# TESTING TIMED FINITE STATE MACHINES WITH GUARANTEED FAULT COVERAGE

Khaled El-Fakih[1], Nina Yevtushenko[2*], Hacene Fouchal[3]

[1]American University of Sharjah, PO Box 26666, UAE
kelfakih@aus.edu
[2]Tomsk State University, 36 Lenin Str.. Tomsk, 634050, Russia
ninayevtushenko@yahoo.com
[3]Univ. Antilles Guyane, Guadeloupe, France
Hacene.Fouchal@univ-ag.fr

**Abstract:** A method is presented for deriving test suites with the guaranteed fault coverage for deterministic possibly partial Timed Finite State Machines (TFSMs). TFSMs have integer boundaries for time guards and the time reset operation at every transition; for TFSM implementations the upper bound on the number of states is known as well as the largest finite boundary and the smallest duration of time guards. We consider two fault models and present corresponding techniques for deriving complete test suites. In the first fault model inputs can be applied at integer time instances while in the second fault model time instances can be rational. The derivation method for integer time instances is extended to the case when the number of states of an implementation under test can be larger than the number of states of the given specification.

## 1. Introduction

Many conformance test derivation methods are based on a specification given in the form of a Finite State Machine (FSM), such as W [3], [15], partial W (Wp) [6], HIS [12], [13], [16] and the H [4] test derivation methods. For surveys see [2], [9]. In FSM-based testing, one usually assumes that the specification and an Implementation Under Test (IUT) can be modeled as FSMs. An IUT is faulty if it has a behavior different than the behavior of the given specification. Two types of implementation faults are usually considered, namely output faults and transfer faults. Each test derivation method mentioned above provides the following fault coverage guarantee under the assumption that the upper bound on the number of states of an IUT is known: If an FSM IUT with at most $m$ states and a given (reduced) specification FSM has $n$ states, $m \geq n$, a test suite can be derived by the method and the IUT will only pass this test suite if and only if it conforms to the specification, i.e. it does not contain any output nor transfer faults. In many cases, one assumes that $m = n$.

* The second author acknowledges a partial support by the FCP Russian Program, contract 02.514.12.4002

Many systems such as telecommunication systems, plant and traffic controllers and others are written using models with time constraints, and thus, a number of papers consider test derivation for timed automata and Timed Finite State Machines (TFSMs). Almost all proposed methods are based on deriving from a given timed automaton (or timed FSM) an untimed FSM and then applying FSM-based test derivation methods for the obtained FSM. For example, Springintveld et al. [14] proposed a rigorous strategy for deriving a complete test suite for a timed automaton. The authors show that under the assumption that the specification and an IUT have deterministic behavior and the upper bound on the number of time regions of an IUT is known a complete test suite can be derived using the well known W-method [3]. The main idea behind the approach is to divide time into very small grids such that to assure that each input is applied at some time instance of each time region of each IUT. The same grids are used for all states and inputs. The method proposed in [14] is not practical since it returns test suites with huge length; however, the method has theoretic significance as it demonstrates that there exists an opportunity to derive test suites with the guaranteed fault coverage for timed FSMs without explicit enumeration of all possible implementations. Many papers inherit the idea proposed in [14]; for example the work in [5] extends the method to non-deterministic behaviors. Recently, Merayo et al. [8], [10] proposed 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. Another timed FSM model is used in [7]. However, [10] and [7] do not consider test derivation, namely, the authors in [8], [10] establish a number of conformance relations and the authors in [7] propose methods for distinguishing timed non-deterministic FSMs. Test derivation for stochastic non-deterministic timed FSMs is considered in [8]. A method has been reported in [18] for generating timed test cases from the model of timed transition systems. For a more detailed review of the above papers and other relevant methods the reader may refer to [5], [8], [14]. We note that many test derivation methods are proposed for timed systems based on simulation relations and thus these methods are not considered in this paper.

In this paper, we consider the TFSM model from [7] and show how a complete test suite can be derived under various test assumptions. We use the same idea as in [14] about the known number of time regions; however, our TFSMs can be partial and thus, time instances when inputs are applied to IUT depend also on the current state of the specification. In other words, different grids are used for different states and inputs. In particular, we consider deterministic possibly partial timed FSMs (TFSMs) where time constraints are used to limit time elapsed at states and we also use one clock variable that is reset at every transition. We consider two fault models and propose corresponding test derivation methods with the guaranteed fault coverage (i.e. methods that derive tests that detect every faulty IUT w.r.t the assumed fault model) More precisely, in the first model, we consider TFSMs with integer boundaries and implementations with the known upper bound on the number of states, known largest finite boundary, and given smallest duration of time guards. In this case timed inputs are applied to an IUT at discrete (integer) time instances. In the second

fault model, input time instances can be rational (i.e. continuous). For each considered fault model we propose a complete test derivation technique for the case when the number $m$ of states of an IUT equals the number $n$ of states of the specification TFSM. The technique with integer time instances is adapted to the case when $m > n$. Our methods are based on the HIS method [12], [13], [16] which is an adaptation of the W method for partial, possibly non-reduced FSMs. In particular, we extend the HIS method by defining appropriate fault models and test derivation algorithms for TFSMs.

This paper is organized as follows. Section 2 includes relevant definitions and notations and Section 3 includes test derivation methods for the cases when $m = n$ and $m > n$ for systems with discrete time inputs and a test derivation method for case $m = n$ for systems with continuous time inputs. Section 4 concludes the paper.

## 2 Preliminaries

In this section, we introduce the notion of a timed Finite State Machine (TFSM) [7] and some other notions and notations used in the paper.

**Definition 1.** An FSM $\mathcal{S}$ is a 5-tuple $(S, I, O, \lambda_S, s_0)$, 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 transition 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 after 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 integers such that $min \leq max$. When $min = max$ we consider the 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 integer 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 $\Pi$ is the set of time guards 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, $S$ responds with output $o$ instantly when $i$ is applied. In this paper, we check only functional equivalence [10] between TFSMs and thus, we do not consider output delays. In other words, in this paper, the transition relation is a 5-tuple, $\lambda_S \subseteq S \times I \times O \times S \times \Pi$.

Given a TFSM $S = (S, I, O, \lambda_S, s_0)$, for every pair $(s, i) \in S \times I$, we use $\Pi_{(s, i)}$ to denote the collection of the guards $g_i$ over all transitions $(s, i, o, s', g_i) \in \lambda_S$. If there is no transition $(s, i, o, s', g_i) \in \lambda_S$ then, by definition, $\Pi_{(s, i)}$ is the empty set. The notion of $\Pi_{(s, i)}$ is very close to the notion of time regions [1]; however, these regions are different for different states and inputs. The latter allows to check transitions with the same input at different states at different time instances.

Given a transition $(s, i, o_1, s', \lceil min, max \rceil) \in \lambda_S$, we refer to $max - min$ as to the *duration* of the time guard of the transition. Moreover, the largest finite boundary, denoted $B_S$ or $B$, over all guards of all transitions is called the *largest boundary* of the TFSM.

The machine $S$ 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 $\lceil min_1, max_1 \rceil \cap \lceil min_2, max_2 \rceil = \varnothing$; otherwise, the machine $S$ is (time) *non-deterministic*.

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

The TFSM $S$ is *complete* if the underlying FSM is complete and for each pair $(s, i) \in S \times I$ of TFSM $S$, 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 $[0, \infty)$. In this paper, we consider only deterministic but possibly partial TFSMs.

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

**Definition 4.** Given a TFSM $S$, 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. A timed input sequence $\alpha = (i_1, t_1) \ldots (i_l, t_l)$ is *defined* for TFSM $S$ at state $s$ if the TFSM has a sequence of transitions $(s_j, i_j, o_j, s_{j+1}, g_j)$ such that $s_1 = s$ and for each $j = 1, \ldots, l$, it holds that $t_j \in g_j$. The set of all defined timed sequences at state $s$ is denoted $\Omega_S(s)$ while denoting $\Omega_S$ the set of defined timed input sequences at the initial state, for short. The corresponding output sequence $o_1 \ldots o_l$ is denoted as $out_S(s, \alpha)$. As usual, we say that the pair $(\alpha, out_S(s, \alpha))$ *takes* the machine $S$ from state $s$ to state $s_{l+1}$. A pair "timed_input_sequence_$\alpha$/output_sequence_ $out_S(s, \alpha)$" is a *timed I/O sequence* or a *timed trace* of $S$ at state $s$. For a deterministic TFSM, given state $s$ and a timed input sequence $\alpha \in \Omega_S(s)$, $s_\alpha$ is the state in the TFSM reached by the sequence $\alpha$. We also say that $\alpha$ *takes* the TFSM to state $s_\alpha$. Given a state $s$ of a deterministic TFSM and a timed input $(i, t)$ defined at $s$, the $(i, t)$ successor of state $s$ is the state reached by applying $(i, t)$ at state $s$.

By the above definition, given a defined timed input sequence $\alpha = (i_1, t_1) \ldots (i_l, t_l)$, we assume that the sequence $\alpha$ is applied to the FSM in the following way. The input $i_1$ is applied at the time instance $t_1$; for each $j$, $1 < j \leq l$, the input $i_j$ is applied at the

time instance $t_j$ while time starts advancing from 0 after the output has been produced to the input $i_{j-1}$.

Consider TFSM $S$ shown in Fig. 1 shown below with three states named 1 (initial state), 2, and 3, and defined over the input alphabet $\{i_1, i_2\}$ and over the output alphabet $\{o_1, o_2, o_3\}$. TFSM $S$ is partial and deterministic. The collection of guards $\Pi_{(1, i1)}$ equals $\{[0,5), [5, 10]\}$, $\Pi_{(1, i2)} = \{ [0, \infty)\}$, $\Pi_{(2, i1)} = \{[0, 5), [5, \infty)\}$, $\Pi_{(2, i2)} = \{ [0, 5], (5, \infty)\}$, $\Pi_{(3, i1)} = \{[0, \infty)\}$, and $\Pi_{(3, i2)} = \{[6, \infty)\}$. The largest finite boundary $B = 10$.

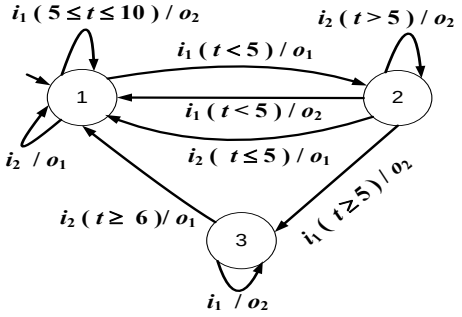


**Fig. 1.** TFSM $S$

The set of all timed traces of $S$ at state $s$ is denoted $TTr_S(s)$, also denoted $TTr_S$ for short if $s$ is the initial state of $S$. As usual, the TFSM $S$ 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 $sp$ 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 5.** Given TFSMs $S$ and $P$, the *intersection* $S \cap P$ is the largest connected submachine of the TFSM $(S \times P, I, O, \lambda_{S \cap P}, s_0p_0)$ where $(sp, i, o, s'p', \lceil min_1, max_1 \rceil) \in \lambda_{S \cap P}$ 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. $\lceil min_2, max_2 \rceil \cap \lceil min_3, max_3 \rceil \neq \varnothing$ and $\lceil min_1, max_1 \rceil = \lceil min_2, max_2 \rceil \cap \lceil min_3, max_3 \rceil$.

**Definition 6.** State $s$ of TFSM $S$ and $p$ of TFSM $P$ are *f-distinguishable* [10], denoted $s \underset{f}{\neq} p$, if there exists a timed input sequence $\alpha \in \Omega_S(s) \cap \Omega_P(p)$ such that $out_S(s, \alpha) \neq out_P(p, \alpha)$; the sequence $\alpha$ is said to *f-distinguish* states $s$ and $p$. If states $s$ and $p$ are not *f*-distinguishable then they are *f-compatible* (*functionally compatible*), denoted $s \underset{f}{\approx} p$. In the same way, *f-distinguishable* states can be introduced for states of a single TFSM. If each two different states of deterministic TFSM $S$ are *f*-distinguishable then $S$ is a *reduced* TFSM. TFSMs $S$ and $P$ are *f-compatible*, denoted $S \underset{f}{\approx} P$, if their initial states are *f*-compatible; otherwise, the machines are *f*-

*distinguishable*, denoted $S \underset{f}{\neq} P$. Timed input sequence $\alpha$ that *f-distinguishes* the initial states of $S$ and $P$ is an *f-distinguishing* sequence of $S$ and $P$. Given a set $W$ of defined timed input sequences at states $s$ and $p$, states $s$ and $p$ are *f-compatible* with respect to the set $W$, written $s \underset{f\,W}{\approx} p$, if $s$ and $p$ are not $f$-distinguishable for every sequence in the set $W$.

**Proposition 1.** Given two deterministic TFSMs $S$ and $P$, the TFSMs $S$ and $P$ are $f$-distinguishable iff there exists a state $(s, p)$ and an input $i$ such that the behavior of the intersection $S \cap P$ is not defined at state $(s, p)$ for a timed input $(i, t)$ while the behavior of $S$ at state $s$ and the behavior of $P$ at state $p$ are defined under $(i, t)$. In this case, each defined timed input sequence $\alpha.(i, t)$ where $\alpha$ takes the intersection $S \cap P$ to state $(s, p)$, $f$-distinguishes TFSMs $S$ and $P$.

**Corollary.** Given complete TFSMs $S$ and $P$, if the intersection $S \cap P$ is completely specified then the TFSMs $S$ and $P$ are not $f$-distinguishable.

A set of timed input sequences $V \subseteq \Omega_S$ is called *a state cover set* of TFSM $S$ if for each state $s_i$ of $S$, there is an input sequence $\alpha_i \in V$ that takes $S$ to state $s_i$.

Since the specification TFSM can be partial, the W-method and many of its derivatives cannot be used for deriving test suites with the guaranteed fault coverage. The reason is that, similar to untimed FSMs, a characterization set may not exist for a partial reduced TFSM. The HIS method can be applied when the specification FSM is partial and not reduced. In this paper, we adapt the HIS method for deriving a test suite with the guaranteed fault coverage; correspondingly, we define and use a separating family [17] of state identifiers, also known as a family of harmonized state identifiers [11], [13] for untimed FSMs.

**Definition 7.** Given state $s_j \in S$ of TFSM $S$, a set $W_j \subseteq \Omega_S(s_j)$ of timed input sequences is called a *state identifier* of state $s_j$ if for any other state $s_i \in S$ there exists $\gamma \in \Omega_S(s_i) \cap W_j$ that $f$-distinguishes $s_j$ and $s_i$, i.e. $out_S(s_i, \gamma) \neq out_S(s_j, \gamma)$. A *separating family* or a family of *harmonized identifiers* is a collection of state identifiers $W_j$, $s_j \in S$, which satisfy the condition: for any two different states $s_j$ and $s_i$, there exist $\beta \in W_j$ and $\chi \in W_i$ which have common prefix $\gamma$ such that $out_S(s_i, \gamma) \neq out_S(s_j, \gamma)$.

In this paper, we consider a Fault Model (FM) $<S, \underset{f}{\approx}, \mathfrak{I}>$, where $S$ is the specification TFSM that is deterministic and reduced, $\underset{f}{\approx}$ is the $f$-compatibility relation and $\mathfrak{I}$ is the fault domain, i.e., $\mathfrak{I}$ is a finite set of deterministic complete TFSMs with the same input alphabet as the specification TFSM $S$. A *test suite* (w.r.t. the FM) is a finite set of finite defined timed input sequences of the specification. A test suite is *complete* w.r.t. the FM if for each TFSM $P \in \mathfrak{I}$ s.t. $S \underset{f}{\neq} P$ the test suite has a sequence that $f$-distinguishes $P$ and $S$.

Given the FM $<S, \underset{f}{\approx}, \mathfrak{I}>$ where $\mathfrak{I}$ is a finite set of TFSMs, a complete test suite can be derived by explicit enumeration of TFSMs of the set $\mathfrak{I}$ using Proposition 1. However, the set $\mathfrak{I}$ can be huge and for this reason, we would like to develop a test

derivation method without the explicit enumeration of the machines in $\Im$. As usual, we impose some restrictions on the specification TFSM and on the fault domain.

## 3 Deriving Complete Test Suites for Timed FSMs

The main problem when deriving a test suite with the guaranteed fault coverage for the specification TFSM $S$ is that the number of defined timed inputs at each state of $S$ can be infinite. For this reason, for deriving a test suite with the guaranteed fault coverage it is not enough to limit, i.e. have the upper bound, the number of states of an IUT but also it is necessary to limit the number of time regions. Therefore, we limit the finite boundary $B_P$ of transition guards in an IUT. If we assume that each input can be applied only at integer time instances then it is enough to check at each state transitions under all the timed inputs $(i, t)$, $i \in I$, $t \in \{0, \ldots, B_P + 1\}$. However, the number of such inputs can also be huge and as usual, we further minimize the number of such timed inputs when the low bound on time interval of guards of an IUT is known.

### 3.1 Separating Family

A separating family for a given reduced TFSM $S$ can be derived in the same way as it is done for untimed FSMs: for every state $s_i$ a of $S$, consider every other state $s_j$, $i \neq j$, then derive and add into $W_i$ a timed input sequence $\gamma$ defined at $s_i$ and $s_j$, i.e. $\gamma \in \Omega_S(s_i)$ and $\gamma \in \Omega_S(s_j)$, that $f$-distinguishes the states $s_i$ and $s_j$. The family of all sets $W_i$ over all states $s_i$ of $S$ is a separating family of TFSM $S$. A timed input sequence that $f$-distinguishes two states can be derived using the intersection of the TFSM $S$ with the initial states $s_i$ and $s_j$, denoted $S/s_i$ and $S/s_j$ (Proposition 1).

As an example, consider the TFSM $S$ shown in Fig. 1 and states 1 and 2 of $S$. The initial state of the intersection of $S/1$ and $S/2$ is undefined under the timed input $(i_1, 2)$, thus, the sequence $(i_1, 2)$ $f$-distinguishes states 1 and 2 of $S$. Thus, we add $(i_1, 2)$ into $W_1$ and $W_2$. In this example, the sequence $(i_1, 2)$ also $f$-distinguishes states 1 and 3. Thus, we add $(i_1, 2)$ into $W_1$ and $W_3$. For states 2 and 3, we derive the intersection of $S/1$ and $S/3$ and find that the sequence $(i_1, 3).(i_1, 2)$ of timed inputs $f$-distinguishes the states. Thus, we add $(i_1, 3).(i_1, 2)$ into $W_2$ and $W_3$ and obtain the set $F = \{W_1, W_2, W_3\} = \{\{(i_1, 2), \}, \{(i_1, 2), (i_1, 3).(i_1, 2)\}, \{(i_1, 2), (i_1, 3).(i_1, 2)\}\}$ that is a separating family of TFSM $S$.

Here we note that two TFSMs can be $f$-compatible or $f$-distinguishable depending if a timed input can be applied only at integer time instances. For example, if an input can be applied only at integer time instances, then TFSMs cannot be distinguished with an input that is in the intersection $(a, a + 2)$ and $(a - 1, a + 1)$. In other words, in this case, two deterministic TFSMs $S$ and $P$ are $f$-distinguishable iff there exist a state $(s, p)$ and an input $i$ such that the behavior of $S$ at state $s$ and the behavior of $P$ at state $p$ are defined under $(i, t)$ and the behavior of the intersection $S \cap P$ is not defined at state $(s, p)$ for the timed input $(i, t)$ where $t$ is an integer. In fact in this case, each

defined timed input sequence $\alpha.(i, t)$ where $t$ is an integer, $\alpha$ takes the intersection $S \cap P$ to state $(s, p)$ and inputs of the sequence $\alpha$ are applied at integer time instances, $f$-distinguishes TFSMs $S$ and $P$. We further establish a statement (Proposition 2) that takes into account such distinguishability.

### 3.2 On Test Derivation for Integer Time Instances

Consider a fault model where the guard boundaries of the specification TFSM specification $S$ and of each implementation TFSM $P$ are integers, an implementation TFSM $P$ has at most $n$ states, where $n$ is the number of states of the specification TFSM $S$, the upper bound $B$ on the largest finite boundary of an implementation TFSM is known and only timed inputs $(i, t)$ where $t$ is a nonnegative integer can be applied to an IUT.

In this case, each TFSM can be represented as an untimed FSM that for each state $s$ has as defined inputs the finite set of timed inputs $(i, t)$, $i \in \Omega_S(s)$, $t \in \{0, \ldots, B + 1\}$ intersected with the union of all guards in $\Pi_{(s, i)}$. Then the classical HIS method and its derivatives can be applied to the obtained FSM for deriving a complete test suite w.r.t. to the assumed fault model. However, this test suite will be huge. Similar to [14] it can be shown that it is not enough to apply inputs at finite boundaries of time guards of the specification. Thus, more rigorous analysis is needed to assess the limitations on time guards of an IUT and propose related test suite derivation with the guaranteed fault coverage. These issues will be addressed in the following section.

When interested in TFSMs with up to $m$ states, we use $\Im_m(B, w)$ to denote the finite set of deterministic complete TFSMs with at most $m$ states, which have the same input alphabet as the specification TFSM $S$, the upper bound $B$ on the largest finite boundary and the minimal duration $w$ of a time guard of an implementation TFSM. When we want to emphasize that inputs can be applied only at integer time instances then we use $\Im^{in}_m(B, w)$ to denote such a set of IUTs.

### 3.3 Test Derivation for TFSMs with Integer Time Instances when $m = n$

In this subsection, we define a fault model, denoted FM_1, and then present an algorithm that returns a complete test suite w.r.t. this model. Consider the fault model FM_1 = $<S, \underset{f}{\approx}, \Im^{in}_n(B, w)>$, where:

1) The minimal (integer) duration $w$ of a finite time guard of an IUT is known.
2) An implementation TFSM $P \in \Im^{in}_n(B, w)$ is a deterministic complete FSM that has at most $n$ states, where $n$ is the number of states of the specification TFSM $S$;
3) The upper (integer) bound $B > 0$ on the largest finite boundary of an implementation TFSM is known.
4) Only timed inputs $(i, t)$ where $t$ is a nonnegative integer can be applied to an IUT.

**Proposition 2.** If only timed inputs $(i, t)$ where $t$ is a nonnegative integer can be applied to TFSM $\mathcal{S}$ then guards of the TFSM $\mathcal{S}$ can be described in the form $[a, b]$ or in the form $[a, \infty)$ where $a$ and $b$ are integers.

According to Proposition 2, for the example in Fig. 1, we can rewrite $\Pi_{(1,\ i1)} = \{[0,5), [5, 10]\}$ as $\{[0,4], [5, 10]\}$, $\Pi_{(2,\ i1)} = \{ [0, 5), [5, \infty)\}$ as $\{ [0, 4], [5, \infty)\}$, and $\Pi_{(2,\ i2)} = \{ [0, 5], (5, \infty)\}$ as $\{ [0, 5], [6, \infty)\}$. We also note that according to Proposition 2, if the specification TFSM has a guard $(a, a + 1)$ then after the transformation this guard is deleted from the transformed TFSM. If the specification TFSM has a guard $(a, a + k)$, $k > 1$, then this guard is transformed to $[a + 1, a + k - 1]$.

**Algorithm 1. Deriving a complete test suite w.r.t. the fault model $<\mathcal{S}, \underset{f}{\approx}, \mathfrak{I}^{in}_{n}(B,w)>$**

**Input**: Deterministic, possibly partial, reduced specification TFSM $\mathcal{S} = (S, I, O, \lambda_S, s_0)$ in the form of Proposition 2, $|S| = n$, a state cover set $V$ and a separating *family F of* $\mathcal{S}$, upper bound $B$ on the largest finite boundary of an IUT and the smallest duration $w$ of a time guard in a TFSM implementation of $\mathcal{S}$.

**Output**: A Complete test suite $TS$ with respect to FM_1 = $<\mathcal{S}, \underset{f}{\approx}, \mathfrak{I}^{in}_{n}(B, w)>$

**Step 1**.

Append every sequence $\alpha \in V$ with a corresponding state identifier. Denote $TS_1$ the obtained set. That is for each $\alpha \in V$ which takes $\mathcal{S}$ to state $s_\alpha$, $TS_1$ has sequences $\alpha.W_\alpha$, where $W_\alpha \in F$ is a state identifier of state $s_\alpha$;

If $w = 0$ or $w = 1$ then assign integer $u = 1$;

Else assign integer $u = w - 1$;

**Step 2**.

For every pair $(s, i) \in S \times I$ such that there exists a transition under $i$ at state $s$:

For each subset $g = [a, b] \in \Pi_{(s, i)}$ do:

Derive a set $T_g = g \cap \{a, a + 1{\cdot}u, \ldots, a + (k\text{-}1){\cdot}u, b\}$, $a + (k\text{-}1){\cdot}u < b$ and $a + k{\cdot}u \geq b$;

Endfor

For each subset $g = [a, \infty) \in \Pi_{(s, i)}$ do:

Derive a set $T_g = g \cap \{a, a + 1{\cdot}u, \ldots, a + (k\text{-}1){\cdot}u, B\}$, $a + (k\text{-}1){\cdot}u < B$ and $a + k{\cdot}u \geq B$;

Endfor

Denote $T_{(s, i)}$ the union of obtained sets of time instances $T_g$;

Endfor

**Step 3**. For every sequence $\alpha \in V$ that takes the specification FSM to state $s_\alpha$ and for each input $i$ such that there exists a transition under $i$ at state $s_\alpha$:

Append $\alpha$ with timed input $(i, t).W_k$ for every $t \in T_{(s_\alpha, i)}$ where $W_k$ is a state identifier in $F$ of the $(i, t)$-successor $s_k$ of state $s_\alpha$. Denote $TS_2$ the obtained set.

Endfor

**Step 4**. Return $TS$: = $TS_1 \cup TS_2$

□

**Example:** As an application example for Algorithm 1, consider TFSM $S$ shown in Fig. 1. The set $V = \{\varepsilon, (i_1, 2), (i_1, 2).(i_1, 6)\}$ is state cover set of $S$. We recall that the set of state identifiers of states 1, 2, and 3 are $W_1 = \{(i_1, 2)\}$, $W_2 = \{(i_1, 2), (i_1, 3).(i_1, 2)\}$, and $W_3 = \{(i_1, 2), (i_1, 3).(i_1, 2)\}$, respectively. The set $F = \{W_1, W_2, W_3\}$ is separating family of $S$. Assume that an IUT $P$ of $S$ has up to 3 states (as $n = 3$), length of each time interval $w$ of $S$ and $P$ is at least 4, highest bound $B$ of IUT equals and $u = w - 1 = 3$.

We apply Step-1 and obtain the sequences $TS_1 = \varepsilon.W_1 + (i_1, 2).W_2 + (i_1, 6).W_3 = \varepsilon.(i_1, 2) + (i_1, 2).(i_1, 2) + (i_1, 2).(i_1, 3)(i_1, 2) + (i_1, 6).(i_1, 2) + (i_1, 6).(i_1, 3)(i_1, 2)$. Then in Step-2, we consider the collection of guards of $S$, $\Pi_{(1, i1)} = \{[0, 4], [5, 10]\}$, $\Pi_{(1, i2)} = \{ [0, \infty)\}$, $\Pi_{(2, i1)} = \{ [0, 4], [5, \infty)\}$, $\Pi_{(2, i2)} = \{ [0, 5], [6, \infty)\}$, $\Pi_{(3, i1)} = \{[0, \infty)\}$, and $\Pi_{(3, i2)} = \{[6, \infty)\}$. For the pair $(1, i_1)$, we have $\Pi_{(1, i1)} = \{[0, 4], [5, 10]\}$ and correspondingly the set of time instances $T_{[0,4]} = \{0, 3, 4\}$ and $T_{[5,10]} = \{5, 8, 10\}$. Thus, $T_{(1, i1)} = \{0, 3, 4, 5, 8, 10\}$. For the pair $(1, i_2)$, the collection $\Pi_{(1, i2)} = \{ [0, \infty)\}$ and consequently $T_{(1, i2)} = \{0, 3, 6, 9, 12\}$. For the pair $(2, i_1)$, $\Pi_{(2, i1)} = \{[0, 4], [5, \infty)\}$, and consequently, $T_{(2, i1)} = \{0, 3, 4, 5, 8, 11\}$. For $(2, i_2)$, $\Pi_{(2, i2)}$ $\{[0, 5], [6, \infty)\}$ and $T_{(2, i2)} = \{0, 3, 5, 6, 9, 12\}$ and for $(3, i_1)$, $\Pi_{(3, i1)} = \{[0, \infty)\}$ and $T_{(3, i1)} = \{0, 3, 6, 9, 12\}$, and finally for the pair $(3, i_1)$, $\Pi_{(3, i2)} = \{[6, \infty)\}$ and $T_{(3, i2)} = \{6, 9, 12\}$.

Then at Step-3, consider $\alpha = \varepsilon \in V$ and $T(s_{\alpha, i1}) = T_{(1, i1)} = \{0, 3, 4, 5, 8, 10\}$. For instance $t = 0 \in T_{(1, i1)}$, the sequence $\varepsilon.(0, i_1)$ reaches state 2, thus form and add into $TS_2$ the sequences $\varepsilon.(i_1, 0).W_2$. Similarly consider every other instance $t \in T_{(1, i1)}$ and add into $TS_2$ the sequences $\varepsilon.(i_1, 3).W_2$; $\varepsilon.(i_1, 4).W_2$; $\varepsilon.(i_1, 5).W_1$; $\varepsilon.(i_1, 8).W_1$; $\varepsilon.(i_1, 10).W_1$. Then, consider $T_{(1, i2)} = \{0, 3, 6, 9, 12\}$ and add into $TS_2$ the sequences $\varepsilon.(i_2, 0).W_1$; $\varepsilon.(i_2, 3).W_1$; $\varepsilon.(i_2, 3).W_1$; $\varepsilon.(i_2, 9).W_1$; $\varepsilon.(i_2, 12).W_1$. For $\alpha = (i_1, 2) \in V$, $T(s_{\alpha, i1}) = T_{(2, i1)} = \{0, 3, 4, 5, 8, 11\}$. Consider every $t \in T_{(2, i1)}$ form and add into $TS_2$ the sequences $(i_1, 2).(i_1, 0).W_1$; $(i_1, 2).(i_1, 3).W_1$; $(i_1, 2).(i_1, 4).W_1$; $(i_1, 2).(i_1, 5).W_3$; $(i_1, 2).(i_1, 8).W_3$; $(i_1, 2).(i_1, 11).W_3$. Then consider $T(s_{\alpha, i2}) = T_{(2, i2)} = \{0, 3, 5, 6, 9 12\}$ and add into $TS_2$ the sequences $(i_1, 2).(i_2, 0).W_1$; $(i_1, 2).(i_2, 3).W_1$; $(i_1, 2).(i_2, 5).W_1$; $(i_1, 2).(i_2, 6).W_3$; $(i_1, 2).(i_2, 9).W_3$; $(i_1, 2).(i_2, 12).W_3$. Finally, for $\alpha = (i_1, 2)$ $(i_1, 6) \in V$, form and add into $TS_2$ corresponding sequences and return $TS_1 \cup TS_2$ .

**Proposition 3.** Given the fault model $<S, \underset{f}{\approx}, \mathfrak{I}^{in}_n(B, w)>$, Algorithm 1 returns a test suite $TS$ that is complete with respect to this fault model.

**Proof.** Let $P = (P, I, O, \lambda_P, p_0) \in \mathfrak{I}^{in}_n(B, w)$ be an implementation TFSM that has the expected output response to each input sequence of the set $TS_1$. In this case, TFSM has exactly $n$ states and moreover, we can establish the one-to-one correspondence $h$ between states of $S$ and $P$: $h(s) = p$ iff $s_j \underset{f}{\approx}_{W_j} p_j$.

Suppose now that output responses of $S$ and $P$ at states $s$ and $h(s)$ to some defined timed input $(i, t) \in \Omega_S(s)$ are different or the $(i, t)$-successor of state $h(s)$ does not equal $h(s')$ where $s'$ is the $(i, t)$-successor of state $h(s)$ then the implementation TFSM does not have the expected output response to each input sequence of the set $TS_2$.

Let $w = 0$ or $w = 1$. In this case, for every input and state, each boundary of $\mathcal{S}$ and $\mathcal{P}$ is in the form $[a, a]$, $[a, a+1]$, $[B, B]$, or $[B, B+1]$, respectively, each timed input $(i, t)$, $t$ is an integer and $t = a, \{a, a+1\}, B, \{B, B+1\}$, such that the specification behavior is defined at state $s$ for the timed input $(i, t)$ is applied to an IUT at state $h(s)$. Thus, if the IUT has the expected behavior for all sequences of the set $TS$ then the IUT is $f$-compatible with the specification.

Assume now that $w > 1$, i.e., $u = w - 1$. It is sufficient to show that for each non-empty intersection that contains at least one integer $t$ of two guards $[a, b] \in \Pi_{(s, i)}$ and $[a_1, b_1] \in \Pi_{(h(s), i)}$ it holds that the intersection of $[a_1, b_1]$ and the set $T_{(s, i)}$ is not empty.

Consider guards $[a, b] \in \Pi_{(s, i)}$, $b - a \geq w$, and $[a_1, b_1] \in \Pi_{(h(s), i)}$, $b_1 - a_1 \geq w$, s.t. the intersection $g$ of $[a_1, b_1]$ and the $T_{(s, i)}$ has at least one integer. A number of cases are possible.

1) $[a_1, b_1] \subset [a, b]$ then since $b_1 - a_1 \geq w$ and $u = w - 1$, there exists $l$ s.t. $a + lu \in [a_1, b_1]$. In this case $a + lu \in T_{(s, i)}$ and thus, $a \in [a_1, b_1] \cap T_{(s, i)}$.
2) $[a_1, b_1] \not\subset [a, b]$ and $a \in [a_1, b_1]$. In this case, $a \in T_{(s, i)}$ and thus, $a \in [a_1, b_1] \cap T_{(s, i)}$.
3) $[a_1, b_1] \not\subset [a, b]$ and $a \notin [a_1, b_1]$, i.e., $b \in [a_1, b_1]$ and thus, $b \in T_{(s, i)}$.

In the same way, we can prove that for each non-empty intersection of the guards $[a, b] \in \Pi_{(s, i)}$ and $[a_1, \infty) \in \Pi_{(h(s), i)}$, $[a, \infty) \in \Pi_{(s, i)}$ and $[a_1, b_1] \in \Pi_{(h(s), i)}$, $[a, \infty) \in \Pi_{(s, i)}$ and $[a_1, \infty) \in \Pi_{(h(s), i)}$ it holds that the intersection of $[a_1, b_1]$ (or correspondingly of $[a_1, \infty)$) and $T_{(s, i)}$ is not empty. □

### 3.4 Test Derivation for TFSMs with Integer Instances when $m > n$

As other FSM-based test derivation methods with the guaranteed fault coverage, the method presented in this paper can be adapted for the case when the number $m$ of states of an IUT can be larger than the number $n$ of states of the specification FSM, i.e. $m > n$. In this case, the fault domain of the fault model contains all TFSM implementations up to $m$ states, i.e. $\Im_m(B, w)$. In this paper we show how Algorithm 1 can be adapted for deriving a complete test suite for the fault model $<\mathcal{S}, \underset{f}{\approx}, \Im^{in}_m(B, w)>$. In this case we derive not only a state cover $V$ but the set $V^{m-n+1}$ in order to cover each timed transition of an IUT and then as usual append sequences of the set $V^{m-n+1}$ with corresponding state identifiers.

**Algorithm 2. Deriving a complete test suite w.r.t. the fault model $<\mathcal{S}, \underset{f}{\approx}, \Im^{in}_m(B,w)>$**

**Input**: Deterministic, possibly partial, reduced specification TFSM $\mathcal{S} = (S, I, O, \lambda_S, s_0)$, $|S| = n$, a state cover set $V$ and a separating *family F of* $\mathcal{S}$, upper bound $B$ on the largest finite boundary of an IUT, integer $m \geq n$, and the smallest duration $w$ of a time guard in a TFSM implementation of $\mathcal{S}$.

**Output**: A Complete test suite $TS$ with respect to FM_2 = $<\mathcal{S}, \underset{f}{\approx}, \Im^{in}_m(B, w)>$

**Step 1**. $TS$: $= \varnothing$;.
If $w = 0$ or $w = 1$ then assign $u = 1$;
Else assign $u = w - 1$;

**Step 2**.
For every pair $(s, i) \in S \times I$ such that there exists a transition under $i$ at state $s$:
For each subset $g = [a, b] \in \Pi_{(s, i)}$ do:
Derive a set $T_g = g \cap \{a, a + 1 \cdot u, \ldots, a + (k\text{-}1) \cdot u, b\}$, $a + (k\text{-}1) \cdot u < b$ and $a + k \cdot u \geq b$;
Endfor
For each subset $g = [a, \infty) \in \Pi_{(s, i)}$ do:
Derive a set $T_g = g \cap \{a, a + 1 \cdot u, \ldots, a + (k\text{-}1) \cdot u, B\}$, $a + (k\text{-}1) \cdot u < B$ and $a + k \cdot u \geq B$;
Endfor
Endfor

**Step 3**. Assign $l$: = 1 and $V^l$: = $V$
While $l \leq m - n + 1$
For every sequence $\alpha \in V^l$ that takes the specification FSM to state $s_\alpha$ and each timed input $(i, t)$ that is defined at state $s_\alpha$:
Include into $V^{l+1}$ a sequence $\alpha(i, t)$ for every $t \in T_{(s_\alpha, i)}$;
Endfor
Increment $l$ by 1;
Endwhile

**Step 4**.
For every $\alpha \in V^1 \cup V^2 \ldots \cup V^{m-n+1}$
Append $\alpha$ with a corresponding state identifier. That is for $\alpha$ where $\alpha$ takes $S$ to state $s_\alpha$, add to $TS$ the sequences $\alpha.W_\alpha$, where $W_\alpha \in F$ is a state identifier of state $s_\alpha$;
Endfor
Return $TS$.

□

Similar to the statement of Proposition 3 we can prove the following statement.

**Proposition 4.** Given the fault model $<S, \underset{f}{\approx}, \Im^{in}_m(B, w)>$ the above described algorithm returns a test suite $TS$ that is complete with respect to $<S, \underset{f}{\approx}, \Im^{in}_m(B, w)>$.

### 3.5 Test derivation for TFSM with Rational Time Instances

Here we use the same fault model FM_1 defined above, in addition, we assume that time instances $t$ of timed inputs $(i , t)$ can be applied to an IUT at rational rather than only at integer time instances. In this case, we cannot transform TFSMs according to Proposition 2 and as the following example shows, it is not enough to apply inputs at integer time instances. Suppose that the specification TFSM has a guard $(a, b)$, $b > a + 1$ for input $i$ while the implementation TFSM has a guard $(a - 1,$

$a$ + 1) for this input. The intersection of these guards is ($a$ - 1, $a$ + 1) and in order to check the behavior of the implementation TFSM at a defined time instance we should apply a timed input ($i$, $t$), $t \in (a, a + 1)$, i.e. $t$ is a rational. Correspondingly such rational time instances have to be considered in Step 2 of Algorithm 1.

**Algorithm 3. Deriving a complete test suite w.r.t. FM_1 when time instances are rational**

**Input**: Deterministic complete reduced specification TFSM $\mathcal{S} = (S, I, O, \lambda_S, s_0)$, a state cover set $V$ and a separating *family F of* $\mathcal{S}$, upper (integer) bound $B$ on the largest boundary of an implementation under test, and the minimal (integer) duration of $w$ of a time guard in a TFSM implementation of $\mathcal{S}$.

**Output**: A Complete test suite $TS$ with respect to $<\mathcal{S}, \underset{f}{\approx}, \Im_n(B, w)>$

**Step 1**. Append every sequence $\alpha \in V$ with a corresponding state identifier. Denote $TS_1$ the obtained set. That is for each $\alpha \in V$, let $s_\alpha$ be the state reached by $\alpha$, $TS_1$ has sequences $\alpha.W_\alpha$, $W_\alpha \in F$;
If $w = 0$ or $w = 1$ assign $u := 1$;
Else $u := w$-1;
Select $\Delta$, $0 < \Delta < 1$;

**Step 2**.
For every pair $(s, i) \in S \times I$ such that there exists a transition under $i$ at state $s$:
For each subset $g = \lceil a, b \rceil \in \Pi_{(s, i)}$ do:
Derive a set $T_g = g \cap \{a, a + \Delta, a + 1 \cdot u, a + 1 \cdot u + \Delta, \ldots, a + (k\text{-}1) \cdot u, a + (k\text{-}1) \cdot u + \Delta, b, b - \Delta\}$, $a + (k\text{-}1) \cdot u < b$ and $a + k \cdot u \geq b$;
Endfor
For each subset $g = \lceil a, \infty) \in \Pi_{(s, i)}$ do:
Derive a set $T_g = g \cap \{ a, a + \Delta, a + 1 \cdot u, a + 1 \cdot u + \Delta, \ldots, a + (k\text{-}1) \cdot u, a + (k\text{-}1) \cdot u + \Delta, B, B + \Delta\}$, $a + (k\text{-}1) \cdot u < B$ and $a + k \cdot u \geq B$;
Endfor
Denote $T_{(s, i)}$ the union of obtained sets of time instances $T_g$;
Endfor

**Step 3**. For every sequence $\alpha \in V$ that takes the specification FSM to state $s_\alpha$ and for each input $i$ such that there exists a transition under $i$ at state $s_\alpha$:
Append $\alpha$ with timed input $(i, t).W_k$ for every $t \in T_{(s_\alpha, i)}$ where $W_k$ is a state identifier in $F$ of the $(i, t)$-successor $s_k$ of state $s_\alpha$. Denote $TS_2$ the obtained set.
Endfor

**Step 4**. Return $TS := TS_1 \cup TS_2$

□

**Proposition 4.** Given the fault model $<\mathcal{S}, \underset{f}{\approx}, \Im_n(B, w) >$, Algorithm 3 returns a test suite $TS$ that is complete with respect to this fault model.

**Proof.** Let FM = $<\boldsymbol{S}, \underset{f}{\approx}, \Im_n(B, w)>$ and $\boldsymbol{P} = (P, I, O, \lambda_P, p_0) \in \Im_n(B, w)$ be an implementation TFSM that has the expected output response to each input sequence of the set $TS_1$. In this case, TFSM has exactly $n$ states and moreover, we can establish the one-to-one correspondence $h$ between states of $\boldsymbol{S}$ and $\boldsymbol{P}$: $h(s) = p$ iff $s_j \underset{f\ W_j}{\approx} p_j$.

Suppose now that output responses of $\boldsymbol{S}$ and $\boldsymbol{P}$ at states $s$ and $h(s)$ to some defined timed input $(i, t) \in \Omega_S(s)$ are different or the $(i, t)$-successor of state $h(s)$ is not equal $h(s')$ where $s'$ is the $(i, t)$-successor of state $h(s)$ then the implementation TFSM does not have the expected output response to each input sequence of the set $TS_2$. Similar to the proof of Proposition 3, it is sufficient to show that for each non-empty intersection that contains at least one integer $t$, of two guards $\lceil a, b \rceil \in \Pi_{(s, i)}$ and $\lceil a_1, b_1 \rceil \in \Pi_{(h(s), i)}$ it holds that the intersection of $\lceil a_1, b_1 \rceil$ and $T_{(s, i)}$ is not empty.

Let $w = 0$. Consider a guard $\lceil a_1, b_1 \rceil$ of an IUT s.t. $g \cap \lceil a_1, b_1 \rceil \neq \varnothing$. If the guard $\lceil a_1, b_1 \rceil$ has at least one integer then $\{0, \Delta, 1, 1 + \Delta, \ldots, B, B + \Delta\} \cap \lceil a_1, b_1 \rceil \neq \varnothing$. Here we note that since $\lceil a_1, b_1 \rceil$ can be $[a_1]$ for each $a_1 \in \{0, 1, \ldots, B\}$, we have to include all these integers into the set intersected with $\lceil a_1, b_1 \rceil$. If $\lceil a_1, b_1 \rceil = (a_1, b_1)$ and $b_1 - a_1 = 1$ then there are no integers inside $(a_1, b_1)$. In this case, we have to apply an input at the time instance $(a_1 + \Delta)$, $a_1 \in \{0, 1, \ldots, B\}$.

Let $w = 1$. Consider a guard $\lceil a_1, b_1 \rceil$ of an IUT s.t. $g \cap \lceil a_1, b_1 \rceil \neq \varnothing$. If the guard $\lceil a_1, b_1 \rceil$ has at least one integer $t \leq B$ then since $b_1 - a_1 \geq 1$, it holds that $\{0, \Delta, 1, 1 + \Delta, \ldots, B, B + \Delta\} \cap \lceil a_1, b_1 \rceil \neq \varnothing$.

Let $w > 1$ and $g \in \Pi_{(s, i)}$, i.e., $u = w - 1$. Consider a guard $\lceil a_1, b_1 \rceil$ of an IUT s.t. $g \cap \lceil a_1, b_1 \rceil \neq \varnothing$.

Consider guards $\lceil a, b \rceil \in \Pi_{(s, i)}$, $b - a \geq w$, and $\lceil a_1, b_1 \rceil \in \Pi_{(h(s), i)}$, $b_1 - a_1 \geq w$. Similar to the proof of Statement 3, a number of cases are possible.

1) $\lceil a_1, b_1 \rceil \subset \lceil a, b \rceil$ then since $b_1 - a_1 \geq w$ and $u = w - 1 > 0$, there exists $t \in [a_1, b_1]$ s.t. $t = a + lu$. If $a_1 = a + lu$ and $\lceil a_1, b_1 \rceil = (a_1, b_1 \rceil$ then $a_1 + lu + \Delta \in T_{(s, i)}$.
2) $\lceil a_1, b_1 \rceil \not\subset \lceil a, b \rceil$ and $a \in \lceil a_1, b_1 \rceil$. If $\lceil a, b \rceil = [a, b \rceil$ then $a \in T_{(s, i)}$ and thus, $a \in \lceil a_1, b_1 \rceil \cap T_{(s, i)}$. If $\lceil a, b \rceil = (a, b \rceil$ then $(a + \Delta) \in T_{(s, i)}$.
3) $\lceil a_1, b_1 \rceil \not\subset [a, b \rceil$ and $a \notin \lceil a_1, b_1 \rceil$, then $b \in \lceil a_1, b_1 \rceil$. If $\lceil a, b \rceil = \lceil a, b]$ then $b \in T_{(s, i)}$ and thus, $b \in \lceil a_1, b_1 \rceil \cap T_{(s, i)}$. If $\lceil a, b \rceil = \lceil a, b)$ then $(b - \Delta) \in T_{(s, i)}$.

In the same way, we prove that for each non-empty intersection of the guards $\lceil a, b \rceil \in \Pi_{(s, i)}$ and $\lceil a_1, \infty) \in \Pi_{(h(s), i)}$, $\lceil a, \infty) \in \Pi_{(s, i)}$ and $\lceil a_1, b_1 \rceil \in \Pi_{(h(s), i)}$, $\lceil a, \infty) \in \Pi_{(s, i)}$ and $\lceil a_1, \infty) \in \Pi_{(h(s), i)}$ it holds that the intersection of $\lceil a_1, b_1 \rceil$ (or correspondingly of $\lceil a_1, \infty)$) and $T_{(s, i)}$ is not empty.

□

## 4. Conclusion and Future Work

Two fault models and corresponding test derivation methods are presented for deriving tests with the guaranteed fault coverage from a deterministic possibly partial specification Timed Finite State Machine (TFSM). In the first model (timed) inputs

can be applied at integer (or discrete) time instances and in the second at rational (or continuous) time instances. The test derivation method with integer time instances is extended to the case when an implementation under test can have more states that the given specification.

The considered TFSM model assumes a reset operation is employed at every transition. We think that the methods presented in the paper can be adapted to consider timed FSMs where there is no reset at every transition. Another possible extension of the proposed work is to adapt test derivation with the guaranteed fault coverage for non-deterministic TFSMs.

## References


1. Alur, R, and Dill. D. L.: A Theory of Timed automata. Theoretical Computer Science, 126(2),183--235 (1994)
2. 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)
3. T. S. Chow.: Test Design Modeled by Finite-state Machines. IEEE TSE, 4(3), pp. 178-187 (1978)
4. Dorofeeva, R., El-Fakih, K., Yevtushenko, N.: An Improved Conformance Testing Method. In Proc. of the IFIP 25th International Conference on Formal Techniques for Networked and Distributed Systems, LNCS 3731, pp. 204-218 (2005)
5. En-Nouaary, A., Dssouli, R., Khendek, F.: Timed Wp-Method: Testing Real-Time Systems, IEEE TSE 28(11), 1023--1038 (2002)
6. Fujiwara, S., Bochmann, G. v., Khendek, F., Amalou, M., Ghedamsi A.: Test Selection Based on Finite State Models. IEEE Trans. SE, 17(6), pp. 591-603 (1991)
7. Gromov, M., El-Fakih, K., Shabaldina, N., Yevtushenko, N.: Distinguishing Non-deterministic Timed Finite State Machines, 11th Formal Methods for Open Object-Based Distributed Systems and 29th Formal Techniques for Networked and Distributed Systems, FMOODS/FORTE, LNCS 5522, pp. 137–151 (2009)
8. [HMN09] 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)
9. Lee, D., Yannakakis, M.: Principles and Methods of Testing Finite State Machines-A Survey. In Proc. of the IEEE, 84(8), 1090--1123 (1996)
10. Merayo M. G., Nunez, M., Rodriguez I.: Formal Testing from Timed Finite State Machines. Computer Networks, 52(2), 432--460 (2008)
11. Petrenko, A.: Checking Experiments with Protocol Machines. Proc. 4th Int. Workshop on Protocol Test Systems (IWPTS), pp. 83-94 (1991)
12. Petrenko, A., Yevtushenko, N.: Testing from Partial Deterministic FSM Specifications. IEEE Trans. Computers 54(9): 1154-1165 (2005)
13. Petrenko, A., Yevtushenko, N., Lebedev, A., Das, A.: Nondeterministic State Machines in Protocol Conformance Testing. Proc. of the IFIP 6th IWPTS, France, pp. 363-378 (1993)
14. Springintveld, J., Vaandrager, F., D'Argenio, P.: Testing Timed Automata. Theoretical Computer Science, 254(1-2), 225–257 (2001)
15. Vasilevskii, M. P.: Failure Diagnosis of Automata. translated from Kibernetika, 4, pp. 98-108 (1973)
16. Yevtushenko, N., Petrenko, A.: Test derivation method for an arbitrary deterministic automaton. Automatic Control and Computer Sciences, Allerton Press Inc., USA, 5 (1990)

17. Yannakakis, M., Lee, D.: Testing Finite State Machines: Fault Detection. Journal of Computer and System Sciences, 50, pp. 209-227 (1995)
18. Cardell-Oliver R., Glover, T.: A Practical and Complete Algorithm for Testing Real-Time Systems. Formal Techniques for Real-Time Fault Tolerant Systems (1998)