

Deriving Test Suites for Timed Finite State Machines

M.Gromov, D.Popov, N.Yevtushenko

Tomsk State University
Faculty of Radiophysics

Department of Computer Science for Discrete Systems

E-mail: {gromov, dimadef}@sibmail.com, ninayevtushenko@yahoo.com

Abstract

This paper is devoted to the derivation of distinguishing sequences for timed Finite State Machines (FSM). Such distinguishing sequences are used when deriving a test suite for a timed FSM with the guaranteed fault coverage.

1 Introduction

The problem of synthesis and analysis of systems with time is a challenging problem [1], as timed constraints constitute the necessary part of real-time systems which in turn are applied almost everywhere. One of the most important time issues the deadline, when the system has to produce a response to an applied input. Another important time issue is a time-out when after some time units the system changes its state if no external input is applied by the environment. The simplest example is a mobile phone that dims light after appropriate time units. This issue is also important almost for all real-time systems.

Lately, a number of formal methods were developed for test derivation for timed automata (see, for example, [7, 4]). In order to derive a test suite with the guaranteed fault coverage, as usual, we need a fault model. The fault model [6] is a triple that has the specification, the conformance relation and the fault domain, that is, the set of all possibly faulty implementations. When deriving test suites for timed systems, in this paper, we assume that the behaviour of the specification and an implementation systems are described by Timed Finite State Machines (TFSM) [4, 5] where actions are divided into inputs and outputs. Time constrains limit the time at which an output has to be produced after an input has been applied. After this, the time variable is reset to zero. Moreover, a state can have a time invariant called time-out. If the time-out ex-

pires and no input is applied the system should change its state according to the specification. In [4, 5], the authors introduce several of conformance relations for such fault model. However, in order to derive a test suite with the guaranteed fault coverage with respect to any of conformance relations we lack methods for deriving distinguishing sequences.

In this paper, we consider timed systems with the logic behaviour that can be described by an FSM [4, 5]. A special discrete clock variable is used in order to represent a timed behaviour. We first consider the functional conformance when two timed FSMs are functionally equivalent, that is, for each timed input sequence they produce the same sequence of output actions. We then propose a method for deriving a distinguishing sequence for two Timed FSMs which are not functionally equivalent. Correspondingly, when the fault domain is finite, a complete test suite with respect to the functional equivalence relation can be derived by the explicit enumeration of all implementation Timed FSMs. A distinguishing sequence is derived based on the special distinguishing automaton. This automaton has designated state `fail`. A timed sequence is a distinguishing sequence if and only if it takes the distinguishing automaton to the state `fail`. If the distinguishing automaton has no state `fail`, then an implementation under test conforms to its specification. We also discuss how a distinguishing sequence, and correspondingly, a complete test suite, can be derived when an implementation system is allowed to be faster (or slower) than the specification system.

The rest of the paper is structured as follows. Section 2 has the preliminaries. Section 3 describes the proposed distinguishing automaton and its properties which are useful for test derivation. Finally, Section 4 concludes the paper.

2 Preliminaries

In this section, we follow papers [4, 5] when defining a Timed Finite State Machine (TFSM).

Definition 1 A *timed-FSM* is γ -tuple $S = \langle S, I, O, s_0, \lambda_S, \sigma_S, \Delta_S \rangle$ where S is a finite non-empty set of states with the initial state s_0 , I and O are finite disjoint input and output alphabets, $\lambda_S \subseteq S \times I \times O \times S$ is transition relation, $\sigma_S : \lambda_S \rightarrow \mathbb{N}$ is a speed function, and $\Delta_S : H \subseteq S \rightarrow S \times \mathbb{N}$ is a delay function.

If $\langle s, i, o, s' \rangle \in \lambda_S$ and $\sigma_S(s, i, o, s') = t$, denoted as $s \xrightarrow{i/o(t)} s'$, we say, that TFMS S , being in state s , accepts the input i and within t time units produces the output o , moves to the state s' and resets the clock (that is in the state s' time units are counted from 0).

If for a state s the function Δ_S is defined and $\Delta_S(s) = \langle s', t \rangle$, denoted as $s \xrightarrow{t} s'$, we say that if no input is applied to the TFMS in state s within t time units then TFMS moves to the state s' and resets the clock (that is in the state s' time units are counted from 0).

As usual, when considering initialised machines, we assume that there exists a special reset that takes the TFMS from each state to the initial state and the clock is reset to 0. Correspondingly, we assume that the TFMS always starts from the initial state with the clock being set to 0. The next input can be applied to the TFMS after the TFMS has already produced an output to the previous input. We remind that after each input or output action the clock is reset to 0.

Definition 2 A TFMS $\langle S, I, O, s_0, \lambda_S, \sigma_S, \Delta_S \rangle$ is called

- deterministic, if for each pair $\langle s, i \rangle \in S \times I$ there exists at most one pair $\langle o, s' \rangle \in O \times S$ such that $\langle s, i, o, s' \rangle \in \lambda_S$;
- input-enabled, if for each pair $\langle s, i \rangle \in S \times I$ there exists at least one pair $\langle o, s' \rangle \in O \times S$ such that $\langle s, i, o, s' \rangle \in \lambda_S$.

In this paper, we assume that each input can be applied at each time instance after the TFMS has produced an output to the previous input and the TFMS produces a unique output to this input, that is we consider only deterministic and input-enabled TFMSs.

Fig. 1 gives an example of TFMS A with the initial state I, which is deterministic and input-enabled, assuming that $I = \{i_1, i_2\}$. If an input i_1 is applied at time $t = 0, 1, 2$ then the system produces the output o_1 within four time units after the input i_1 has been

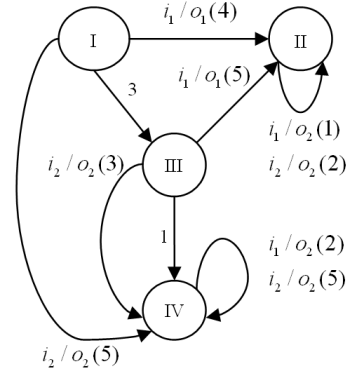


Figure 1. TFMS A

applied and enters state II. If the input i_1 is applied at $t = 3$ then the system reaches state II and produces the output o_1 within five time units after the input i_1 has been applied. However, if the input i_1 is applied at $t = 4, 5, \dots$ then the system moves to the state IV, produces o_2 within two time units and will stay in the state IV until a new input is applied.

Definition 3 A timed input for a TFMS $S = \langle S, I, O, s_0, \lambda_S, \sigma_S, \Delta_S \rangle$ is a pair $\langle i, t \rangle \in I \times \mathbb{Z}_+$, and a timed output is a pair $\langle o, t \rangle \in O \times \mathbb{N}$, where $\mathbb{Z}_+ \stackrel{\text{def}}{=} \mathbb{N} \cup \{0\}$ is the set of all non-negative integers.

A timed input can be applied to a system if an output to the previous input has been produced by the system. Let TFMS be in a state s . To calculate a set $\mathbf{out}_T(s, \langle i, t \rangle)$ of possible timed outputs of the system $\langle S, I, O, s_0, \lambda_S, \sigma_S, \Delta_S \rangle$ in state $s \in S$ for given timed input $\langle i, t \rangle$ we need to determine state s' such that

$$\sum_{j=1}^k (\Delta_S(s_j)) \downarrow_{\mathbb{Z}_+} \leq t < \sum_{j=1}^{k+1} (\Delta_S(s_j)) \downarrow_{\mathbb{Z}_+},$$

where $s_1 = s$, $s_k = s'$, $s_{j+1} = (\Delta_S(s_j)) \downarrow_S$, $j = \overline{1, k}$. Here we assume that if the function Δ_S is not defined for some state s_j , then $\Delta_S(s_j) = \langle s_j, \infty \rangle$, that is the system can stay in the state s_j infinitely long and moreover, given a set of pairs $D = A \times B$, we denote by $D \downarrow_A$ the set $\{a \in A \mid \exists b \in B : \langle a, b \rangle \in D\}$. Informally, we determine such a state s' which is reached by the TFMS within t time units starting from the state s . Note that due to definition of TFMS, there can be only one such a state. We denote the set of timed outputs of the system in the state s for the timed input $\langle i, t \rangle$ as $\mathbf{out}_T(s, \langle i, t \rangle) = \{\langle o, t' \rangle \in O \times \mathbb{N} \mid \exists s'' \in S : s' \xrightarrow{i/o(t')} s''\}$ and the set of out-

puts as $\mathbf{out}(s, \langle i, t \rangle) = (\mathbf{out}_T(s, \langle i, t \rangle)) \downarrow_O$. The notations $\mathbf{out}_T(s, \langle i, t \rangle)$ and $\mathbf{out}(s, \langle i, t \rangle)$ are naturally extended on timed input sequences $\alpha \in (I \times \mathbb{Z}_+)^*$. Let $\beta \in \mathbf{out}_T(s, \langle i, t \rangle)$ and $\gamma \in \mathbf{out}(s, \langle i, t \rangle)$. The set of all timed sequences α/β is called the set of *t-traces* (timed traces) of the system in the state s and denoted as $T\text{-traces}_S(s)$, while the set of sequences α/γ is the set of *f-traces* (functional traces) of the system in the state s and denoted as $F\text{-traces}_S(s)$. If s is the initial state, then we use notations $T\text{-traces}_S(s_0) = T\text{-traces}_S$ and $F\text{-traces}_S(s_0) = F\text{-traces}_S$.

Consider the TFMSM A from Fig. 1. Let $\langle i_1, 5 \rangle$ be a timed input. We obtain $\mathbf{out}_T(I, \langle i_1, 5 \rangle) = \{\langle o_2, 2 \rangle\}$ and $\mathbf{out}(I, \langle i_1, 5 \rangle) = \{o_2\}$.

3 Deriving a Complete Test Suite

In this paper, we consider a test suite as a finite set of finite timed input sequences. In order to guarantee that a derived test suite has an appropriate fault coverage we need the notion of a fault model.

3.1 Fault Model And a Complete Test Suite

As usual [6], we consider the fault model as a triple $\langle A, \approx, \Phi_A \rangle$ where A is an input-enabled deterministic TFMSM, \approx is a conformance relation while Φ_A is a (finite) set of TFMSMs which describe the behaviour of all, possibly faulty, implementation systems. Each implementation TFMSM has the same input and output alphabets, as the specification TFMSM. As discussed in [4], a number of conformance relations can be introduced for TFMSMs. In this paper, we first consider the so-called f-equivalence relation.

Let $A = \langle A, I, O, a_0, \lambda_A, \sigma_A, \Delta_A \rangle$ and $B = \langle B, I, O, b_0, \lambda_B, \sigma_B, \Delta_B \rangle$ be two input-enabled deterministic TFMSMs. TFMSMs A and B are *f-equivalent* if $F\text{-traces}_A = F\text{-traces}_B$, that is, if the output sequences of A and B for each timed input sequence coincide. Otherwise, the TFMSMs A and B are f-distinguishable. In this case, a timed input sequence α such that $\mathbf{out}(a_0, \alpha) \neq \mathbf{out}(b_0, \alpha)$ is an *f-distinguishing* sequence for TFMSMs A and B . In the same way the notions of the f-equivalence and f-distinguishability can be defined for states of a single TFMSM.

Given a fault model $\langle A, \approx, \Phi_A \rangle$ and a set $Test$ of finite timed input sequences of A , the set $Test$ is *complete* test suite with respect to $\langle A, \approx, \Phi_A \rangle$ if for each TFMSM $B \in \Phi_A$ that is f-distinguishable with A the set $Test$ has a timed input sequence that f-distinguishes the TFMSMs A and B .

When the set Φ_A is finite, a complete test suite $Test$ can then be derived by the explicit enumeration of TFMSMs of the set Φ_A . For each TFMSM $B \in \Phi_A$ that is f-distinguishable with A we add a timed input sequence that f-distinguishes the TFMSMs A and B to the set $Test$. Thus, the main problem is how to derive a timed input sequence that f-distinguishes two input-enabled deterministic TFMSMs A and B .

3.2 Deriving F-distinguishing Sequences

Definition 4 An automaton is a 4-tuple $A = \langle A, Act, a_0, \mu_A \rangle$, where A is finite set of states with designated initial state a_0 , Act is finite set of actions and $\mu_A \subseteq A \times Act \times A$ is transition relation.

Definition 5 Let $A = \langle A, I, O, a_0, \lambda_A, \sigma_A, \Delta_A \rangle$ and $B = \langle B, I, O, b_0, \lambda_B, \sigma_B, \Delta_B \rangle$ be two deterministic and input-enabled TFMSMs. The DF-intersection $A \sqcap B$ is an automaton $D = \langle D, Act, d_0, \mu_D \rangle$, where $D \subseteq (A \times \mathbb{Z}_+ \times B \times \mathbb{Z}_+) \cup \{\mathbf{fail}\}$, $a_0 = \langle a_0, 0, b_0, 0 \rangle$, $Act = (I \times O) \cup \{1, 2, \dots, t_{max}\}$, $t_{max} = \min(\max(\Delta_A \downarrow_{\mathbb{N}}), \max(\Delta_B \downarrow_{\mathbb{N}}))$. D is minimal set driven by the following rules for μ_D (assuming that a state $\langle a, k_1, b, k_2 \rangle$ is under consideration):

- If $\Delta_A(a)$ and $\Delta_B(b)$ are defined and $(\Delta_A(a)) \downarrow_{\mathbb{N}} - k_1 = (\Delta_B(b)) \downarrow_{\mathbb{N}} - k_2$, then $\langle a, k_1, b, k_2 \rangle \xrightarrow{k} \langle a', 0, b', 0 \rangle$, where $k = (\Delta_A(a)) \downarrow_{\mathbb{N}} - k_1$, $a' = (\Delta_A(a)) \downarrow_A$, $b' = (\Delta_B(b)) \downarrow_B$.
- If $\Delta_A(a)$ and $\Delta_B(a)$ are defined and $(\Delta_A(a)) \downarrow_{\mathbb{N}} - k_1 < (\Delta_B(b)) \downarrow_{\mathbb{N}} - k_2$, then $\langle a, k_1, b, k_2 \rangle \xrightarrow{k} \langle a', 0, b, k_2 + k \rangle$, where $k = (\Delta_A(a)) \downarrow_{\mathbb{N}} - k_1$, $a' = (\Delta_A(a)) \downarrow_A$.
- If $\Delta_A(a)$ and $\Delta_B(b)$ are defined and $(\Delta_A(a)) \downarrow_{\mathbb{N}} - k_1 > (\Delta_B(b)) \downarrow_{\mathbb{N}} - k_2$, then $\langle a, k_1, b, k_2 \rangle \xrightarrow{k} \langle a, k_1 + k, b', 0 \rangle$, where $k = (\Delta_B(b)) \downarrow_{\mathbb{N}} - k_2$, $b' = (\Delta_B(b)) \downarrow_B$.
- If $\Delta_A(a)$ is defined and $\Delta_B(b)$ is not defined, then $\langle a, k_1, b, k_2 \rangle \xrightarrow{k} \langle a', 0, b, 0 \rangle$, where $a' = (\Delta_A(a)) \downarrow_A$ and $k = (\Delta_A(a)) \downarrow_{\mathbb{N}} - k_1$.
- If $\Delta_A(a)$ is not defined and $\Delta_B(b)$ is defined, then $\langle a, k_1, b, k_2 \rangle \xrightarrow{k} \langle a, 0, b', 0 \rangle$, where $b' = (\Delta_B(b)) \downarrow_B$ and $k = (\Delta_B(b)) \downarrow_{\mathbb{N}} - k_2$.
- If for an action-pair $\langle i, o \rangle \in I \times O$ there exist $a' \in A$ and $b' \in B$ such that $\langle a, i, o, a' \rangle \in \lambda_A$ and $\langle b, i, o, b' \rangle \in \lambda_B$, then $\langle a, k_1, b, k_2 \rangle \xrightarrow{\langle i, o \rangle} \langle a', 0, b', 0 \rangle$.

- If for an action-pair $\langle i, o \rangle \in I \times O$ there exists $a' \in A$ such that $\langle a, i, o, a' \rangle \in \lambda_A$ and for any $b' \in B$ $\langle b, i, o, b' \rangle \notin \lambda_B$, then $\langle a, k_1, b, k_2 \rangle \xrightarrow{\langle i, o \rangle} \text{fail}$.
- If for an action-pair $\langle i, o \rangle \in I \times O$ there exists $b' \in B$ such that $\langle b, i, o, b' \rangle \in \lambda_B$ and for any $a' \in A$ $\langle a, i, o, a' \rangle \notin \lambda_A$, then $\langle a, k_1, b, k_2 \rangle \xrightarrow{\langle i, o \rangle} \text{fail}$.

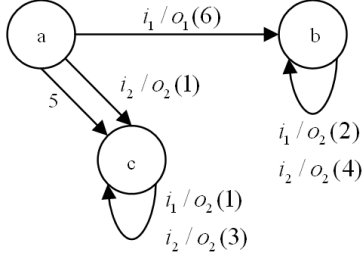


Figure 2. TFSM B

Consider TFSM A in Fig. 1 and TFSM B in Fig. 2. The DF-intersection $A \sqcap B$ is shown in Fig. 3. The automaton $A \sqcap B$ has the designated fail state.

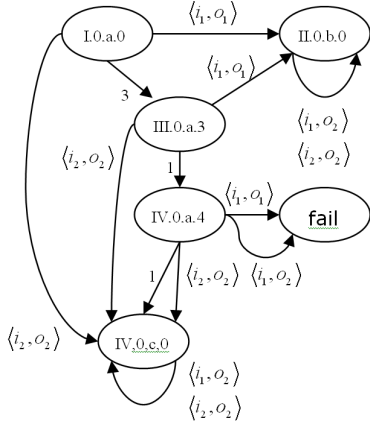


Figure 3. DF-intersection $A \sqcap B$

By construction of the automaton $A \sqcap B$, a state $\langle a, k_1, b, k_2 \rangle$ can be reached from the initial state if and only if there exists a timed input sequence that takes the TFSM A from the initial state to the state a while taking the TFSM B from the initial state to the state b, that is, the following proposition holds.

Proposition 1 *Given two deterministic and input-enabled TFSMs A and B, the automaton $A \sqcap B$ has a state $\langle a, k_1, b, k_2 \rangle$ for some k_1 and k_2 if and only if there exists a timed input sequence that takes the TFSM A*

from the initial state to state a while taking the TFSM B from the initial state to the state b.

An appropriate timed input sequence mentioned in Proposition 1 can be calculated by summarising sequential timeouts through the corresponding path of the automaton $A \sqcap B$. Given sequence $\nu = e_1 \cdot e_2 \cdot \dots \cdot e_n \in \text{Act}^*$, let $A \sqcap B$ be defined for the sequence ν and let $\nu \downarrow_{I \times O} = e_{k_1} \cdot e_{k_2} \cdot \dots \cdot e_{k_m} = \langle i_{k_1}, o_{k_1} \rangle \cdot \langle i_{k_2}, o_{k_2} \rangle \cdot \dots \cdot \langle i_{k_m}, o_{k_m} \rangle$. Underlying timed input sequence is

$$\alpha = \langle i_{k_1}, \sum_1^{k_1-1} e_j \rangle \cdot \dots \cdot \langle i_{k_m}, \sum_{k_{m-1}+1}^{k_m-1} e_j \rangle.$$

Here, as usual, we assume, that $\sum_{k_1}^{k_2} e_j = 0$, when $k_2 < k_1$. Thus, if sequence ν takes the automaton $A \sqcap B$ from the initial state to the state fail, then, due to construction, $e_{k_m} = e_n$, and the timed input sequence α f-distinguishes the TFSMs A and B, since the TFSMs have different output responses to this timed input sequence. This gives a way of checking by an experiment, which one of two known TFSMs – A or B – is given as a system under test. We just need to build the automaton $A \sqcap B$ and if the automaton $A \sqcap B$ has state fail and if it does then we derive a sequence ν that takes the automaton $A \sqcap B$ from the initial state to the state fail. Then, getting from ν timed input sequence α as shown above, we apply α to a system under test and after the observation of an output sequence we can draw a conclusion which one of two TFSMs is “hidden in the black box”.

Theorem 1 *Given two deterministic and input-enabled TFSMs A and B, the following statements hold:*

1. *If the automaton $A \sqcap B$ has the state fail and it is reachable from the initial state, then TFSMs A and B are f-distinguishable.*
2. *If a sequence $k_{1,1} \cdot \dots \cdot k_{1,r_1} \cdot \langle i_1, o_1 \rangle \cdot \dots \cdot k_{m,1} \cdot \dots \cdot k_{m,r_m} \cdot \langle i_m, o_m \rangle$, where set $\{k_{j,1}, \dots, k_{j,r_j}\} \subset \mathbb{N}$, $j = \overline{1, m}$, may be empty, $\{\langle i_1, o_1 \rangle, \dots, \langle i_m, o_m \rangle\} \subseteq I \times O$, takes the automaton $A \sqcap B$ from the initial state to the state fail, then the timed input sequence $\langle i_1, k_1 \rangle \cdot \dots \cdot \langle i_m, k_m \rangle$, $k_j = \sum_{p=1}^{r_j} k_{j,p}$, $j = \overline{1, m}$, f-distinguishes the TFSMs A and B.*

Consider the DF-intersection $A \sqcap B$ in Fig. 3. The automaton $A \sqcap B$ has the state fail, reachable from the initial state (I, 0, a, 0) by the sequence 3.1. $\langle i_1, o_1 \rangle$. According to Theorem 1, a timed input sequence (rather short, but still) $\langle i_1, 3 + 1 \rangle = \langle i_1, 4 \rangle$ f-distinguishes TFSMs A and B, since the output response of the TFSM A in the initial state I to the timed input $\langle i_1, 4 \rangle$

is o_2 , while the output response of the TFMSM B in the initial state a is o_1 , so $\mathbf{out}(I, \langle i_1, 4 \rangle) \equiv \{o_2\} \neq \mathbf{out}(a, \langle i_1, 4 \rangle) \equiv \{o_1\}$.

As discussed above, using the results of this section, a test suite with the guaranteed fault coverage can be constructed by the explicit enumeration of TFMSMs of the fault domain. Moreover, for each pair $\langle a, b \rangle \in I \times O$, the automaton $A \sqcap B$ has at most $2k \cdot |A| \cdot |B|$ states, where $k = \min(\max(\Delta_A \downarrow_{\mathbb{N}}), \max(\Delta_B \downarrow_{\mathbb{N}}))$ (note, that this evaluation is very rough and can be improved by more rigorous analysis). Thus, similar to untimed FSMs [2], given two f-distinguishable deterministic TFMSMs A and B , the length of an f-distinguishable sequence is at most $2k \cdot |A| \cdot |B| - 1$ and all the methods for deriving complete test suites for untimed FSMs can be tried for deriving a complete test suite for TFMSMs with respect to the f-equivalence relation.

3.3 Using Other Conformance Relations

Additionally, Proposition 1 gives a guide how to derive a test suite with the guaranteed fault coverage with respect to other conformance relations when we are interested in how fast an output to an applied input an implementation system produces. According to Proposition 1, all possible pairs of states of the specification and an implementation TFMSMs are states of the automaton $A \sqcap B$ and thus, this information can be also obtained using this automaton. The only thing we need when constructing an intersection, is to add new rules for transition relation. When we are interested in not “slower” implementations, then for a state $\langle a, k_1, b, k_2 \rangle$ for some input-output pair $\langle i, o \rangle$ we add transition to the state **fail**, if the output o for timed input $\langle i, k_1 \rangle$ can be produced by the specification A in state a *earlier* than the same output o for timed input $\langle i, k_2 \rangle$ can be produced by the implementation B . When we are interested in not “faster” implementations, then for a state $\langle a, k_1, b, k_2 \rangle$ for some input-output pair $\langle i, o \rangle$ we add transition to the state **fail**, if the output o for timed input $\langle i, k_1 \rangle$ can be produced by the specification A in state a *later* than the same output o for timed input $\langle i, k_2 \rangle$ can be produced by the implementation B .

Note, that timed sequences derived from such an intersection does not guarantee, that a distinguishing conclusion can be drawn since the definition of TFMSM does not require an output to be produced at exact moment, but within some time units, and thus this approach demands “all-weather conditions” [3].

4 Conclusion

In this paper, we have considered the problem of deriving a test suite with the guaranteed fault coverage for timed FSMs. A timed FSM takes into account two time aspects. One of them is the performance of output actions while other is related to timeouts. The expressiveness of this model is less than that of the model described in [1], since input and output actions should occur alternatively, value of the clock variable is discrete and it is reset to 0 after each (input or output) action. Nevertheless, the results of the paper can be applied for reactive systems which allow such limitations. Currently, we are continuing to study the derivation of tests for the above model extending the set of considering conformance relations.

References

- [1] R. Alur and D. L. Dill. A theory of timed automata. *Theoretical Computer Science*, 126(2):183–235, 1994.
- [2] A. Gill. *Introduction to the theory of Finite-State Machines*. McGraw-Hill, 1962.
- [3] G. Luo, A. Petrenko, and G. V. Bochmann. Selecting test sequences for partially-specified nondeterministic finite state machines. Technical Report IRO-864, 1993.
- [4] M. G. Merayo, M. Núñez, and I. Rodríguez. Extending efsms to specify and test timed systems with action durations and time-outs. *IEEE Transactions on Computers*, 57(6):835–844, 2008.
- [5] M. G. Merayo, M. Núñez, and I. Rodríguez. Formal testing from timed finite state machines. *Computer Networks*, 52(2):432–460, 2008.
- [6] A. Petrenko, N. Yevtushenko, and G. v. Bochmann. Fault models for testing in context. 1996.
- [7] J. Springintveld, F. Vaandrager, and P. D’Argenio. Testing timed automata. *Theoretical Computer Science*, 254(1-2):225–257, 2001.