.

Definition 7. [1]

- TFSMs S and P are *equivalent* if $TTr_S = TTr_P$; otherwise, the machines are *distinguishable*. A timed input sequence α s.t. $out_S(s_0, \alpha) \neq out_P(p_0, \alpha)$ is said to *distinguish* machines S and P.
- TFSM S is a *reduction* of TFSM P if $TTr_S \subseteq TTr_P$; otherwise, S is not a *reduction* of TFSM P. If a complete TFSM S is not a *reduction* of a complete TFSM P then there exists a timed input sequence α such that $out_S(s_0, \alpha) \not\subset out_P(p_0, \alpha)$ and α is said to *r-distinguish* the TFSM S from the TFSM P.
- TFSMs S and P are *non-separable* if for each timed input sequence α it holds that $out_S(s_0, \alpha) \cap out_P(p_0, \alpha) \neq \emptyset$. If there exists a timed input sequence α s.t. $out_S(s_0, \alpha) \cap out_P(p_0, \alpha) = \emptyset$ then TFSMs S and P are *separable* and α is said to *separate* machines S and P.
- TFSMs S and P are *r-compatible* if there exists a complete TFSM that is a reduction of both machines S and P. If TFSMs S and P are not *r*-compatible then they are *r-distinguishable*. Similar to untimed FSMs, *r*-distinguishable TFSMs are not necessary *r*-distinguishable by a single sequence. □

In this paper, we propose methods for deriving separating and *r*-distinguishing sequences for two complete and observable TFSMs (when such sequences exist). As the number of timed inputs of a complete TFSM is infinite, the methods used for untimed FSMs cannot be directly used.

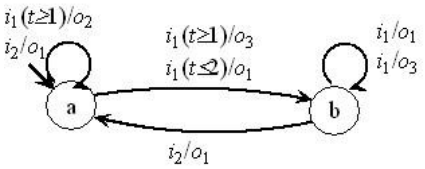

Figure 1.a Timed FSM S

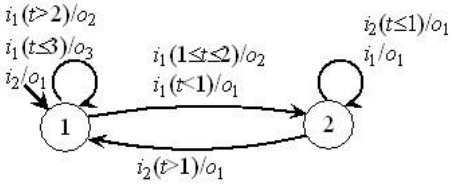

Figure 1.b Timed FSM P

[1] In the same way, the compatibility and distinguishability relations can be introduced for two states of two TFSMs or for two states of a TFSM.

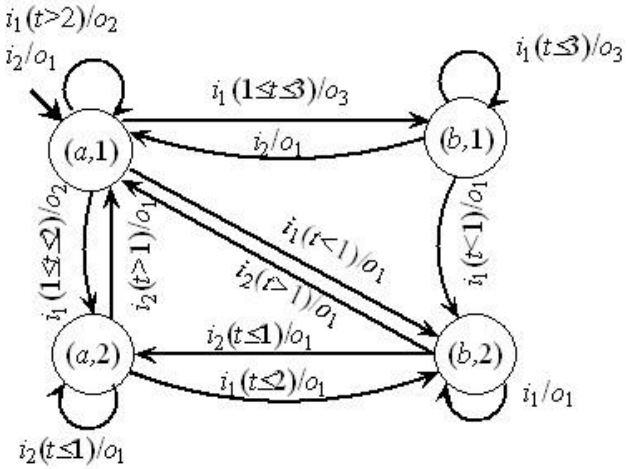

Figure 2. Timed FSM $S \cap P$

3. Separability Relation and Separating Sequences

Similar to untimed FSMs, the separability of TFSMs S and P can be checked by using the intersection $S \cap P$. The following statement holds.

Proposition 2. Given complete TFSMs S and P, if the intersection $S \cap P$ is complete then the TFSMs S and P are non-separable. □

In fact, state s of TFSM S and state p of TFSM P can be separated by a timed input (i, t) if and only if $out_S(s, (i, t)) \cap out_P(p, (i, t)) = \varnothing$. If the intersection $S \cap P$ is complete then for each state (s, p) and each timed input (i, t) it holds that $out_S(s, (i, t)) \cap out_P(p, (i, t)) \neq \varnothing$. Correspondingly, for each timed input sequence α it holds that $out_S(s_0, \alpha) \cap out_P(p_0, \alpha) \neq \varnothing$.

We now present an algorithm for deriving a minimum length separating sequence for two complete observable TFSMs. Algorithm 1 uses the intersection of two partitions. Given two partitions $\Pi_{(q, i)}$ and $\Pi_{(s, i)}$ over $[0, \infty)$, the intersection of these partitions contains non-empty intersections $g \cap h$, $g \in \Pi_{(q, i)}$, $h \in \Pi_{(s, i)}$.

Algorithm 1: **Deriving a minimum length separating sequence of two TFSMs**

Input: Complete observable TFSMs $S = (S, I, O, \lambda_S, s_0)$ and $P = (P, I, O, \lambda_P, p_0)$

Output: A separating sequence of TFSMs S and P (when such a sequence exists)

Derive the intersection $Q = S \cap P$;

If Q is a complete TFSM then END Algorithm 1. TFSMs S and P are non-separable.

Otherwise, assign

$k := 0$;

$Edge := \varnothing$;

$Q_{k0} := \{(s_0, p_0)\}$;

Q_k: = {Q_{k0}};
While

(for some $Q_{kj} \in Q_k$, $j \geq 0$, there exists a timed input (i, t) such that for each state (s,p) of the set Q_{kj}, states s and p are separated by (i, t) ($Rule$ 1)
or
for each $Q_{kj} \in Q_k$, there exists $Q_{am} \in Q_a$, $a < k$, s.t. each state (s,p) $\in Q_{kj}$ is a reduction of some state (s',p') $\in Q_{am}$ (*Rule* 2))

Do:

Q_{k+1}: = ∅;
For each subset $Q_{kj} \in Q_k$, j = 0, ..., |Q_k| - 1, for which there is no $Q_{am} \in Q_a$, $a < k$, s.t. each state (s,p) $\in Q_{kj}$ is a reduction of some state (s',p') $\in Q_{am}$ and for each input i,
Do:

Derive the set Π as the intersection of $\Pi_{(q, i)}$ over all state pairs $q \in Q_{kj}$;
For each guard $g \in \Pi$, derive the set M as the union of (i, g)-successors over all $q \in Q_{kj}$ of the TFSM Q ;
Add M to Q_{k+1};
Add a triple (Q_{kj}, (i, g), M) to the set *Edge*;
Increment k by 1;

If for some $Q_{k\,j_k}$, $j \geq 0$, there exists a timed input (i, t) such that for each state (s,p) of the set Q_{kj}, states s and p are separated by (i, t) ($Rule$ 1) then derive a timed sequence α as follows. Given the set *Edge*, derive the sequence (Q_{00}, (i_1, g_1), $Q_{1\,j_1}$), ($Q_{1\,j_1}$, (i_2, g_2), $Q_{2\,j_2}$), …, ($Q_{(k-1)\,j_{k-1}}$, (i_k, g_k), $Q_{k\,j_k}$) such that ($Q_{(l-1)\,j_{l-1}}$, (i_l, g_l), $Q_{l\,j_l}$) ∈ *Edge* for each $l \in \{1, \ldots, k\}$ and then derive a sequence of timed inputs $\alpha = (i_1, t_1) \ldots (i_k, t_k)$ s.t. $t_j \in g_j$, $j = 1, \ldots, k$. The sequence α is a shortest separating sequence of TFSMs S and P.

If for each $Q_{kj} \in Q_k$, there exists $Q_{am} \in Q_a$, $a < k$, s.t. each state (s,p) $\in Q_{kj}$ is a reduction of some state (s',p') $\in Q_{am}$ then TFSMs S and P are non-separable. □

Proposition 3. If TFSMs S and P are separable then Algorithm 1 returns a shortest separating sequence of S and P. □

In fact, in [20] an algorithm is given for deriving a shortest separating sequence for two untimed FSMs based on the successor tree of the intersection of two FSMs. Algorithm 1 uses also the intersection and successor tree when deriving a shortest separating sequence of two timed FSMs. However, for TFSMs, the number of timed inputs is infinite and thus, each state has an infinite number of timed successors. In order to make this number finite we introduce and then use in Algorithm 1 the notion

of a partition $\Pi_{(q, i)}$. According to Proposition 1, given a state q of the intersection $S \cap P$, an input i, and a region $g \in \Pi_{(q, i)}$, for each $t_1, t_2 \in g$, the set of (i, t_1)- and (i, t_2)-successors of state q coincide. Correspondingly, all such successors coincide with the set of (i, g)-successors of state q.

Proposition 4. Given two complete TFSMs S and P with n and m states, if the machines are separable then there exists a separating sequence with length at most 2^{mn-1} and the upper bound 2^{mn-1} is reachable. □

The first part of the statement is implied by Algorithm 1, as by construction, according to Rule 2, k cannot be greater than $2^{mn-1} + 1$. The second part holds since the upper bound is reachable for untimed FSMs [20] which can be considered as a particular case of timed FSMs where for each pair (s, i) the set $\Pi_{(s, i)}$ has a singleton $[0, \infty)$.

In order to show that the upper bound in Proposition 4 is reachable it is enough to show that is reachable for untimed complete non-deterministic FSMs. For any n and m, there exist observable untimed FSMs S and P with n and m states which can be separated only by a timed input sequence of length 2^{nm-1}. As an example, we can consider such untimed FSMs from [20]; these machines have the input alphabet I, $|I| = 2^{nm-1}$, and the output alphabet O, $|O| = 2nm$. However, determining the minimal number of inputs, for separating two separable machines, such that the upper bound of Proposition 4 is reachable is still an unsolved problem.

Example: As an application example for Algorithm 1, consider TFSMs S (Fig. 1a) and P (Fig. 1b) with initial states **a** and **1** defined over inputs $\{i_1, i_2\}$, outputs $\{o_1, o_2, o_3\}$. The intersection $S \cap P$ is shown in Fig. 2. By definition, the set $Q_0 = \{Q_{00}\}$, where $Q_{00} = \{(a,1)\}$. Given the intersection $S \cap P$, the set $\Pi_{(a1,\ i1)} = \{(0, 1), (1, 2), (2, 3), (3, \infty), \{0\}, \{1\}, \{2\}, \{3\}\}$, and thus, for Q_{00} and i_1, $\Pi = \{(0, 1), (1, 2), (2, 3),(3, \infty), \{0\}, \{1\}, \{2\}, \{3\}\}$, while for Q_{00} and i_2 , $\Pi = \{(0, \infty), \{0\}\}$. Correspondingly, we obtain the set $Edge = \{(Q_{00}, (i_1, 0 < t < 1), \{(b,2)\}); (Q_{00}, (i_1, 1 < t < 2), \{(a,2), (b,1)\}); (Q_{00}, (i_1, 2 < t < 3), \{(a,1), (b,1)\}); (Q_{00}, (i_1, t > 3), \{(a,1)\}); (Q_{00}, (i_1, 0), \{(b,2)\}); (Q_{00}, (i_1, 1), \{(a,2), (b,1)\}); (Q_{00}, (i_1, 2),\{(a,2), (b,1)\}), (Q_{00}, (i_1, 3), \{(a,1), (b,1)\}), (Q_{00}, (i_2, t > 0), \{(a,1)\}), (Q_{00}, (i_2, \{0\}), \{(a,1)\})\}$. Therefore, the set $Q_1 = \{\{(b,2)\}, \{(a,2), (b,1)\}, \{(a,1), (b,1)\}, \{(a,1)\}, \{(b,2)\}, \{(a,1), (b,1)\}, \{(a,1)\}\}$.

For states $(a,1)$ and $(b,2)$, the union of time guards in the intersection $S \cap P$ is $[0, \infty)$ for both inputs i_1 and i_2, and thus, states a and 1 and states b and 2 are not 1-separable. However, we observe that the behavior of the intersection $S \cap P$ is not defined at states $(a,2)$ and $(b,1)$ for timed inputs $(i_1, t > 3)$. Thus, states a and 2 and states b and 1 are separable by a timed input $(i_1, 4)$. Given timed input $(i_1, 1)$, the intersection reaches from the initial state $(a,1)$ states $(a,2)$ and $(b,1)$ and thus, the sequence of timed inputs $(i_1, 1)$ $(i_1, 4)$ separates TFSMs S and P.

In order to distinguish two separable timed FSMs we do not need the "all weather conditions" assumption. It is enough to apply a separating input sequence once since the sets of outputs of the machines to this sequence are disjoint. However, it is well-known that when a common reduction of non-separable complete non-determinisitic untimed FSMs does not exist such machines can be distinguished without "all weather conditions" assumption [18] by a so-called r-distinguishing set. Similar to untimed non-deterministic FSMs, if two timed complete observable FSMs do not

have a common complete reduction then these machines can be distinguished by an adaptive experiment using the *r*-distinguishability relation. In the following section, we present an algorithm for an adaptive experiment that checks the *r*-distinguishability of two observable TFMSs and if the machines are *r*-distinguishable an *r*-distinguishing set is derived.

4. R-distinguishability Relation and *r*-distinguishing Sets

Two complete TFSMs S and P which have no common complete reduction are *r-distinguishable*. If TFSMs S and P have a common complete reduction then these TFSMs are *r-compatible*. Generally the number of pair-wise non-equivalent complete reductions of a timed FSM is infinite and thus, it is not trivial to decide if two complete timed TFSMs are *r*-distinguishable. However, if TFSMs S and P are observable then, similar to observable untimed non-deterministic FSMs, we can use another (equivalent) definition of the *r*-distinguishability relation that helps us when checking *r*-distinguishability by an adaptive experiment.

Given observable timed FSMs S and P and their intersection $Q = S \cap P$, states s and p are 1-*r*-distinguishable if states s and p can be separated by a timed input, i.e. the intersection is partially specified at state $q = (s,p)$. In other words, there exists an input i s.t. in the intersection $S \cap P$ the union Ω of guards over all transitions $((s,p), i, o, (s',p'), g) \in \lambda_{S\cap P}$ is different from $[0, \infty)$. A set $R_{sp} = \{(i, t)/o: o \in out_S(s, (i, t))$ or $o \in out_P(p, (i, t))\}$ where $t \in [0, \infty)\backslash\Omega$, is an *r-distinguishing set* of states s and p. We note that one timed input (i, t) is sufficient for *r*-distinguishing 1-*r*-distinguishable states s and p.

Consider $k > 1$ and assume that all pairs of $(k\text{-}1)$-*r*-distinguishable states are determined and for each pair of $(k\text{-}1)$-*r*-distinguishable s and p an *r-distinguishing set* R_{sp} is also determined. States s and p are *k-r-distinguishable* if these states are $(k\text{-}1)$-*r*-distinguishable or for some input i there exists $t \in [0, \infty)$ such that for each transition $((s,p), i, o, (s',p'), g) \in T_{S\cap P}$, $g \ni t$, states s' and p' are $(k\text{-}1)$-*r*-distinguishable. In this case, an *r*-distinguishing set for states s and p is constructed as the concatenation of $(i, t)/o$, $t \in g$, $o \in out_{S\cap P}((s,p), (i, t))$, with each sequence of each set $R_{s'p'}$ such that $S \cap P$ has the transition $(s,p) \rightarrow (i, t)/o \rightarrow (s',p')$. We refer to such a timed input (i, t) as a *k-r-distinguishing timed input* of states s and p.

Similar to untimed FSMs, it can be shown that observable TFSMs S and P are *r*-distinguishable if there exists an integer k s.t. their initial states are *k-r*-distinguishable. A set of sequences that *r*-distinguish the initial states of TFSMs is an *r-distinguishing* set of TFSMs S and P.

Let observable TFSMs S and P be *r*-distinguishable. Then they can be distinguished based on an *r*-distinguishing set of TFSMs S and P by using an adaptive experiment. For TFSMs with n and m states length of each sequence in the *r*-distinguishing set is at most nm and this upper bound is reachable. Moreover, during an adaptive experiment only one sequence of timed inputs of an *r*-distinguishing set will be

applied to *r*-distinguish considered machines. However, the following proposition shows that the total length of an *r*-distinguishing set can be exponential.

Proposition 5. Given integers n and m, $n \geq 1$, $m \geq 1$, there always exist *r*-distinguishable TFSMs S and P with n and m states s.t. the total length of all sequences of some *r*-distinguishing set is at most $(nm+2)2^{nm-3}$ and this upper bound is reachable. □

In fact, the proposition is a corollary to the similar proposition [24] for untimed FSMs which can be considered as a particular case of timed FSMs where for each pair (s, i) the set $\Pi_{(s, i)}$ has a singleton $[0, \infty)$. However, below we show that similar to untimed FSMs, an *r*-distinguishing set can be represented as the set of traces of a partial timed FSM that has at most $nm + 2$ states and thus, there exists a representation of an *r*-distinguishing set with the polynomial complexity.

Algorithm 2: Deriving an r-distinguishing set of two TFSMs

Input: Complete observable TFSMs $S = (S, I, O, \lambda_S, s_0)$ and $P = (P, I, O, \lambda_P, p_0)$

Output: Partial initially connected TFSM $R_{(S,P)}$ if TFSMs S and P are *r*-distinguishable

Derive the tuple $R = (R, I, O, \lambda_R)$ where, λ_R is empty and R contains two states which we call r_S and r_P;
Derive the intersection $Q = S \cap P$ of TFSMs S and P;
k: = 1;
Q_k: = Q, where Q is the set of states of $S \cap P$;
While $((s_0, p_0) \in Q_k$ and the set Q_k has pairs of *k*-*r*-distinguishable states), do:
Determine all pairs of the set Q_k which have *k*-*r*-distinguishable states;
For each pair (s,p) of the set Q_k s.t. s and p are *k*-*r*-distinguishable
Determine a *k*-*r*-distinguishing timed input (i, t) of states s and p;
Add state (s,p) into set R;
For each $o \in O$ s.t. there is the transition $((s,p), i, o, (s',p'), g) \in \lambda_Q$ where $g \ni t$, add the tuple $((s,p), i, o, (s',p'), [t])$ to λ_R;
For each $o \in O$ s.t. there is no transition $((s, p), i, o, (s',p'), g) \in \lambda_Q$ where $g \ni t$, add to λ_R the tuple $((s,p), i, o, r_S, [t])$ if $o \in out_S(s, (i, t))$. If $o \in out_P(p, (i, t))$ add the tuple $((s,p), i, o, r_P, [t])$;
Delete state (s,p) from the set Q_k;
Increment k by 1;
Q_k: = Q_{k-1};

If $(s_0,p_0) \notin Q_k$ then convert the tuple R into TFSM by claiming state (s_0,p_0) as the initial state of the TFSM. The largest initially connected submachine of TFSM R is TFSM $R_{(S,P)}$; END Algorithm 2.

If states of each pair of Q_k are not k-r-distinguishable then End Algorithm 2. TFSMs S and P are r-compatible, i.e. are not r-distinguishable. □

By construction of TFSM $R_{(S,P)}$, the following statement holds.

Proposition 6. Given two r-distinguishable observable TFSMs S and P with n and m states, Algorithm 1 returns an acyclic partial TFSM $R_{(S,P)}$ such that for each state (s,p) of $R_{(S,P)}$ there exists exactly one input i for which $\Pi_{(s,\ i)}$ is not empty. Moreover, no input is defined at states r_S and r_P. □

According to Proposition 6, if Algorithm 2 returns a TFSM $R_{(S,P)}$ then an r-distinguishing set R of TFSMs S and P is the set of all timed traces, which take the TFSM $R_{(S,P)}$ from the initial state to states r_S and r_P. Correspondingly, the final state of an executed trace uniquely indicates which TFSM S or P is under experiment. In other words, if the final state of an executed trace is r_S (r_P) then the TFSM under experiment is S (P).

Example: As an example of Algorithm 2, consider TFSM S with the initial state 1 and also TFSM S with the initial state 3 (Figure 3). Since in this example we consider two submachines of S starting from initial states 1 and 3, we denote the first machine as S_1 and the second as S_3 and we add into R two states $\boldsymbol{r_{S1}}$ and $\boldsymbol{r_{S3}}$ with subscripts indicating the initial states of the TFSMs. Part of the intersection $Q = S_1 \cap S_3$ is shown in Figure 4. Set $Q_1 = Q$ (for $k = 1$) includes all states of the TFSM Q. States 3 and 2 of Q_1 are 1-r-distinguishable by a timed input (i_2, 1) and states 2 and 4 are 1- r-distinguishable by a timed input (i_1, 2). Thus, we remove states (3, 2) and (2, 4) from Q_1 and obtain Q_2 which does not include states (3, 2) and (2, 4). States 1 and 3 of the initial state (1, 3) in Q_2 are 2-r-distinguishable. By direct inspection, one can observe that states (3, 2) and (2, 4) are reached from the initial state by a timed input (i_1, 3) and thus, TFSM $R_{(S1,\ S3)}$, shown in Figure 5, represents an r-distinguishing set {(i_1, 3)/o_1.(i_2, 1)/o_1, (i_1, 3)/o_1.(i_2, 1)/o_2, (i_1, 3)/o_2.(i_1, 2)/o_1, (i_1, 3)/o_2.(i_1, 2)/o_2}.

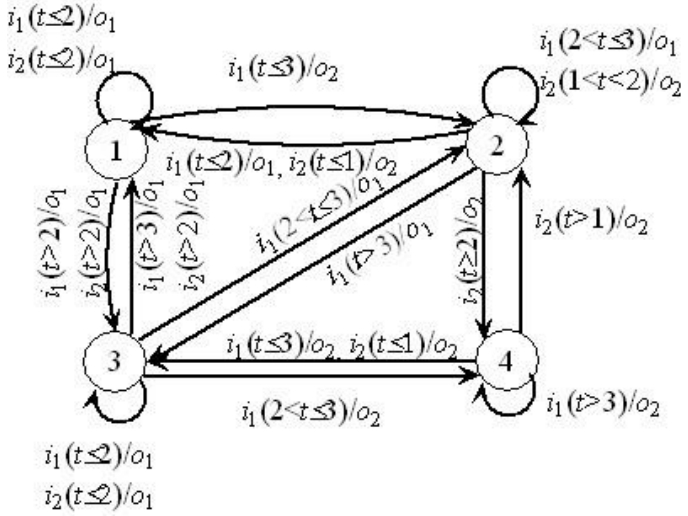

Figure 3. TFSM S where states 1 and 3 are not separable but they are r-distinguishable

$Q = S_1 \cap S_3$	**(1,3)**	**(3,2)**	**(2,4)**	**(2,2)**
$i_1, t \le 2$	(1,3) / o_1	(3,1) / o_1	-	(1,1) / o_1
$i_1, 2 < t \le 3$	(3,2) / o_1 (2,4) / o_2	(2,2) / o_1	-	(2,2) / o_1
$i_1, t > 3$	(3,1) / o_1	(1,3) / o_1	-	(3,3) / o_1
$i_2, t \le 1$	(1,3) / o_1	-	(1,3) / o_2	(1,1) / o_2
$i_2, 1 < t < 2$	(1,3) / o_1	-	(2,2) / o_2	(2,2) / o_2
$i_2, t = 2$	(1,3) / o_1	-	(4,4) / o_2	(4,4) / o_2
$i_2, t > 2$	(1,3) / o_1	-	(4,4) / o_2	(4,4) / o_2

Figure 4. Part of the intersection TFSM $Q = S \cap P$

$R_{(S1,S3)}$	**(1,3)**	**(3,2)**	**(2,4)**	r_{S1}	r_{S3}
$i_1, t=3$	(3,2) / o_1 (2,4) / o_2	-	r_{S1}/o_1	-	-
$i_1, t=2$			r_{S3}/o_2		
$i_2, t = 1$		r_{S1}/o_1	-	-	-
	-	r_{S3}/o_2			

Figure 5. TFSM $R_{(S1,S3)}$

5. Conclusion and Further Research Work

In this paper, we present algorithms for distinguishing timed non-deterministic finite state machines (TFSMs). More precisely, we present a preset algorithm for separating two separable TFSMs and an adaptive algorithm for distinguishing two *r*-distinguishable possibly non-separable TFSMs. The algorithms take into account the fact that in general, unlike untimed FSMs, in a TFSM the number of timed inputs is usually infinite. We also state that the upper bounds on length of distinguishing sequences are as those of untimed FSMs. In this paper, we only consider complete TFSMs where for every state and input action of the TFSM the set of outgoing transitions of the state under the input action is not empty and the time guards of these outgoing transitions are defined over $[0, \infty)$. In order to apply our work to partial TFSMs, one can complete a TFSM in the well-known way: for every state and input action where there is no outgoing transitions under the input action at some time instance, add a self-loop transition to the state with the *Null* output and with a corresponding time guard.

The work presented in this paper can be extended in various ways. For example, the presented algorithms can be used as a basis for test derivation of TFSMs with the guaranteed fault coverage. In addition, the algorithms can be adapted for other distinguishability relations as those defined for untimed non-deterministic FSMs.

References

1. Alur, R, and Dill. D. L.: A Theory of Timed automata. Theoretical Computer Science, 126(2),183--235 (1994)
2. Alur, R., Courcoubetis, C., Yannakakis, M.: Distinguishing Tests for Nondeterministic and Probabilistic Machines. In Proc. the 27th ACM Symposium on Theory of Computing, pp. 363--372 (1995)
3. Bochmann G. v., Petrenko, A.: Protocol Testing: Review of Methods and Relevance for Software Testing. In International Symposium on Software Testing and Analysis, Seattle, pp. 109--123 (1994)
4. Dorofeeva, M., El-Fakih, K., Maag, S., Cavalli, A.R., Yevtushenko, N.: Experimental Evaluation of FSM-Based Testing Methods. In Proc. IEEE Software Engineering and Formal Methods, pp. 23--32 (2005)
5. En-Nouaary, A., Dssouli, R., Khendek, F.: Timed Wp-Method: Testing Real-Time Systems, IEEE TSE 28(11), 1023--1038 (2002)
6. Hierons, R. M.: Testing from a Non-Deterministic Finite State Machine Using Adaptive State Counting. IEEE Transactions on Computers, 53,10, 1330--1342 (2004)
7. Hierons, R. M.: Using Candidates to Test a Deterministic Implementation against a Non-Deterministic Finite State Machine. The Computer Journal, 46,3, 307--318 (2003)
8. Hierons, R. M.: Adaptive Testing of a Deterministic Implementation Against a Nondeterministic Finite State Machine. The Computer Journal, 41(5), 349–355 (1998)
9. Hierons, R. M, Merayo M. G., Nunez: Testing from a Stochastic Timed System with a Fault Model. Journal of Logic and Algebraic Programming, 72(8), 98-115 (2009)
10. Kohavi, Z.: Switching and Finite Automata Theory. McGraw- Hill, New York (1978)

11. Lee, D., Yannakakis, M.: Principles and Methods of Testing Finite State Machines-A Survey. In Proc. of the IEEE, 84(8), 1090--1123 (1996)
12. Luo, G., Petrenko, A., Bochmann, G. v.: Selecting Test Sequences for Partially Specified Nondeterministic Finite State Machines. In Proc. 7th International Workshop on Protocol Test Systems (1994)
13. Milner. R.: A Calculus of Communicating Systems. Lecture Notes in Computer Science, vol 92 (1980)
14. Milner. R.: Communication and Concurrency. Prentice-Hall (1989)
15. Merayo M. G., Nunez, M., Rodriguez I.: Extending EFSMs to Specify and Test Timed Systems with Action Durations and Time-outs. IEEE Transactions on Computers, 57(6), 835—844 (2008)
16. Merayo M. G., Nunez, M., Rodriguez I.: Formal Testing from Timed Finite State Machines. Computer Networks, 52(2), 432--460 (2008)
17. Petrenko, A., Yevtushenko, N., Bochmann, G. v.: Testing Deterministic Implementations from their Nondeterministic Specifications. In Proc. of the IFIP Ninth International Workshop on Testing of Communicating Systems, pp. 125--140 (1996)
18. Petrenko, A., Yevtushenko, N.: Conformance Tests as Checking Experiments for Partial Nondeterministic FSM. In Proceedings of the 5th International Workshop on Formal Approaches to Testing of Software, LNCS vol. 3997, pp. 118—133 (2005)
19. Spitsyna, N.: FSM-based test suite derivation strategies for discrete event systems. Ph.D. Thesis, Tomsk State University, pp. 1--158 (2005)
20. Spitsyna, N., El-Fakih, K., Yevtushenko, N.: Studying the Separability Relation between Finite State Machines. Software Testing, Verification and Reliability, 17(4), 227--241 (2007)
21. Shabaldina, N., El-Fakih, K., Yevtushenko, N:. Testing Nondeterministic Finite State Machines with respect to the Separability Relation. In Proc. of the IFIP 19th International Conference on Testing of Communicating Systems and the 7th International Workshop on Formal Approaches to Testing of Software. Lecture Notes in Computer Science vol. 4581, pp. 305-318 (2007)
22. Springintveld, J., Vaandrager, F., D'Argenio, P.: Testing Timed Automata. Theoretical Computer Science, 254(1-2), 225–257 (2001)
23. Starke, P.: Abstract automata, American Elsevier, 3–419 (1972)
24. Yevtushenko, N., Spitsyna, N.: On the Upper of Length of Separating and r-distinguishing Sequences for Observable Nondeterministic FSMs. In Proc. of Artificial intelligence systems and computer sciences. pp 124--126 (2006) (in Russian)