Acta Cybernetica **21** (2013) 205–222.

Distinguishing Experiments for Timed Nondeterministic Finite State Machines*

Khaled El-Fakih[†], Maxim Gromov[‡], Natalia Shabaldina[‡], and Nina Yevtushenko[‡]

Abstract

The problem of constructing distinguishing experiments is a fundamental problem in the area of finite state machines (FSMs), especially for FSM-based testing. In this paper, the problem is studied for timed nondeterministic FSMs (TFSMs) with output delays. Given two TFSMs, we derive the TFSM intersection of these machines and show that the machines can be distinguished using an appropriate (untimed) FSM abstraction of the TFSM intersection. The FSM abstraction is derived by constructing appropriate partitions for the input and output time domains of the TFSM intersection. Using the obtained abstraction, a traditional FSM-based preset algorithm can be used for deriving a separating sequence for the given TFSMs if these machines are separable. Moreover, as sometimes two non-separable TFSMs can still be distinguished by an adaptive experiment, based on the FSM abstraction we present an algorithm for deriving an r-distinguishing TFSM that represents a corresponding adaptive experiment.

Keywords: nondeterministic untimed and timed finite state machines, preset and adaptive distinguishing experiments, state identification

1 Introduction

Finite State Machines (FSMs) are widely used for modeling systems in many application domains. For instance, (Mealy) FSMs are used as the underlying models for formal description techniques such as SDL and UML State Diagrams. In many cases, the behavior of a given machine can be considered as a mapping of input sequences (sequences of input symbols) to corresponding output sequences (sequences

*This work was partially supported by AUS FRG-III and Russian ministry of Science and High Education (contract No. 14.B37.21.0622)

†American University of Sharjah, Department of Computer Science and Engineering, PO Box 26666, Sharjah, UAE, Tel: (971) 06 5152492, Mobile: (971) 050 3073091 Fax: (971) 6 515 2979, E-mail: kelfakih@aus.edu

‡Tomsk State University, 36 Lenin Str., Tomsk, 634050, Russia, E-mail: gromov@sibmail.com, NataliaMailBox@mail.ru, ninayevtushenko@yahoo.com

DOI: 10.14232/actacyb.21.2.2013.1

of output symbols). A machine is *deterministic* if it produces a single output sequence in response to an input sequence and a machine is *nondeterministic* if it can produce several output sequences in response to an input sequence. Nondeterminism may occur due to various reasons such as limited controllability, abstraction level, modeling concurrency and real time systems, etc. [1, 7, 21].

When distinguishing FSMs, we have a machine under test about which we lack some information, and we want to deduce this information by conducting experiments on this machine. An experiment consists of applying input sequences to the machine, observing corresponding output responses and drawing some conclusions about the machine under test. An experiment is *simple* if a single input sequence is applied to a machine under experiment; otherwise, the experiment is referred to as a *multi* experiment. An experiment is *preset* if input sequences are known before starting the experiment and an experiment is *adaptive* if at each step of the experiment the next input is selected based on previously observed outputs. Distinguishing experiments with FSMs are widely used as a basis for solving fundamental testing problems such as the fault detection (or conformance testing) and/or the machine identification problems. For related surveys and algorithms on FSM-based distinguishing experiments, the reader may refer to [2–4, 9, 11–13].

Unlike deterministic FSMs, for nondeterministic FSMs, there are a number of distinguishability relations, other than the equivalence relation, such as the *non-reduction*, *separability*, and *r-distinguishability* relations [1, 16, 20]. Two machines can be distinguished by a simple preset experiment if these machines are separable. The separability relation is defined by Starke in [20] and studied in [1] and [19]. Two nondeterministic machines are *separable* if there is an input sequence, called a *separating sequence*, such that the sets of output responses of the machines to the input sequence are disjoint. Thus, two separable machines can be distinguished by applying a separating sequence only once. Two complete non-separable machines still can be distinguished by a simple adaptive experiment if they are *r-distinguishable*, i.e., if they have no common complete reduction [17, 23]. A machine is a *reduction* of another machine if its behavior is contained in the behavior of the other machine.

Currently, models of many systems such as telecommunication systems, plant and traffic controllers etc, take into account time constraints and correspondingly timed FSMs are getting a lot of attention. Merayo et al. [5, 14, 15] consider a timed possibly nondeterministic FSM model where time constrains limit a time elapsed when an output has to be produced after an input has been applied to the FSM. Hierons et al. [8] introduce a timed stochastic FSM model. Gromov et al. [6] consider a timed complete nondeterministic FSM model where transitions are guarded by time constraints over a single clock. The clock is reset at the execution of a transition. In this paper, we consider a model similar to that in [6], yet extended to deal with non-zero output delays sometimes called output timeouts. The considered model can be regarded as a temporal extension of FSMs where a transition is fired only if a given input is given in time (bounded by given lower and upper bounds) that is counted from the moment when a current state is reached. Firing a transition also takes time between the reception of the input and the emission of the output, i.e., the output delay represents the transition execution/processing time. In the considered model, the identification of input and output time domains of a state can be done independent of time domains of other states, and thus, there are technical benefits in using the considered model for distinguishability and testing.

Given two possibly nondeterministic timed FSMs, we study the problem of deriving an input sequence that distinguishes these machines. At the first step, the TFSM intersection of the given two machines is derived from which an FSM abstraction is then constructed. It is shown that distinguishing experiments for the given timed FSMs can be determined based on the constructed FSM abstraction. In particular, we show how a traditional preset FSM-based method can be adapted for the FSM abstraction of the intersection when deriving a separating sequence for two given timed FSMs. In addition, using the FSM abstraction we present an algorithm for deriving an r-distinguishing TFSM that represents an adaptive distinguishing experiment for the given two TFSMs if the machines are r-distinguishable.

This paper extends a related preliminary work in [6] to TFSMs which can have non-zero output delays. Moreover, the presented work provides a simpler strategy for deriving distinguishing experiments. In particular, in [6] two TFSMs are distinguished based on their intersection using more complex algorithms that inherit ideas from traditional untimed FSM methods mixed with the derivation of appropriate partitions of input domains for handling time constraints. The strategy proposed in this paper is based on a corresponding (untimed) FSM abstraction of the intersection of two TFSMs and this allows simpler adaptation of existing traditional FSM-based methods for distinguishing TFSMs. The methods presented in this paper and in [6] produce experiments of the same length as the FSM abstraction has the same number of states as the TFSM intersection of the given two machines.

We note that another possible strategy for distinguishing two given TFSMs using algorithms for untimed machines is to first build an FSM abstraction for each of the given machines, derive the intersection of the obtained FSM abstractions, and afterwards, tune traditional FSM-based methods for deriving distinguishing sequences and their corresponding timed sequences using the obtained FSM intersection and the given TFSMs. However, in this case, the number of (abstract) inputs and outputs of the FSM abstractions and their intersection are larger than those derived using our proposed strategy. This is due to the fact that in this case the derivation time domains of inputs and outputs has to be carried out considering all the states of the given machines whereas it is sufficient to consider, as in our approach, pairs of states that appear in the intersection of the given machines.

Finally, it is worth stating that in [10] some work has been presented for distinguishing Timed Input/Output Automata (TIOA) with multiple clocks. Given a TIOA and a clock model, the product of the given automaton with the clock model is transformed into a so-called Bisimulation Quotient Graph, and afterwards, the obtained graph is transformed into a special possibly nondeterministic (untimed) Mealy machine which is actually a tranducer over sequences of abstract inputs and outputs written as regular languages. However, a distinguishing sequence derived from the obtained tranducer in [10] cannot be applied to distinguishing states of the original timed machine since the regular languages (corresponding to sequences of abstract outputs) labeling transitions of the obtained Mealy machine may intersect, and thus, corresponding states of the initial automaton cannot be separated. In addition, the obtained Mealy machine can be non-observable, and thus the traditional FSM method given in [1] cited in [10] cannot be applied.

This paper is organized as follows. Section 2 includes preliminaries and Section 3 presents the FSM abstraction and distinguishability algorithms. Section 4 concludes the paper.

2 Preliminaries

An FSM S[1] is a 5-tuple $\langle S, I, O, \lambda_{\mathsf{S}}, \hat{s}\rangle$, where S, I and O are finite sets of states, inputs and outputs, respectively, $\hat{s}$ is the initial state and $\lambda_{\mathsf{S}} \subseteq S \times I \times O \times S$ is a transition relation. A timed FSM (TFSM) S or simply a timed machine is a 5-tuple $\langle S, I, O, \lambda_{\mathsf{S}}, \hat{s}\rangle$ with the transition relation $\lambda_{\mathsf{S}} \subseteq S \times (I \times \Pi) \times (O \times \aleph) \times S$, where Π is the set of input time guards and $\aleph$ is the set of output time guards for representing output delays. Each guard $g \in \Pi = \lceil min, max\rceil$ (each guard $f \in \aleph = \lceil min, max\rceil$) where min is a nonnegative integer, while max is a nonnegative integer or the infinity, $min \leqslant max$, and $\lceil \in \{(, [\}$ while $\rceil \in \{),]\}$. From the practical point of view, we assume that all the output guards have a finite upper bound **B**. For every pair $\langle s, i\rangle \in S \times I$, we use $G_{\langle s,i\rangle}$ to denote the collection of input time guards g such that there is a transition $\langle s, \langle i, g\rangle, \langle o, f\rangle, s'\rangle \in \lambda_{\mathsf{S}}$ and for every pair $\langle s, o\rangle \in S \times O$ we use $G_{\langle s,o\rangle}$ to denote the collection of output time guards f such that there is a transition $\langle s, \langle i, g\rangle, \langle o, f\rangle, s'\rangle \in \lambda_{\mathsf{S}}$.

The behavior of a TFSM S can be described as follows. If $\langle s, \langle i, g\rangle, \langle o, f\rangle, s'\rangle \in$ $\in \lambda_{\mathsf{S}}$, where $g = \lceil m_1, m_2\rceil$ and $f = \lceil n_1, n_2\rceil$, we say that TFSM S being at state s *accepts* input i applied at time $t \in \lceil m_1, m_2\rceil$ measured from the moment TFSM S entered state s; the clock then is set to zero, and S *responds* with (or *produces*) output o after t' time units, $t' \in \lceil n_1, n_2\rceil$, and time is set to zero as S enters state s'.

A TFSM S is *observable* if for each two transitions $\langle s, \langle i, \lceil m_1, m_2\rceil\rangle, \langle o, \lceil n_1, n_2\rceil\rangle, s'\rangle \in \lambda_{\mathsf{S}}$ and $\langle s, \langle i, \lceil m_1', m_2'\rceil\rangle, \langle o, \lceil n_1', n_2'\rceil\rangle, s''\rangle \in \lambda_{\mathsf{S}}$ it holds that if $\lceil m_1, m_2\rceil \cap \lceil m_1', m_2'\rceil \neq \varnothing$ and $\lceil n_1, n_2\rceil \cap \lceil n_1', n_2'\rceil \neq \varnothing$, then $o' = o$ implies $s' = s''$. In this paper, we consider only observable TFSMs as similar to untimed FSMs, for every unobservable timed machine there exists an observable timed machine that has the same behavior.

TFSM S is (*time*) *deterministic* if for each two transitions $\langle s, \langle i, \lceil m_1, m_2\rceil\rangle, \langle o, \lceil n_1, n_2\rceil\rangle, s'\rangle \in \lambda_{\mathsf{S}}$, $\langle s, \langle i, \lceil m_1', m_2'\rceil\rangle, \langle o', \lceil n_1', n_2'\rceil\rangle, s''\rangle \in \lambda_{\mathsf{S}}$, $\lceil m_1, m_2\rceil \cap \lceil m_1', m_2'\rceil = \varnothing$. Otherwise, S is (*time*) *nondeterministic.*

TFSM S is *complete* if each input is a defined at each state and for each pair $\langle s, i\rangle \in S \times I$ of S, it holds that the union of all $g \in G_{\langle s,i\rangle}$ equals $[0, \infty)$; otherwise, the machine is called *partial.* A partial machine can be completed by adding appropriate self-loop transitions. In particular, for every time domain g where an input i

[1]If there is no ambiguity we will use the notation S for an FSM and S for its set of states.

at state s is not defined, a self-loop transition $\langle s, \langle i, g\rangle, \langle o, [0, \infty)\rangle, s\rangle$ is added. Consequently, in this paper, we study distinguishing experiments with nondeterministic complete TFSMs.

Given TFSMs $\mathsf{S} = \langle S, I, O, \lambda_{\mathsf{S}}, \hat{s}\rangle$ and $\mathsf{P} = \langle P, I, O, \lambda_{\mathsf{P}}, \hat{p}\rangle$, the *intersection* $\mathsf{S}\cap\mathsf{P}$ is the largest connected submachine of the TFSM $\langle S \times P, I, O, \lambda_{\mathsf{S}\cap\mathsf{P}}, \langle \hat{s}, \hat{p}\rangle\rangle$ where $\langle\langle s, p\rangle, \langle i, \lceil m_1, m_2\rceil\rangle, \langle o, \lceil n_1, n_2\rceil\rangle, \langle s', p'\rangle\rangle \in \lambda_{S\cap P}$ if and only if there are transitions $\langle s, \langle i, \lceil m'_1, m'_2\rceil\rangle, \langle o, \lceil n'_1, n'_2\rceil\rangle, s'\rangle \in \lambda_{\mathsf{S}}$ and $\langle p, \langle i, \lceil m''_1, m''_2\rceil\rangle, \langle o, \lceil n''_1, n''_2\rceil\rangle, p'\rangle \in \lambda_{\mathsf{P}}$ such that $\lceil m'_1, m'_2\rceil \cap \lceil m''_1, m''_2\rceil = \lceil m_1, m_2\rceil$ and $\lceil n'_1, n'_2\rceil \cap \lceil n''_1, n''_2\rceil = \lceil n_1, n_2\rceil$. As a running example, consider TFSM S (Figure 1) with the initial state 1 (hereafter denoted S_1) and the TFSM S with the initial state 3 (hereafter denoted S_3). In the figures, a transition $\langle s, \langle i, \lceil m_1, m_2\rceil\rangle, \langle o, \lceil n_1, n_2\rceil\rangle, s'\rangle$ is depicted as s (column), i (row), and corresponding entry $(\lceil m_1, m_2\rceil), s'/\langle o, \lceil n_1, n_2\rceil\rangle$. The intersection $\mathsf{Q} = \mathsf{S}_1 \cap \mathsf{S}_3$ is shown in Figure 2.

S	1	2	3	4
i_1	$(t \leqslant 2), 1/\langle o_1, t < 3\rangle$ $(t \leqslant 3), 2/\langle o_2, 0 \leqslant t < 5\rangle$ $(t > 2), 3/\langle o_1, 0 \leqslant t < 5\rangle$	$(t \leqslant 2), 1/\langle o_1, 0 \leqslant t < 5\rangle$ $(2 < t \leqslant 3), 2/\langle o_1, 0 \leqslant t < 5\rangle$ $(t > 3), 3/\langle o_1, 0 < t < 5\rangle$	$(t \leqslant 2), 3/\langle o_1, t > 2\rangle$ $(t > 3), 1/\langle o_1, 0 \leqslant t < 5\rangle$ $(2 < t \leqslant 3), 2/\langle o_1, t < 2\rangle$ $(2 < t \leqslant 3), 4/\langle o_2, 0 \leqslant t < 5\rangle$	$(t \leqslant 3), 3/\langle o_2, 0 \leqslant t < 5\rangle$ $(t > 3), 1/\langle o_1, 0 \leqslant t < 5\rangle$
i_2	$(t \leqslant 2), 1/\langle o_1, 0 \leqslant t < 5\rangle$ $(t > 2), 3/\langle o_1, 0 \leqslant t < 5\rangle$	$(t \leqslant 1), 1/\langle o_2, 0 \leqslant t < 5\rangle$ $(1 < t < 2), 2/\langle o_2, 0 \leqslant t < 5\rangle$ $(t \geqslant 2), 4/\langle o_2, 0 \leqslant t < 5\rangle$	$(t \leqslant 2), 3/\langle o_1, 0 \leqslant t < 5\rangle$ $(t > 2), 1/\langle o_1, 0 \leqslant t < 5\rangle$	$(t \leqslant 1), 3/\langle o_2, 0 \leqslant t < 5\rangle$ $(t > 1), 2/\langle o_2, 0 \leqslant t < 5\rangle$

Figure 1: TFSM S, TFSM S_1 is S with initial state 1, and TFSM S_3 is S with initial state 3

$\mathsf{S}_1 \cap \mathsf{S}_3$	$\langle 1, 3\rangle$	$\langle 3, 2\rangle$	$\langle 2, 4\rangle$	$\langle 2, 2\rangle$
i_1	$(t \leqslant 2), \langle 1, 3\rangle/\langle o_1, 2 < t < 3\rangle$ $(2 < t \leqslant 3), \langle 3, 2\rangle/\langle o_1, t < 2\rangle$ $(t > 3), \langle 3, 1\rangle/\langle o_1, 0 \leqslant t < 5\rangle$ $(2 < t \leqslant 3), \langle 2, 4\rangle/\langle o_2, 0 \leqslant t < 5\rangle$	$(t \leqslant 2), \langle 3, 1\rangle/\langle o_1, 0 < t < 5\rangle$ $(2 < t \leqslant 3), \langle 2, 2\rangle/\langle o_1, t < 2\rangle$ $(t > 3), \langle 1, 3\rangle/\langle o_1, 0 \leqslant t < 5\rangle$		$(t \leqslant 2), \langle 1, 1\rangle/\langle o_1, 0 \leqslant t < 5\rangle$ $(2 < t \leqslant 3), \langle 2, 2\rangle/\langle o_1, 0 \leqslant t < 5\rangle$ $(t > 3), \langle 3, 3\rangle/\langle o_1, 0 \leqslant t < 5\rangle$
i_2	$(t \leqslant 2), \langle 1, 3\rangle/\langle o_1, 0 \leqslant t < 5\rangle$ $(t > 2), \langle 3, 1\rangle/\langle o_1, 0 \leqslant t < 5\rangle$		$(t \leqslant 1), \langle 1, 3\rangle/\langle o_2, 0 \leqslant t < 5\rangle$ $(1 < t < 2), \langle 2, 2\rangle/\langle o_2, 0 \leqslant t < 5\rangle$ $(t \geqslant 2), \langle 4, 2\rangle/\langle o_2, 0 \leqslant t < 5\rangle$	$(t \leqslant 1), \langle 1, 1\rangle/\langle o_2, 0 \leqslant t < 5\rangle$ $(1 < t < 2), \langle 2, 2\rangle/\langle o_2, 0 \leqslant t < 5\rangle$ $(t \geqslant 2), \langle 4, 4\rangle/\langle o_2, 0 \leqslant t < \infty\rangle$

Figure 2: The intersection TFSM $\mathsf{Q} = \mathsf{S}_1 \cap \mathsf{S}_3$

Given a TFSM S, a pair $\langle i, t\rangle/\langle o, t'\rangle$, where $i \in I$, $o \in O$, t and t' are non-negative rational numbers, is a *timed input-output* pair where $\langle i, t\rangle$ is a *timed input* that states that input i is applied at time t measured from the moment when the machine entered its current state and $\langle o, t'\rangle$ is a *timed output* that states that output o is produced at time t' measured from the moment when the timed input $\langle i, t\rangle$ has been applied.

Consider a TFSM S and a timed input-output pair $\langle i, t\rangle/\langle o, t'\rangle$. Given a state s, there is a *clocked transition* $\langle s, \langle i, t\rangle, \langle o, t'\rangle, s'\rangle$ in S if λ_{S} has a transition $\langle s, \langle i, g\rangle, \langle o, f\rangle, s'\rangle \in \lambda_{\mathsf{S}}$ such that $t \in g$ and $t' \in f$. A timed input-output pair $\langle i, t\rangle/\langle o, t'\rangle$ is a timed input-output pair at state s if there exists a clocked transition $\langle s, \langle i, t\rangle, \langle o, t'\rangle, s'\rangle$ in S.

A sequence of timed input-output pairs is a *timed trace*. A *timed trace* $\alpha/\beta = \langle i_1, t_1\rangle/\langle o_1, t'_1\rangle, \ldots, \langle i_k, t_k\rangle/\langle o_k, t'_k\rangle$ is a *timed trace* at state s if there exist states $s_1, \ldots, s_{k+1}$ such that $s_1 = s$ and for each $j = 1, \ldots, k$, there exists a clocked transition $\langle s_j, \langle i_j, t_j\rangle, \langle o_j, t'_j\rangle, s_{j+1}\rangle$ in S.

By the above definition, given a timed trace $\alpha/\beta = \langle i_1, t_1\rangle/\langle o_1, t'_1\rangle, \ldots, \langle i_k, t_k\rangle/\langle o_k, t'_k\rangle$ at state s, we assume that the input sequence α is applied to the TFSM in the following way. For each j, $1 \leqslant j \leqslant k$, the input i_j is applied at the time instance t_j measured from the time when the TFSM entered the state s_j, the clock starts advancing from 0 and the output o_j is produced at time t'_j.

A timed input sequence α is *defined* at state s if and only if at state s there exists a timed trace α/β for some timed output sequence β.

A TFSM $\mathsf{S} = \langle S, I, O, \lambda_{\mathsf{S}}, \hat{s}\rangle$ is a *submachine* of TFSM $\mathsf{P} = \langle P, I, O, \lambda_{\mathsf{P}}, \hat{p}\rangle$ if $S \subseteq P$, $\hat{s} = \hat{p}$, and each clocked transition $\langle s, \langle i, t\rangle, \langle o, t'\rangle, s'\rangle$ of S is a clocked transition of P.

Two complete TFSMs S and P are *separable* if there exists a timed input sequence for both TFSMs such that the sets of timed output responses to this input sequence do not intersect and in addition, S and P are *r-distinguishable* if for each complete TFSM M it holds that there exists a timed input sequence α such that the set of output responses of M to α is not a subset of responses of S to α or of responses of P to α.

3 Distinguishing Timed Finite State Machines

Given two TFSMs S and P, in order to distinguish these machines, as usual, we first derive the TFSM intersection $\mathsf{Q} = \mathsf{S} \cap \mathsf{P}$. Given the intersection Q, an abstract FSM $\mathsf{A}(\mathsf{Q})$ is then constructed for which we can apply the traditional FSM distinguishability algorithms when deriving distinguishing sequences over abstract inputs; the distinguishing sequences are then transformed into timed sequences over timed inputs using the established correspondence between Q and $\mathsf{A}(\mathsf{Q})$.

3.1 Deriving an FSM Abstraction

Given TFSM $\mathsf{Q} = \mathsf{S} \cap \mathsf{P}$, an FSM abstraction $\mathsf{A}(\mathsf{Q})$ of Q is derived as follows. For each input $i \in I$ of Q, the collection G_i of time guards over all states with an input i and the corresponding partition Π_i over $[0, \infty)$ is constructed. There is an input $\langle i, g\rangle$ in the abstraction if and only if $g \in \Pi_i$. More precisely, given input $i \in I$, 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 of collection G_i. The finite set Π_i is defined as the (finite) set $\{(j_1, j_2), \ldots, (j_{m-1}, j_m), (j_m, \infty), \{j_1\}, \{j_2\}, \{j_3\}, \ldots \{j_m\}\}$, i.e., the set Π_i has singletons all boundaries and all (infinite) domains with consecutive boundaries of the set G. For each state $q \in Q$ and each $g_j \in \Pi_i$, the abstraction $\mathsf{A}(\mathsf{Q})$ has a transition from state q under abstract input $\langle i, g_j\rangle$ if and only if it holds that there exists a transition $\langle q, \langle i, g\rangle, \langle o, f\rangle, q'\rangle \in \lambda_{\mathsf{Q}}$ such that g contains g_j. For our running example, Π_{i_1} of TFSM Q in Figure 2 equals $\{\{0\},(0,2),\{2\},(2,3),\{3\},(3,\infty)\}$ and $\Pi_{i_2} = \{\{0\},(0,1),\{1\},(1,2),\{2\},(2,\infty)\}$.

Proposition 1. *Given a TFSM $\mathsf{Q} = \langle Q, I, O, \lambda_{\mathsf{Q}}, \hat{q}\rangle$, an input $i \in I$ and a set Π_i of time domains for the input i, let $g \in \Pi_i$ and $t_1, t_2 \in g$. For each $q \in Q$, there is a clocked transition $\langle q, \langle i, t_1\rangle, \langle o, f\rangle, q'\rangle \in \lambda_{\mathsf{Q}}$ if and only if there is a clocked transition $\langle q, \langle i, t_2\rangle, \langle o, f\rangle, q'\rangle \in \lambda_{\mathsf{Q}}$.*

Similarly, the partition Π_o of output guards is derived. For each output $o \in O$ of Q, the collection F_o based on the collections $F_{\langle q,o\rangle}$ over all states where the output o can be produced is derived. An output o *can be produced at time instances* $t \in f$ if and only if there exists a state q and pair $\langle i, g\rangle$ such that $\langle q, \langle i, g\rangle, \langle o, f\rangle, q'\rangle \in \lambda_{\mathsf{Q}}$. Let now $F = \{j_1 = 0, j_2, \ldots, j_m\}$, $j_a < j_{a+1}$, $a = 1, \ldots, m-1$, be the finite ordered set of boundaries of guards of the collection F_o. Based on F the (finite) set $\Pi_o = \{(j_1, j_2), \ldots, (j_{m-1}, j_m), (j_m, \mathbf{B}), \{j_1\}, \{j_2\}, \{j_3\}, \ldots, \{j_m\}\}$ is built, i.e., the set Π_o has singletons for all boundaries and all (infinite) domains with consecutive boundaries of the set F where the output o can be produced. In our running example, Π_{o_1} of TFSM Q (Figure 2) equals $\{\{0\},(0,2),\{2\},(2,3),\{3\},(3,5)\}$ and $\Pi_{o_2} = \{\{0\},(0,5)\}$.

Proposition 2. *Given a TFSM $\mathsf{Q} = \langle Q, I, O, \lambda_{\mathsf{Q}}, \hat{q}\rangle$, an output $o \in O$ and a set Π_o of output domains for the output o, let $f \in \Pi_o$ and $t', t'' \in f$. For each $q \in Q$ and a timed input $\langle i, t\rangle$, either TFSM Q cannot produce both timed outputs $\langle o, t'\rangle$ and $\langle o, t''\rangle$ at state q under $\langle i, t\rangle$ or there is a clocked transition $\langle q, \langle i, t\rangle, \langle o, t'\rangle, q'\rangle \in \lambda_{\mathsf{Q}}$ if and only if there is a clocked transition $\langle q, \langle i, t\rangle, \langle o, t''\rangle, q'\rangle \in \lambda_{\mathsf{Q}}$.*

Given TFSMs S and P, the TFSM intersection $\mathsf{Q} = \langle Q, I, O, \lambda_{\mathsf{Q}}, \hat{q}\rangle$ of S and P, and partitions Π_i and Π_o, a corresponding abstract FSM $\mathsf{A}(\mathsf{Q}) = \langle Q, I_{\mathsf{A}(\mathsf{Q})}, O_{\mathsf{A}(\mathsf{Q})}, \lambda_{\mathsf{A}}, \hat{q}\rangle$ of the intersection can be derived as follows. The FSM $\mathsf{A}(\mathsf{Q})$ has the same set of states and the same initial state as Q, and $\mathsf{A}(\mathsf{Q})$ has (abstract) inputs $I_{\mathsf{A}(\mathsf{Q})} = \{\langle i, g\rangle : i \in I, g \in \Pi_i\}$, (abstract) outputs $O_{\mathsf{A}(\mathsf{Q})} = \{\langle o, f\rangle : o \in O, f \in \Pi_o\}$ and transition relation λ_{A}; there is a transition $\langle q, \langle i, g\rangle, \langle o, f\rangle, q'\rangle$ in λ_{A} if and only if there is a transition $\langle q, \langle i, g'\rangle, \langle o, f'\rangle, q'\rangle \in \lambda_{\mathsf{Q}}$ such that $g \subseteq g'$ and $f \subseteq f'$. Considering the running example, abstract inputs of $\mathsf{A}(\mathsf{Q})$ are the pairs from $\{i_1\} \times \Pi_{i_1}$ and $\{i_2\} \times \Pi_{i_2}$ and abstract outputs are the pairs from $\{o_1\} \times \Pi_{o_1}$ and $\{o_2\} \times \Pi_{o_2}$. A fragment of $\mathsf{A}(\mathsf{Q})$ for the TFSM Q in Figure 2 is shown in Figure 3.

Based on the above construction, the following statements can be established.

Proposition 3. *The following statements hold.*

1. *(a) If TFSMs S and P are observable then TFSM $\mathsf{Q} = \mathsf{S} \cap \mathsf{P}$ is observable. (b) TFSM Q is observable if and only if FSM $\mathsf{A}(\mathsf{Q})$ is observable.*

2. *Given a state q of TFSM Q, a timed input-output pair $\langle i, t\rangle / \langle o, t'\rangle$ is defined at state q if and only if there exists a transition $\langle q, \langle i, g\rangle, \langle o, f\rangle, q'\rangle$ in the abstract FSM such that $t \in g$ and $t' \in f$. Moreover, given a defined (abstract)*

input-output pair $\langle i, g\rangle/\langle o, f\rangle$ *at state* q *of the FSM* $A(Q)$, $t_1, t_2 \in g$, $t'_1, t'_2 \in f$, *there is a clocked transition* $\langle q, \langle i, t_1\rangle, \langle o, t'_1\rangle, q'\rangle \in \lambda_Q$ *if and only if there is a clocked transition* $\langle q, \langle i, t_2\rangle, \langle o, t'_2\rangle, q'\rangle \in \lambda_Q$.

3. *Given an abstract input-output sequence* $\langle i_1, g_1\rangle/\langle o_1, f_1\rangle \ldots \langle i_k, g_k\rangle/\langle o_k, f_k\rangle$ *at state* q *of the FSM* $A(Q)$, *each timed input-output sequence* $\langle i_1, t_1\rangle/\langle o_1, t'_1\rangle \ldots \langle i_k, t_k\rangle/\langle o_k, t'_k\rangle$ *such that* $t_j \in g_j$, $t'_j \in f_j$, $j = 1, \ldots, k$, *is a timed input-output sequence at state* q *of TFSM* Q*, and vice versa, given a timed trace* $\langle i_1, t_1\rangle/\langle o_1, t'_1\rangle \ldots \langle i_k, t_k\rangle/\langle o_k, t'_k\rangle$ *at state* q *of TFSM* Q *there always exists a defined input sequence* $\langle i_1, g_1\rangle/\langle o_1, f_1\rangle \ldots \langle i_k, g_k\rangle/\langle o_k, f_k\rangle$ *at state* q *of the FSM* $A(Q)$ *such that* $t_j \in g_j$, $t'_j \in f_j$, $j = 1, \ldots, k$.

4. *TFSM* Q *has a timed trace* $\langle i_1, t_1\rangle/\langle o_1, t'_1\rangle \ldots \langle i_k, t_k\rangle/\langle o_k, t'_k\rangle$ *at state* q *if and only if the FSM* $A(Q)$ *has a trace* $\langle i_1, g_1\rangle/\langle o_1, f_1\rangle \ldots \langle i_k, g_k\rangle/\langle o_k, f_k\rangle$ *such that* $t_j \in g_j$, $t'_j \in f_j$, $j = 1, \ldots, k$, *at state* s.

Proof. 1. (a) If TFSMs S and P are observable, then for every two timed transitions $\langle s, \langle i, t\rangle, \langle o, t'\rangle, s'\rangle \in \lambda_S$, $\langle s, \langle i, t\rangle, \langle o, t'\rangle, s''\rangle \in \lambda_S$ (or $\langle p, \langle i, t\rangle, \langle o, t'\rangle, p'\rangle \in \lambda_P$, $\langle p, \langle i, t\rangle, \langle o, t'\rangle, p''\rangle \in \lambda_P$) it holds that $s' = s''$ (or correspondingly $p' = = p''$). Thus, there are no timed transitions $\langle\langle s, p\rangle, \langle i, t\rangle, \langle o, t'\rangle, \langle s', p'\rangle\rangle \in \lambda_Q$ and $\langle\langle s, p\rangle, \langle i, t\rangle, \langle o, t'\rangle, \langle s'', p''\rangle\rangle \in \lambda_Q$ such that $\langle s', p'\rangle \neq \langle s'', p''\rangle$.

(b) TFSM Q is observable if and only if for every two timed transitions $\langle q, \langle i, t\rangle, \langle o, t'\rangle, q'\rangle \in \lambda_Q$ and $\langle q, \langle i, t\rangle, \langle o, t'\rangle, q''\rangle \in \lambda_Q$ it holds that $q' = q''$. Correspondingly, by construction of the FSM $A(Q)$, for each defined input $\langle i, g\rangle$ at state q of the FSM $A(Q)$ it holds that there are no two transitions $\langle q, \langle i, g\rangle, \langle o, f\rangle, q'\rangle \in \lambda_A$ and $\langle q, \langle i, g'\rangle, \langle o, f'\rangle, q''\rangle \in \lambda_A$ such that $g \cap g' \neq \varnothing$, $f \cap f' \neq \varnothing$ while $q' \neq q''$, i.e., FSM $A(Q)$ is observable if and only if TFSM Q is observable.

2. Statement 2 of the above proposition is a direct corollary to the definition of time domains.

3. Statement 3 can be shown by induction on the length of a defined input sequence.

4. Statement 4 is implied by the definition of the FSM $A(Q)$ and Statement 3. □

We recall that an abstract FSM $A(Q)$ and TFSM Q have the same number of states, while, $A(Q)$ has more transitions as it has more inputs. However, the number of transitions of an $A(Q)$ is polynomial w.r.t. the number of transitions of Q as it mainly depends on the number of (abstract) inputs $I_{A(Q)}$ which is of order $|I| \cdot m$ where m is the maximum number of items of partitions Π_i.

3.2 Deriving an r-distinguishing TFSM

In order to check whether nondeterministic machines S and P can be distinguished by an adaptive experiment a so-called r-distinguishing machine can be used. The derivation of such a machine is described in [5, 16] for complete untimed FSMs and in [6] for complete TFSMs S and P without output delays. In this paper, such a machine is derived based on the abstraction $A(Q)$ for TFSMs S and P with output delays.

Similar to FSMs [5, 16, 17], an adaptive experiment is represented by a special acyclic so-called single-input output-complete TFSM. Given complete observable TFSMs $S = \langle S, I, O, \lambda_S, \hat{s} \rangle$ and $P = \langle P, I, O, \lambda_P, \hat{p} \rangle$, let $R = \langle R, I, O, \lambda_R, \hat{r} \rangle$ be an acyclic initially connected TFSM such that the set R of states has two designated deadlock states called r_S and r_P. If after the experiment the machine R reaches state r_S then the TFSM under experiment is S while if the final state is r_P then the TFSM under experiment is P. Only one timed input $\langle i, t \rangle$ is defined at each other state of R with all possible outputs, i.e., TFSM R represents an adaptive experiment with a TFSM over input alphabet I and output alphabet O. TFSM R is an r-distinguishing TFSM $R_{(S,P)}$ of S and P (or TFSM $R_{(S,P)}$ r-distinguishes TFSM S and P) if for each state $\langle s, r \rangle$ of the intersection $S \cap R_{(S,P)}$ it holds that $r \neq r_P$ and for each $\langle p, r \rangle$ of the intersection $P \cap R_{(S,P)}$ it holds that $r \neq r_S$.

Similar to FSMs [16], here, we define the notion of *k-undefined* states in order to derive $R(S, P)$ using $A(Q)$. Given (complete observable) TFSMs S and P, $Q = S \cap P$, and FSM abstraction $A(Q)$, state $q = \langle s, p \rangle$ of $A(Q)$ is 1-undefined if there exists an undefined (abstract) input $\langle i, g \rangle$ at state q. Consider $k > 1$ and assume that all $(k-1)$-undefined states of $A(Q)$ are determined. State $q = \langle s, p \rangle$ of $A(Q)$ is k-undefined if q is $(k-1)$-undefined or there exists an abstract input $\langle i, g \rangle$ defined at state q such that for each transition $\langle q, \langle i, g \rangle, \langle o, f \rangle, q' \rangle \in \lambda_A$, each state q' is $(k-1)$-undefined. It can be shown as in [16], that given complete observable TFSMs S and P, these TMSMs are r-distinguishable iff there exists an integer k such that the initial state of the abstraction $A(Q)$ is k-undefined for some $k > 0$.

We use Algorithm 1 in order to derive an r-distinguishing TFSM for two given TFSMs S and P based on the abstract FSM $A(Q)$ of $Q = S \cap P$. If an r-distinguishing FSM over abstract inputs of $A(Q)$ is derived, then the machine is converted to corresponding timed inputs in order to represent an r-distinguishing TFSM for TFSMs S and P.

Based on the TFSM $R_{(S,P)}$ an adaptive experiment for distinguishing TFSMs S and P can be performed in the following way. Given TFSM under test, which is either TFSM S or P, the experiment starts at the initial state $\hat{r} = \hat{q}$ of TFSM $R_{(S,P)}$. At any state of $R_{(S,P)}$ only one timed input $\langle i, t \rangle$ is defined, in addition, any state of $R_{(S,P)}$ is always reached at time $t = 0$. Thus, when reaching a current state r of $R_{(S,P)}$ the clock advances from 0 and the only defined input $\langle i, t \rangle$ is applied to a TFSM under test. In response, the TFSM under test produces a timed output $\langle o, t' \rangle$, $t' \in f$, and accordingly the TFSM $R_{(S,P)}$ moves from a current state r to the next state r' according to the clocked transition $\langle r, \langle i, [t, t] \rangle, \langle o, f \rangle, r' \rangle$. The procedure terminates when the TFSM $R_{(S,P)}$ reaches one of the deadlock states r_S

Algorithm 1 Deriving an r-distinguishing TFSM of two TFSMs

Input: Complete observable TFSMs $\mathsf{S} = \langle S, I, O, \lambda_{\mathsf{S}}, \hat{s}\rangle$ and $\mathsf{P} = \langle P, I, O, \lambda_{\mathsf{P}}, \hat{p}\rangle$
Output: A distinguishing TFSM $\mathsf{R}_{(\mathsf{S},\mathsf{P})}$ if TFSMs S and P are r-distinguishable

$\mathsf{Q} := \mathsf{S} \cap \mathsf{P}$;
derive the FSM abstraction $\mathsf{A}(\mathsf{Q})$;
$\mathsf{R} := \langle R, I, O, \lambda_{\mathsf{R}}\rangle$, where initially λ_{R} is empty and R contains two deadlock states r_{S} and r_{P};
$k := 1$;
$Q_k := Q$; *//Q is the set of states of TFSM Q which are pairs of states of S and P*
while ($\hat{q} \in Q_k$ **and** the set Q_k has k-undefined states) **do**
  determine all states of the set Q_k which are k-undefined in $\mathsf{A}(\mathsf{Q})$;
  for all k-undefined states $q = \langle s, p\rangle$ of the set Q_k **do**
    if (k == 1) **then**
      determine an abstract input $\langle i, g\rangle$ such that it is undefined at state q;
    else
      determine an abstract input $\langle i, g\rangle$ such that for each transition $\langle q, \langle i, g\rangle, \langle o, f\rangle, q'\rangle \in \lambda_{\mathsf{Q}}$, state q' is $(k-1)$-undefined;
    end if
    add state q into the set R;
    for all abstract outputs $\langle o, f\rangle$ **do**
      if there is a transition $\langle q, \langle i, g\rangle, o, f, q'\rangle \in \lambda_{\mathsf{A}}$ **then** *//implies that $k > 1$*
        add to λ_{R} the tuple $\langle(q, \langle i, [t, t]\rangle, \langle o, f\rangle, q'\rangle$, $t \in g$;
      else
        add to λ_{R} the tuple $\langle q, \langle i, [t, t]\rangle, \langle o, f\rangle, r_{\mathsf{S}}\rangle$ if for each $t \in g$ the output o can be produced by S for time instances $t' \in f$;
        add to λ_{R} the tuple $\langle q, \langle i, [t, t]\rangle, \langle o, f\rangle, r_{\mathsf{P}}\rangle$ if for each $t \in g$ the output o can be produced by P for time instances $t' \in f$;
      end if
    end for
    delete state q from the set Q_k;
  end for
  $k := k + 1$; $Q_k := Q_{k-1}$;
end while
if $\hat{q} \notin Q_k$ **then**
  convert the tuple $R = \langle R, I, O, \lambda_{\mathsf{R}}\rangle$ into a TFSM R by claiming state $\hat{q}$ as the initial state of the TFSM and augment R (if it is necessary) to an output-complete TFSM by adding transitions to deadlock states;
  return the largest initially connected submachine of TFSM R as the TFSM $\mathsf{R}_{(\mathsf{S},\mathsf{P})}$;
else
  return TFSMs S and P are not r-distinguishable.
end if

or r_{P}. Correspondingly, if state r_{S} (r_{P}) of $\mathsf{R}_{(\mathsf{S},\mathsf{P})}$ is reached then the TFSM under test is S (P).

Similar to [6], it can be shown that each trace of a TFSM $\mathsf{R}_{(\mathsf{S},\mathsf{P})}$ obtained in the above algorithm is of order $|S| \cdot |P|$ where S and P are the sets of states of TFSMs S and P, respectively and only one trace of $\mathsf{R}_{(\mathsf{S},\mathsf{P})}$ is used when performing the experiment. In this paper, as for other distinguishing experiments, the complexity of an adaptive experiment is measured using the height of the experiment, i.e., the length of a longest trace to a deadlock state in the (acyclic) TFSM $\mathsf{R}_{(\mathsf{S},\mathsf{P})}$. As TFSM $\mathsf{R}_{(\mathsf{S},\mathsf{P})}$ has at most $|S| \cdot |P|$ states, this length, and thus, the complexity of an adaptive experiment, is at most $|S| \cdot |P|$ and this upper bound is reachable as this upper bound is reachable for two untimed FSMs [22].

Example 1. Consider the running example and TFSMs S_1 and S_3 with the initial states 1 and 3, respectively. We add into R two deadlock states r_{S_1} and r_{S_3} with subscripts indicating the initial states of the machines. The intersection $\mathsf{Q} = \mathsf{S}_1 \cap \mathsf{S}_3$ is shown in Figure 2. The FSM abstraction $\mathsf{A}(\mathsf{Q})$ is constructed from Q by having the same states and splitting every transition of Q using the abstract inputs and outputs given above. A fragment of $\mathsf{A}(\mathsf{Q})$ for states $\langle 1,3\rangle$ and $\langle 3,2\rangle$ under the input i_1 of the intersection Q is shown in Figure 3. In particular, Figure 3 includes the transitions at states $\langle 1,3\rangle$ and $\langle 3,2\rangle$ under i_1 of Q (in Figure 2) and their corresponding transitions in $\mathsf{A}(\mathsf{Q})$ derived using the partitions Π_{i_1}, Π_{o_1} and Π_{o_2} given above. By applying Algorithm 1, initially, $k = 1$, the set $Q_1 = Q$ includes all

$\mathsf{A}(\mathsf{Q})$	$\langle 1,3\rangle$	$\langle 3,2\rangle$
i_1	$(t=0), \langle 1,3\rangle / \langle o_1, 2<t<3\rangle$; $(0<t<2), \langle 1,3\rangle / \langle o_1, 2<t<3\rangle$ $(t=2), \langle 1,3\rangle / \langle o_1, 2<t<3\rangle$; $(2<t<3), \langle 3,2\rangle / \langle o_1, t=2\rangle$ $(2<t<3), \langle 3,2\rangle / \langle o_1, 0<t<2\rangle$; $(t=3), \langle 3,2\rangle / \langle o_1, t=0\rangle$ $(t=3), \langle 3,2\rangle / \langle o_1, 0<t<2\rangle$; $(t>3), \langle 3,1\rangle / \langle o_1, t=0\rangle$ $(t>3), \langle 3,1\rangle / \langle o_1, 0<t<2\rangle$; $(t>3), \langle 3,1\rangle / \langle o_1, t=2\rangle$ $(t>3), \langle 3,1\rangle / \langle o_1, 2<t<3\rangle$; $(t>3), \langle 3,1\rangle / \langle o_1, t=3\rangle$ $(t>3), \langle 3,1\rangle / \langle o_1, 3<t<5\rangle$; $(2<t<3), \langle 2,4\rangle / \langle 0_2, t=0\rangle$ $(2<t<3), \langle 2,4\rangle / \langle 0_2, 0<t<5\rangle$; $(t=3), \langle 2,4\rangle / \langle o_2, t=0\rangle$ $(t=3), \langle 2,4\rangle / \langle 0_2, 0<t<5\rangle$	$(t=0), \langle 3,1\rangle / \langle o_1, 2<t<3\rangle$; $(t=0), \langle 3,1\rangle / \langle o_1, t=3\rangle$ $(t=0), \langle 3,1\rangle / \langle o_1, 3<t<5\rangle$; $(0<t<1), \langle 3,1\rangle / \langle o_1, 2<t<3\rangle$ $(0<t<1), \langle 3,1\rangle / \langle o_1, t=3\rangle$; $(0<t<1), \langle 3,1\rangle / \langle o_1, 3<t<5\rangle$ $(t=2), \langle 3,1\rangle / \langle o_1, 2<t<3\rangle$; $(t=2), \langle 3,1\rangle / \langle o_1, t=3\rangle$ $(t=2), \langle 3,1\rangle / \langle o_1, 3<t<5\rangle$; $(2<t<3), \langle 2,2\rangle / \langle o_1, t=0\rangle$ $(2<t<3), \langle 2,2\rangle / \langle o_1, 0<t<2\rangle$; $(t=3), \langle 2,2\rangle / \langle o_1, t=0\rangle$ $(t=3), \langle 2,2\rangle / \langle o_1, 0<t<2\rangle$; $(t>3), \langle 1,3\rangle / \langle o_1, t=0\rangle$ $(t>3), \langle 1,3\rangle / \langle o_1, 0<t<2\rangle$; $(t>3), \langle 1,3\rangle / \langle o_1, t=2\rangle$ $(t>3), \langle 1,3\rangle / \langle o_1, 2<t<3\rangle$; $(t>3), \langle 1,3\rangle / \langle o_1, t=3\rangle$ $(t>3), \langle 1,3\rangle / \langle o_1, 3<t<5\rangle$

Figure 3: Fragment of the abstract FSM $\mathsf{A}(\mathsf{Q})$

states of TFSM Q with the initial state $\langle 1,3\rangle$. States 3 and 2 of state $\langle 3,2\rangle$ in Q_1 are 1-r-distinguishable by abstract input $\langle i_2, 1\rangle$ and states 2 and 4 of state $\langle 2,4\rangle$ in Q_1 are 1-r-distinguishable by $\langle i_1, 2\rangle$. Thus, we add states $\langle 3,2\rangle$ and $\langle 2,4\rangle$ into the set R, that initially contains only deadlock states r_{S_1} and r_{S_3}, remove these states from Q_1, obtain Q_2 as $Q_1 \setminus \{\langle 3,2\rangle, \langle 2,4\rangle\}$, and add into (initially empty) λ_{R} the tuples

$$\langle\langle 3,2\rangle, \langle i_2, [1,1]\rangle, \langle o_1, [0,0]\rangle, r_{\mathsf{S}_1}\rangle,$$
$$\langle\langle 3,2\rangle, \langle i_2, [1,1]\rangle, \langle o_1, (0,2)\rangle, r_{\mathsf{S}_1}\rangle,$$
$$\langle\langle 3,2\rangle, \langle i_2, [1,1]\rangle, \langle o_1, [2,2]\rangle, r_{\mathsf{S}_1}\rangle,$$
$$\langle\langle 3,2\rangle, \langle i_2, [1,1]\rangle, \langle o_1, (2,3)\rangle, r_{\mathsf{S}_1}\rangle,$$
$$\langle\langle 3,2\rangle, \langle i_2, [1,1]\rangle, \langle o_1, [3,3]\rangle, r_{\mathsf{S}_1}\rangle,$$
$$\langle\langle 3,2\rangle, \langle i_2, [1,1]\rangle, \langle o_1, (3,5)\rangle, r_{\mathsf{S}_1}\rangle,$$

and add the tuples

$$\begin{array}{l}\langle\langle 2,4\rangle,\langle i_2,[2,2]\rangle,\langle o_1,[0,0]\rangle, r_{S_1}\rangle,\\ \langle\langle 2,4\rangle,\langle i_2,[2,2]\rangle,\langle o_1,(0,2)\rangle, r_{S_1}\rangle,\\ \langle\langle 2,4\rangle,\langle i_2,[2,2]\rangle,\langle o_1,[2,2]\rangle, r_{S_1}\rangle,\\ \langle\langle 2,4\rangle,\langle i_2,[2,2]\rangle,\langle o_1,(2,3)\rangle, r_{S_1}\rangle,\\ \langle\langle 2,4\rangle,\langle i_2,[2,2]\rangle,\langle o_1,[3,3]\rangle, r_{S_1}\rangle,\\ \langle\langle 2,4\rangle,\langle i_2,[2,2]\rangle,\langle o_1,(3,5)\rangle, r_{S_1}\rangle,\\ \langle\langle 2,4\rangle,\langle i_2,[2,2]\rangle,\langle o_2,[0,0]\rangle, r_{S_3}\rangle,\\ \langle\langle 2,4\rangle,\langle i_2,[2,2]\rangle,\langle o_2,(0,5)\rangle, r_{S_3}\rangle.\end{array}$$

Afterwards, in a second iteration of the loop, we observe that states 1 and 3 of state $\langle 1,3\rangle$ in Q_2 are 2-r-distinguishable. In fact, the abstract input $\langle i_1,3\rangle$ when applied at state $\langle 1,3\rangle$ of $A(Q)$ reaches only states $\langle 3,2\rangle$ and $\langle 2,4\rangle$ which are both 1-undefined. Thus, we add state $\langle 1,3\rangle$ into R, add into λ_R the tuples $\langle\langle 1,3\rangle,\langle i_1,[3,3]\rangle,\langle o_1,[0,0]\rangle,\langle 2,4\rangle\rangle$, $\langle\langle 1,3\rangle,\langle i_1,[3,3]\rangle,\langle o_1,(0,2)\rangle,\langle 3,2\rangle\rangle$, and add the tuples, $\langle\langle 1,3\rangle,\langle i_1,[3,3]\rangle,\langle o_2,[0,0]\rangle,\langle 2,4\rangle\rangle$, $\langle\langle 1,3\rangle,\langle i_1,[3,3]\rangle,\langle o_2,(0,5)\rangle,\langle 3,2\rangle\rangle$. Afterwards by deleting $\langle 1,3\rangle$, which is the initial state of $A(Q)$, from Q_2 we stop. Convert the tuple R into TFSM $R_{(S_1,S_3)}$ with initial state $\langle 1,3\rangle$ and obtain a partial TFSM as shown in Figure 4.

$R_{(S_1,S_3)}$	$\langle 1,3\rangle$	$\langle 3,2\rangle$	$\langle 2,4\rangle$	r_{S_1}	r_{S_3}
$\langle i_1,[3,3]\rangle$	$\langle 3,2\rangle/\langle o_1,[0,0]\rangle$ $\langle 3,2\rangle/\langle o_1,0<t<2\rangle$ $\langle 2,4\rangle/\langle o_2,[0,0]\rangle$ $\langle 2,4\rangle/\langle o_2,0<t<5\rangle$				
$\langle i_1,[2,2]\rangle$			$r_{S_1}/\langle o_1,[0,0]\rangle$; $r_{S_1}/\langle o_1,0<t<2\rangle$ $r_{S_1}/\langle o_1,[2,2]\rangle$; $r_{S_1}/\langle o_1,2<t<3\rangle$ $r_{S_1}/\langle o_1,[3,3]\rangle$; $r_{S_1}/\langle o_1,3<t<5\rangle$ $r_{S_3}/\langle o_2,[0,0]\rangle$; $r_{S_3}/\langle o_2,0<t<5\rangle$		
$\langle i_2,[1]\rangle$		$r_{S_1}/\langle o_1,[0,0]\rangle$; $r_{S_1}/\langle o_1,0<t<2\rangle$ $r_{S_1}/\langle o_1,[2,2]\rangle$; $r_{S_1}/\langle o_1,2<t<3\rangle$ $r_{S_1}/\langle o_1,[3,3]\rangle$; $r_{S_1}/\langle o_1,3<t<5\rangle$ $r_{S_3}/\langle o_2,[0,0]\rangle$; $r_{S_3}/\langle o_2,0<t<5\rangle$			

Figure 4: A part of the TFSM $R_{(S_1,S_3)}$

3.3 Deriving a Separating Sequence

In order to derive a separating sequence for two given TFSMs S and P, in the following, we adapt the algorithm given in [19] to deal with the abstract FSM $A(Q)$ of $Q = S \cap P$. Correspondingly, a separating sequence (if exists) will be derived for TFSMs S and P with output delays. If a separating sequence over abstract inputs $\langle i,g\rangle$ is derived from $A(Q)$, then the sequence is replaced by a corresponding timed sequence, over timed inputs $\langle i,t\rangle$, $t \in g$, that is a separating sequence for TFSMs S and P.

Here we define the following notion used in Algorithm 2. Given state s of an FSM $S = \langle S, I, O, \lambda_S, \hat{s}\rangle$, state s' is an i-successor of state s if there exists is a

Algorithm 2 Deriving a Separating Sequence of Two TFSMs

Input: Complete observable TFSMs $\mathsf{S} = \langle S, I, O, \lambda_{\mathsf{S}}, \hat{s}\rangle$ and $\mathsf{P} = \langle P, I, O, \lambda_{\mathsf{P}}, \hat{p}\rangle$

Output: A (shortest) separating sequence of TFSMs $\mathsf{S} = \langle S, I, O, \lambda_{\mathsf{S}}, \hat{s}\rangle$ and $\mathsf{P} = = \langle P, I, O, \lambda_{\mathsf{P}}, \hat{p}\rangle$ (if such a sequence exists)

1: derive the intersection $\mathsf{Q} = \mathsf{S} \cap \mathsf{P}$;
2: **if** Q is a complete TFSM **then**
3: the TFSMs $\mathsf{S} = \langle S, I, O, \lambda_{\mathsf{S}}, \hat{s}\rangle$ and $\mathsf{P} = \langle P, I, O, \lambda_{\mathsf{P}}, \hat{p}\rangle$ are non-separable;
4: **end** Algorithm 2;
5: **end if**
6: derive from $\mathsf{Q} = \mathsf{S} \cap \mathsf{P}$ (with input and output partitions Π_i and Π_o), the abstract FSM $\mathsf{A}(\mathsf{Q})$ with abstract inputs and outputs $\{\langle i, g\rangle : i \in I, g \in \Pi_i\}$ and $\{\langle o, f\rangle : o \in O, f \in \Pi_o\}$;
7: derive a truncated successor tree of the FSM $\mathsf{A}(\mathsf{Q})$. The root of this tree, which is at the 0^{th} level, is the initial state $\langle \hat{s}, \hat{p}\rangle$ of $\mathsf{A}(\mathsf{Q})$; the nodes of the tree are labeled with subsets of states of $\mathsf{A}(\mathsf{Q})$. Given already derived j tree levels, $j \geqslant 0$, a non-leaf (intermediate) node of the j^{th} level labeled with a subset C of states of $\mathsf{A}(\mathsf{Q})$ and a abstract input $\langle i, g\rangle$, there is an outgoing edge from this non-leaf node labeled with $\langle i, g\rangle$ to the node with the subset of the $\langle i, g\rangle$-successors of states of the subset C. A current node *Current*, at the k^{th} level, $k \geqslant 0$, labeled with the subset C of states, is claimed as a leaf node if one of the following conditions holds:
8: **Rule 1:** There exists an input $\langle i, g\rangle$ such that each state $\langle s, p\rangle$ of the set C has no $\langle i, g\rangle$-successors in $\mathsf{A}(\mathsf{Q})$;
9: **Rule 2:** There exists a node at the j^{th} level, $j < k$, labeled with a subset R of states with the property $R \subseteq C$;
10: **if** none of the paths of the truncated tree derived at Step 7 is terminated using **Rule 1 then**
11: the TFSMs $\mathsf{S} = \langle S, I, O, \lambda_{\mathsf{S}}, \hat{s}\rangle$ and $\mathsf{P} = \langle P, I, O, \lambda_{\mathsf{P}}, \hat{p}\rangle$ are non-separable;
12: **end** Algorithm 2;
13: **end if**
14: **if** there is a leaf node, *Leaf*, labeled with the subset C of states such that for some (abstract) input $\langle i, g\rangle$, each state of the set C has no $\langle i, g\rangle$-successors **then**
15: select such a path with minimal length, append an input sequence that labels the path with input $\langle i, g\rangle$ and transform the obtained input sequence replacing each abstract input of the sequence $\langle i, h\rangle$ by a timed input $\langle i, t\rangle$, $t \in h$;
16: the obtained timed input sequence is a shortest separating sequence of TFSMs S and P;
17: **end if**

transition $\langle s, i, o, s'\rangle$ in λ_{S}. Generally, for a nondeterministic FSM, the set of i-successors of state s can have several states. Given a set of states $M \subseteq S$ of the complete FSM S, and an input i, the set M' of states is an i-successor of the set M if M' is the union of the sets of i-successors over all states of the set M.

Similar to [19] it can be shown that Algorithm 2 returns a separating sequence α if and only if the TFSMs S and P are separable. The separating sequence α can be applied to a TFSM under experiment (S or P) and since the sets of output responses of TFSMs S and P do not intersect, after getting the output response to α the conclusion can be drawn which TFSM is under the experiment. In addition, it can be shown that the complexity (length of a separating sequence) is exponential w.r.t. to the number of states of TFSMs S and P as it happens for untimed FSMs [19]. The length of a separating sequence of two FSMs with n and m states is at most 2^{mn-1} [19] and this upper bound is reachable, and thus, it is reachable for TFSMs as well.

The above algorithm is based on deriving a successor tree using an (FSM) abstraction $A(Q)$ of the intersection $Q = S \cap P$. As $A(Q)$ can have more inputs than Q, we compare the above approach with another approach where a successor tree can be derived using Q instead [6]. In both approaches, in the worst case, each path p from the root node to a leaf node has to be traversed and a number o of elementary operations (**Rule 1** and **Rule 2**) have to be applied at each node of a path. Let l be the maximum length of a path, then the complexity of the algorithm equals the product $p \cdot l \cdot o$. The maximal length l is the same for the two approaches and l is of the order $O(2^{mn})$ for TFSMs S and P with m and n states, respectively [19]. Further, in both approaches, **Rule 1** and **Rule 2** of the above algorithm have to be checked at each node of the derived successor tree where a node is labeled with the set C of states of a corresponding TFSM Q or of the abstraction FSM $A(Q)$. Checking these rules using $Q = S \cap P$ is more complex since at each node for each input i and each subset Q_{kj} of states at the node we have to derive the set Π as the intersection of $\Pi\langle q, i\rangle$ over all states $q \in Q_{kj}$ while in the approach based on $A(Q)$, the intersection is calculated only once when deriving $A(Q)$. As the number of guards we need to intersect is proportional to the product of the finite upper bound of guards for input i and the number of states of the set Q_{kj}, in the approach based on $Q = S \cap P$, the number of calculations which have to be performed for deriving the intersection of guards at each node polynomially grows compared with the approach based on $A(Q)$. On the other hand, the number of inputs of $A(Q)$ can be larger than that of Q. If $\mathbf{B}$ is the maximum finite bound for a given input i over all states then for each i, the number of (abstract) inputs of $A(Q)$ can be $2 \cdot \mathbf{B}$ times bigger than that of Q, since in $A(Q)$ time domains for an i are derived based on the corresponding guards for all states of $A(Q)$. As the number p of paths of the successor tree exponentially depends on the number of inputs considered at each tree node, this implies that the complexity of the approach based on $A(Q)$ will exponentially grow compared to the approach based on Q, since p is of the order $O(|I|^l)$ where $|I|$ is the number of inputs of Q or $A(Q)$, respectively. This difference between the two approaches can be bypassed by considering for each input i only guards corresponding to a given state of Q when deriving the abstraction $A(Q)$, i.e., not taken into account guards under this input over other states of Q. In this case, it can well happen that $A(Q)$ is partially specified. The above algorithm can be adapted to partial FSM $A(Q)$; however, this is not done in this paper in order to simplify the presentation of the algorithms and to avoid presenting more complex FSM related definitions that consider defined and undefined input sequences at states. If partially specified FSM $A(Q)$ is used, the number p will be the same for both approaches. Generally, the approach based on the partial FSM abstraction of the intersection performs less computations than the approach based on the intersection Q instead. However, the best way to assess any abstraction method is thorough experimental evaluation with large size specifications and this could be the topic of another paper. It is worth mentioning that though the length of a separating sequence can reach length 2^{mn-1} (for TFSMs S and P with m and n states) [19]; nevertheless, experiments with various size FSM specifications show that this length usually does not exceed mn [18].

As $A(Q)$ can have more inputs than Q, here we also compare the approach given in this paper (Algorithm 1) based on using $A(Q)$ with another approach [6] based on using Q instead for deriving an adaptive distinguishing sequence (represented as a distinguishing machine). For both approaches, in the worst-case, the maximum length l of a path from the initial state of the constructed FSM $R_{(S,P)}$ to the deadlock state r_S or r_P is the same and is of the order $O(mn)$ for TFSMs S and P with m and n states, respectively [5]. In addition, as both approaches are based on deriving a submachine of a $A(Q)$ or of Q, the number of paths p included as transitions in the tuples of λ_R in both approaches is the same, and p is of the order $O(2^{mn})$ [22]. Moreover, in the approach that is based on the intersection Q, in the worst case, for a given input, we have to consider all possible time domains $\langle i, g \rangle$, $g \in \Pi$, over all states $q \in Q_k$. As the number of guards we need to intersect when deriving the set Π is proportional to the product of the finite upper bound of guards for input i and the number of states of the set Q_k, the number of calculations which have to be performed at each step almost coincide in both approaches. However, unlike the algorithm based on Q, the algorithm based on using $A(Q)$ performs less computations at each node as the intersection of guards for each input and each set Q_k of states will be performed only once when deriving $A(Q)$. To the best of our knowledge, no experiments were conducted for deriving adaptive distinguishing sequences and it would be interesting to assess the length of adaptive distinguishing sequences in practice and to evaluate the performance of the above approaches with respect to large size FSM specifications.

4 Conclusion

In this paper, a method for distinguishing two complete possibly nondeterministic TFSMs is presented based on an FSM abstraction of the intersection of the two TFSMs. The abstraction is derived by appropriate partitioning the input and output time domains. It is shown how a traditional preset FSM-based method can be used for deriving a separating sequence for the given TFSMs using the FSM abstraction. In addition, using the FSM abstraction, we present an algorithm for deriving an r-distinguishing TFSM that represents a simple adaptive distinguishing experiment for two given TFSMs. We compare the complexity of a proposed approach with that of another approach that is based directly on the intersection of two given TFSMs and show that in both approaches, similar to untimed FSMs, when distinguishing two TFSMs with m and n states, the length of a longest trace of a corresponding r-distinguishing machine is at most mn, while the length of a separating sequence is at most 2^{mn-1}, and these upper bounds are reachable [19,22].

As a future work, it would be interesting to investigate the possibility of adapting the presented work for distinguishing more than two machines as well as for a TFSM model with multiple clocks where the main challenge is the derivation of appropriate partitions of input and output time domains. In addition, it would be interesting to experiment and assess the performance of the proposed methods using large size specifications.

Acknowledgements

The authors would like to thank Dr. Zoltán Ésik and the anonymous reviewers for their helpful comments for improving the manuscript.

References

[1] Alur, Rajeev, Courcoubetis, Costas, and Yannakakis, Mihalis. Distinguishing tests for nondeterministic and probabilistic machines. In *Proceedings of the twenty-seventh annual ACM symposium on Theory of computing*, STOC '95, pages 363–372, New York, NY, USA, 1995. ACM.

[2] Bochmann, Gregor V. and Petrenko, Alexandre. Protocol testing: review of methods and relevance for software testing. In *Proceedings of the 1994 ACM SIGSOFT international symposium on Software testing and analysis*, ISSTA '94, pages 109–124, New York, NY, USA, 1994. ACM.

[3] Dorofeeva, Rita, El-Fakih, Khaled, Maag, Stephane, Cavalli, Ana R., and Yevtushenko, Nina. Fsm-based conformance testing methods: A survey annotated with experimental evaluation. *Inf. Softw. Technol.*, 52(12):1286–1297, December 2010.

[4] Gill, Arthur. Sate-identification experiments in finite automata. *Information and Control*, 4(2-3):132–154, 1961.

[5] Gromov, M. L., Evtushenko, N. V., and Kolomeets, A. V. On the synthesis of adaptive tests for nondeterministic finite state machines. *Program. Comput. Softw.*, 34(6):322–329, 2008.

[6] Gromov, Maxim, El-Fakih, Khaled, Shabaldina, Natalia, and Yevtushenko, Nina. Distinguing non-deterministic timed finite state machines. In *Proceedings of the Joint 11th IFIP WG 6.1 International Conference FMOODS '09 and*

29th IFIP WG 6.1 International Conference FORTE '09 on Formal Techniques for Distributed Systems, FMOODS '09/FORTE '09, pages 137–151, Berlin, Heidelberg, 2009. Springer-Verlag.

[7] Hierons, Rob M. Testing from a nondeterministic finite state machine using adaptive state counting. *IEEE Trans. Comput.*, 53(10):1330–1342, October 2004.

[8] Hierons, Robert M., Merayo, Mercedes G., and Núñez, Manuel. Testing from a stochastic timed system with a fault model. *J. Log. Algebr. Program.*, 78(2):98–115, 2009.

[9] Kohavi, Zvi. *Switching and Finite Automata Theory.* McGraw-Hill, 1978.

[10] Krichen, Moez and Tripakis, Stavros. State identification problems for timed automata. In *Proceedings of the 17th IFIP TC6/WG 6.1 international conference on Testing of Communicating Systems*, TestCom'05, pages 175–191, Berlin, Heidelberg, 2005. Springer-Verlag.

[11] Lee, David and Yannakakis, Mihalis. Testing finite-state machines: State identification and verification. *IEEE Trans. Comput.*, 43(3):306–320, March 1994.

[12] Lee, David and Yannakakis, Mihalis. Principles and methods of testing finite state machines-a survey. *Proceedings of the IEEE*, 84(8):1090–1123, 1996.

[13] Mathur, Aditya P. *Foundations of Software Testing.* Addison-Wesley Professional, 1st edition, 2008.

[14] Merayo, Mercedes G., Núñez, Manuel, and Rodríguez, Ismael. Extending efsms to specify and test timed systems with action durations and timeouts. In *Proceedings of the 26th IFIP WG 6.1 international conference on Formal Techniques for Networked and Distributed Systems*, FORTE'06, pages 372–387, Berlin, Heidelberg, 2006. Springer-Verlag.

[15] Merayo, Mercedes G., Núñez, Manuel, and Rodríguez, Ismael. Formal testing from timed finite state machines. *Computer Networks*, 52(2):432–460, 2008.

[16] Petrenko, Alexandre and Yevtushenko, Nina. Conformance tests as checking experiments for partial nondeterministic fsm. In *Proceedings of the 5th international conference on Formal Approaches to Software Testing*, FATES'05, pages 118–133, Berlin, Heidelberg, 2006. Springer-Verlag.

[17] Petrenko, Alexandre and Yevtushenko, Nina. Adaptive testing of deterministic implementations specified by nondeterministic fsms. In *Proceedings of the 23rd IFIP WG 6.1 international conference on Testing software and systems*, ICTSS'11, pages 162–178, Berlin, Heidelberg, 2011. Springer-Verlag.

[18] Shabaldina, Natalia, El-Fakih, Khaled, and Yevtushenko, Nina. Testing non-deterministic finite state machines with respect to the separability relation. In *Proceedings of the 19th IFIP TC6/WG6.1 international conference, and 7th international conference on Testing of Software and Communicating Systems*, TestCom'07/FATES'07, pages 305–318, Berlin, Heidelberg, 2007. Springer-Verlag.

[19] Spitsyna, Natalia, El-Fakih, Khaled, and Yevtushenko, Nina. Studying the separability relation between finite state machines. *Softw. Test. Verif. Reliab.*, 17(4):227–241, December 2007.

[20] Starke, Peter H. *Abstract Automata*. Elsevier, 1972.

[21] Tanenbaum, Andrew S. *Computer networks*. Prentice-Hall, 3 edition, 1996.

[22] Yevtushenko, Nina and Spitsyna, Natalia. On the upper of length of separating and *r*-distinguishing sequences for observable nondeterministic FSMs. In *Proceedings of Artificial intelligence systems and computer sciences*, pages 124–126, 2005. (in Russian).

[23] Yevtushenko, Nina, Vetrova, Maria, and Petrenko, Alexandre. *Analysis and synthesis of nondeterministic FSMs: operators and relations*. Tomsk State University publishing, 2006. (in Russian).

Received 9th May 2012