Deriving Tests with Guaranteed Fault Coverage for Finite State Machines with Timeouts

Aleksandr Tvardovskii[1], Khaled El-Fakih[2(✉)], and Nina Yevtushenko[3]

[1] Tomsk State University, Tomsk, Russia
tvardal@mail.ru
[2] American University of Sharjah, Sharjah, UAE
kelfakih@aus.edu
[3] Ivannikov Institute for System Programming of the RAS, Moscow, Russia
evtushenko@ispras.ru

Abstract. In contrast to untimed FSMs, two minimal initialized FSMs with timeouts can be equivalent but not isomorphic. Accordingly, we propose an appropriate fault model and a method for complete test derivation for initialized deterministic FSMs with timeouts based on an appropriate FSM abstraction of the timed FSM specification. We also show how the same approach can be used for deriving tests for FSMs with both time guards and timeouts.

Keywords: Conformance testing · Timed finite state machines

1 Introduction

A multitude of approaches are given for test derivation from formal specifications modeled as Finite State Machines (FSMs). The W method [1] paved the way for many derivatives to work on the test derivation considering various classes of FSM specifications and Implementations Under Test (IUT). For related summary and experiments the reader may refer to [2, 3]. Extensions to the W-based methods are also considered in the context of systems with timed constraints [4, 5]. Merayo et al. [6] establish a number of conformance relations for possibly non-deterministic FSM with input and output timeouts; however, test derivation is not considered in [6]. El-Fakih et al. [7] consider test derivation and assessment for timed FSMs with timed guards and single clock that is reset at every transition. Zhigulin et al. [8] presented a method for deriving complete test suites for FSMs with timeouts considering a traditional fault domain assuming that the number of states of an implementation TFSM does not exceed that of the reduced specification TFSM as well as the maximal finite timeout of the IUT does not exceed this of the specification. Recently, Bersolin et al. [9] investigated many timed FSM models with a single clock.

In this paper, we consider complete test derivation against FSMs with timeouts, hereafter denoted as TFSMs. In contrast to untimed FSMs, we show that two minimal initialized TFSMs can be equivalent but not isomorphic; moreover, we show that these

I. Medina-Bulo et al. (Eds.): ICTSS 2018, LNCS 11146, pp. 149–154, 2018.
https://doi.org/10.1007/978-3-319-99927-2_13

TFSMs can have different number of states. According to [9], the behavior of a TFSM can be completely described by its corresponding (untimed) FSM abstraction and the reduced initially connected forms of corresponding FSM abstractions of two initialized equivalent TFSMs are isomorphic. This hints that the fault model and complete test derivation can be developed based on the reduced form of the FSM abstraction of a given TFSM specification. We consider complete test derivation with respect to an appropriate fault domain that contains every TFSM over the same input alphabet as the specification such that the reduced form of the FSM abstraction of an IUT has at most $m > 1$ states, and thus, the proposed approach is easily extended to FSMs with timeouts and timed guards.

2 Preliminaries

An initialized FSM is a 5-tuple $S = (S, I, O, h_S, s_0)$ where I and O are input and output alphabets, S is a finite non-empty set of states with the designated initial state s_0, and $h_S \subseteq (S \times I \times O \times S)$ is the transition relation. We consider complete and deterministic FSMs, i.e., for each pair $(s, i) \in S \times I$ there exists exactly one transition $(s, i, o, s') \in h_S$. The equivalence and distinguishability relations between different states of FSMs are defined in a usual way [3]. It is known that given a complete deterministic initialized initially connected FSM, any two reduced initially connected forms of this FSM are isomorphic.

An *FSM with timeouts*, a TFSM for short, is an FSM annotated with a *clock* that is reset to zero at the execution of any transition. In addition, such a TFSM has input timeout transitions. When an input timeout expires at a state, the TFSM can spontaneously move to the destination state of the timeout transition while resetting the time to zero. An initialized TFSM is a 6-tuple $S = (I, S, O, h_S, \Delta_S, s_1)$ where I and O are input and output alphabets, S is the finite non-empty set of states, $h_S \subseteq S \times I \times O \times S$ is the *transition relation* and Δ_S: $\Delta_S : S \to S \times (N \cup \{\infty\})$ is the timeout function, where N is the set of positive integers: for each state, this function specifies the maximum time for waiting for an input. Given state s of TFSM S such that $\Delta_S(s) = (s', T)$, if no input is applied before the timeout T expires, S moves to state s' and the clock is set to zero. If $s = s'$ then the clock is set to zero when timeout is expired. The transition $(s, i, o, s') \in S \times I \times O \times S$ means that S being at state s accepts an input i applied at time $t < T$ measured from the moment when the clock was reset at state s of S; the clock then is set to zero and S produces o. Hereafter, the timeout at state s can be written as T_s or T when s is known from the context, for short.

TFSM S is a *deterministic complete* TFSM if for each pair $(s, i) \in S \times I$, there is exactly one transition $(s, i, o', s') \in h_S$. In this paper, we consider only deterministic complete TFSMs. TFSM is (initially) connected if each state is reachable from the initial state. Given a TFSM S, a *timed input* is a pair (i, t) where $i \in I$ and t is a real; a timed input (i, t) means that input i is applied to the TFSM at time instance t where t is a local time. A sequence of timed inputs $\alpha = (i_1, t_1) \ldots (i_n, t_n)$ is a *timed input sequence*. A sequence $\alpha/\gamma = (i_1, t_1)/o_1 \ldots (i_n, t_n)/o_n$ of consecutive pairs of timed inputs and outputs starting at the state s is a *timed trace* of TFSM S at state s. Given complete deterministic TFSMs S and P, states s of S and p of P are *equivalent* if output responses

at these states coincide for each timed input sequence; otherwise, s and p are *distinguishable*. Two initialized TFSMs S and P are *equivalent* if their initial states are equivalent. If any two different states of TFSM S are distinguishable then S is (*state*) *reduced* or *minimal*.

Consider two complete deterministic TFSMs in Fig. 1 which are equivalent. Each state in S_1 (a) and S_2 (b) is reachable from the initial state and both machines are reduced. However, these two equivalent machines are not isomorphic; moreover, they have different number of states.

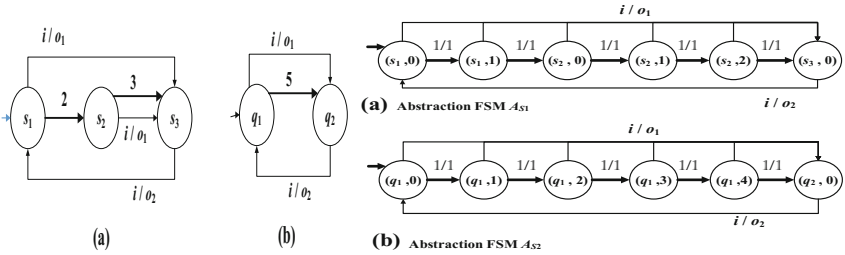

Fig. 1. Two equivalent yet not isomorphic TFSMs S_1 (a) and S_2 (b) and their FSM abstractions.

In order to calculate an output for a timed input (i, t) for each state s of TFSM S we consider the function $time(s,t) = s'$ that determines state s' that will be reached by S through timeouts if no input was applied during t time units. The output response β of S to a sequence $\alpha = (i_1, t_1)(i_2, t_2)\ldots(i_n, t_n)$ at state s is iteratively determined starting from state s.

Determining if two states of a TFSM S are equivalent or distinguishable can be done using the (untimed) FSM-abstraction A_S of S defined in [9].

FSM Abstraction: Given a complete deterministic TFSM $S = (S, I, O, h_S, \Delta_S, s_0)$, we derive the FSM abstraction of S as the FSM $A_S = (S_A, I_A, O_A, \lambda_{AS}, (s_0, 0)$, where $I_A = I \cup \{1\}, O_A = O \cup \{1\}$. The input (output) 1 is a special input (output) of the FSM abstraction denoting the time duration. For each state s, the set S_A has a state $(s, 0)$. Moreover, for each state s where the timeout T_s is finite, the set S_A has the states $\{(s, 1), .., (s, T_s - 1)\}$. Given state $(s, t_j) \in S_A$ of A_S and input i, a transition $((s, t_j), i, o, (s', 0))$ is a transition of the abstraction A_S iff there exists a transition $(s, i, o, s') \in h_S$. Transitions under the input 1 correspond to timeout transition between states. Given state s such that $\Delta_S(s) = (s', T_s)$ where $1 < T_s < \infty$, there are transitions $((s, 0), 1, 1, (s, 1)), \ldots, ((s, T_s - 2), 1, 1, (s, T_s - 1)), ((s, T_s - 1))$, in λ_{AS}. If $\Delta_S(s) = (s', T_s)$ then there is a transition $((s, T_s - 1), 1, 1, (s', 0))$ while there is a transition $((s, 0), 1, 1, (s, 0)) \in \lambda_{AS}$ iff $T_s = \infty$. In [9], it is shown that the FSM abstraction of a complete and deterministic TFSM S is also complete and deterministic. As an example, consider the FSMs S_1 and S_2 in Fig. 1(a) and (b), their corresponding isomorphic FSM abstractions A_{S1} and A_{S2} are also shown in Fig. 1.

By definition, given an FSM with timeouts with n states and k inputs, the corresponding FSM abstraction has $(k + 1)$ inputs and the number of states of the FSM

abstraction equals $\sum_{s \in S'} (T_s + |S \backslash S'|$ where S' is the subset of all FSM states for which the timeout T_s is finite.

A timed input sequence α of TFSM S can be transformed into a corresponding input sequence α_{FSM} of the FSM abstraction A_S. In this case, each timed input (i, t) is replaced by sequence $1.1 \ldots 1.\ i$ of inputs of the FSM abstraction where the number of inputs 1 equals t. At the same time the response of the FSM abstraction to sequence $1.1 \ldots 1.\ i$ is the sequence $1.1 \ldots 1.\ o$ where the number of outputs 1 is the same as for the timed input (i, t) and o is the response of the TFSM to timed input (i, t). Thus, the output sequence of the FSM abstraction γ_{FSM} is exactly the output sequence γ after removing all outputs 1. As there is no ambiguity, we further do not distinguish sequences γ_{FSM} and γ.

Proposition 1. Given a complete deterministic TFSM S and its corresponding FSM abstraction A_S, a timed trace α/γ exists for TFSM S if and only if there exists a trace α_{FSM}/γ for the FSM abstraction A_S.

Proposition 2 [9]. Two complete deterministic TFSMs are equivalent if and only if their FSM abstractions are equivalent.

The following proposition describes an input sequence that distinguishes two non-equivalent TFSMs.

Proposition 3. Given two non-equivalent complete deterministic TFSMs S and P over the same input and output alphabets, let A_S and A_P be their FSM abstractions. If an input sequence $\alpha_{FSM} = 1.1 \ldots 1.\ i_1 \ldots 1.1 \ldots 1.\ i_k$ distinguishes FSM abstractions A_S and A_P, then the timed input sequence $(i_1, t_1) \ldots (i_k, t_k)$ where t_j is the number of inputs before the input i_j, $1 \leq j \leq k$, distinguishes machines S and P.

An FSM abstraction of a TFSM can be reduced using a traditional way. Then the FSM abstraction of a TFSM implementation can be compared with the FSM abstraction of the specification TFSM and if they are not equivalent then corresponding TFSMs can be distinguished by some input sequence α_{FSM}. Moreover, a corresponding timed input sequence α will distinguish the TFSM implementation from the specification TFSM (Proposition 3). Correspondingly, a complete test suite can be derived based on the minimal form of the FSM abstraction of the specification TFSM. Such a test suite is derived for timed sequences over local time and later we discuss how the test cases can be written over global time. We also note that when distinguishing two initialized deterministic complete FSMs A_S and A_P, a distinguishing input can be only $i \in I$, as input 1 is defined at each state with the output 1. The sequence $\alpha_{FSM}.\ i$ distinguishes FSMs A_S and A_P and based on it a corresponding distinguishing sequence for TFSMs S and P can be constructed (Proposition 3).

When applying test cases to an IUT, we reasonably assume that each transition is performed with some small output delay θ such that the sum of all delays during a test case application is less than 1 and since timeouts are integers and these delays are very small they do not effect a proposed fault model.

3 Fault Models and Test Derivation

Given a specification TFSM S, we consider the fault model $<S, \cong, FD_m>$, where FD_m contains every TFSM P over the same input alphabet as S such that the reduced form of the FSM abstraction of P has at most $m > 1$ states. We note that it can well happen that some timed FSMs with less states than the specification TFSM are not included into the fault domain and vice versa a number of timed FSMs with more states than the specification TFSM are included into the fault domain.

Algorithm 1: Test Derivation Algorithm
Input : The deterministic complete specification FSM S with timeouts
Output: A complete test suite w.r.t. the fault model $<S, \cong, FD_m>$
Step 1. Derive the reduced form of the FSM abstraction A_S of S
Step 2. Derive using the W-method (or any of its derivatives) a complete test suite TS^{A_S} for the fault model $<A_S, \cong, \Omega_m>$ where Ω_m contains every minimal FSM with up to m states.
Step 3. Transform test cases of the test suite TS^{A_S} into corresponding timed sequences over the TFSM S (according to Proposition 1) and obtain TS^S. Transform the sequences of TS^S into timed input sequences over global time (by adding a negligible output delay θ) and obtain the test suite TS.

Theorem 1. The test suite TS obtained by Algorithm 1 is complete with respect to the fault model $<S, \cong, FD_m>$.

4 Deriving Tests for FSMs with Timed Guards and Timeouts

In [9], FSMs with timed guards and timeouts are considered. Input timed guards describe the behavior at a given state for inputs, which arrive at different time instances. Formally, an initialized TFSM is a 6-tuple $S = (I, S, O, h_S, \Delta_S, s_0)$ where I and O are input and output alphabets, S is the finite non-empty set of states, $h_S \subseteq S \times I \times O \times S \times \Pi$ is the *transition relation* and Δ_S is the timeout function. The set Π is a set of *input timed guards.* An input timed guard $g \in \Pi$ describes the time domain when a transition can be executed and is given in the form of interval $\lceil min, max \rceil$ from $[0; T)$, where $\lceil \in \{(, [\}, \rceil \in \{),]\}$ and T is the value of the (input) timeout at the current state. The transition $(s, i, o, s', g) \in S \times I \times O \times S \times \Pi$ means that TFSM S being at state s accepts an input i applied at time $t \in g$ measured from the moment when S entered state s; the clock then is set to zero and S produces output o. TFSM S is a *deterministic complete* TFSM if for each two transitions $(s, i, o_1, s_1, g_1), (s, i, o_2, s_2, g_2) \in h_s$ it holds that $g_1 \cap g_2 = \emptyset$ and the union of all input timed guards at state s under input i equals $[0; T)$ when $\Delta_S(s) = (s', T)$. Given a complete deterministic TFSM S, the largest finite boundary B_S of input timed guards and timeouts, we derive the FSM abstraction of S as the FSM $A_S(B) = (S_A, I \cup \{1\}, O \cup \{1\}, \lambda_{AS}, (s_0, 0)), B \geq B_S$, where $S_A = \{(s, 0), (s, (0, 1)), \ldots, (s, (B-1, B)), (s, B), (s, (B, \infty)) : s \in S\}$. In [9], it is shown that such an FSM abstraction of a complete and deterministic TFSM S is also complete and deterministic and a timed input sequence α of TFSM S can be transformed into a

corresponding input sequence α_{FSM} of the FSM abstraction $A_S(B)$ similar to an FSM with timeouts. We then consider the fault model $< S, \cong, FD_m(B) >$, where $FD_m(B)$ contains every TFSM P over the same input alphabet as S such that the reduced form of the FSM abstraction of P has at most $m > 1$ states and the largest finite boundary of input timed guards and timeouts is $B \geq B_S$. In our case, the test derivation technique completely coincides with Algorithm 1 where the FSM abstraction A_S is considered and the test suite TS obtained by Algorithm 1 is complete w.r.t. the fault model $< S, \cong, FD_m(B) >$.

5 Conclusion

A proper fault domain is considered for complete test derivation against timed FSMs. The fault domain takes into account the fact that a reduced TFSM specification and a reduced TFSM implementation with timeouts can be equivalent yet not isomorphic. A proper characterization of the fault domain is then considered using the unique reduced form of the FSM abstraction of the given timed FSM specification. The fault domain is extended to consider FSMs with timeouts and timed guards.

Acknowledgement. This work is partly supported by Russian Science Foundation (RSF), Project No. 16-49-03012.

References

1. Chow, T.S.: Testing software design modeled by finite-state machines. IEEE TSE **4**(3), 178–187 (1978)
2. Simao, A., Petrenko, A., Maldonado, J.C.: Comparing finite state machine test coverage criteria. IET Softw. **3**(2), 91–105 (2009)
3. Dorofeeva, R., El-Fakih, K., Maag, S., Cavalli, A.R., Yevtushenko, N.: FSM-based conformance testing methods: a survey annotated with experimental evaluation. Inf. Softw. Technol. **52**, 1286–1297 (2010)
4. Springintveld, J., Vaandrager, F., D'Argenio, P.: Testing timed automata. Theor. Comput. Sci. **254**(1–2), 225–257 (2001)
5. En-Nouaary, A., Dssouli, R., Khendek, F.: Timed Wp-method: testing real-time systems. IEEE TSE **28**(11), 1023–1038 (2002)
6. Merayo, M.G., Nunez, M., Rodriguez, I.: Formal testing from timed finite state machines. Comput. Netw. **52**(2), 432–460 (2008)
7. El-Fakih, K., Yevtushenko, N., Simao, A.: A practical approach for testing timed deterministic finite state machines with single clock. Sci. Comput. Program. **80**(1), 343–355 (2014)
8. Zhigulin, M., Yevtushenko, N., Maag, S., Cavalli, A.: FSM-based test derivation strategies for systems with time-outs. In: International Conference on Quality Software, pp. 141–150 (2011)
9. Bersolin, D., El-Fakih, K., Villa, T., Yevtushenko, N.: Timed finite state machines: equivalence checking and expressive. In: International Symposium on Games, Automata, Logic and Formal Verification, pp. 203–216 (2014)