Refining the Specification FSM When Deriving Test Suites w.r.t. the Reduction Relation

Aleksandr Tvardovskii

Refining the specification FSM when deriving test suites w.r.t. the reduction relation

Aleksandr Tvardovskii

National Research Tomsk State University, 36 Lenin str., Tomsk, Russia
tvardal@mail.ru

Abstract. Finite State Machines (FSMs) are widely used when deriving tests for components of discrete event systems. In general, the specification FSM can be nondeterministic and in this case, a test suite with the guaranteed fault coverage is derived with respect to the reduction relation. However, when deriving such tests for nondeterministic FSMs, the existing methods return rather long test suites which cannot be used for real systems. In order to shorten a test suite, the set of possible implementation FSMs can be reduced. We present an approach for deriving shorter test suites for nondeterministic FSMs with respect to the reduction relation via refining the specification FSM.

Keywords: Finite State Machines (FSM), nondeterministic FSM, test derivation.

1 Introduction

Finite State Machine (FSM) based test derivation is an active research area that has a long history [1, 2]. The well-known method is the W-method [2] and many derivatives of this method have been developed including those for FSMs with the nondeterministic behaviour [see, for example, [3, 4, 5]. In FSM based testing, the specification behaviour and the behaviour of an implementation under test (IUT) are described by FSMs and by applying input sequences to the IUT and observing the produced outputs a tester should conclude whether the IUT conforms to its specification. Best known conformance relations are the equivalence and reduction relations [4]. In the former case, the IUT has to have the same behaviour as the specification FSM; in the latter case, the behaviour of the IUT has to be contained in the behaviour of the specification FSM.

In this paper, we propose to refine the FSM specification via deleting some transitions in such a way that the refined specification has an (adaptive) distinguishing sequence that distinguishes every two different states and all the states are definitely reachable from the initial state [5], i.e., each state is (adaptively) reachable from the initial state. Under such conditions, the length of a test suite against nondeterministic FSMs is comparable with that for deterministic FSMs.

The rest of the paper has the following structure. Section 2 contains the preliminaries. A procedure for deriving complete test suites against nondeterministic FSMs with respect to the reduction relation is presented in Section 3. Section 4 contains a proposed procedure for reducing the specification FSM. Section 5 concludes the paper.

2 Preliminaries

In this section, we introduce necessary definitions and notations which are mainly taken from the paper [5].

A *finite state machine* (FSM), or simply a *machine*, is a 5-tuple $\mathsf{S} = \langle S, I, O, h_S, s_0 \rangle$ where S is a finite nonempty set of states with the designated state s_0, I and O are finite input and output alphabets, and $h_S \subseteq S \times I \times O \times S$ is a *transition* (*behavior*) *relation*. FSM S is *nondeterministic* if for some pair $(s, i) \in S \times I$, there can exist several pairs $(o, s') \in O \times S$ such that $(s, i, o, s') \in h_S$; otherwise, the FSM is deterministic. FSM S is *complete* if for each pair $(s, i) \in S \times I$ there exists $(o, s') \in O \times S$ such that $(s, i, o, s') \in h_S$; otherwise, the FSM is *partial*. FSM S is *observable* if for every two transitions (s, i, o, s_1), $(s, i, o, s_2) \in h_S$ it holds that $s_1 = s_2$. In the following, we consider complete observable possibly nondeterministic FSM specifications, while an implementation is a complete deterministic FSM.

Figure 1 shows a FSM A for which $I = \{i_1, i_2, i_3\}$, $O = \{o_1, o_2, o_3\}$, $S = \{1, 2, 3\}$ and 1 is the initial state. Suppose that input i_1 is applied to this FSM at the state 1. After applying input i_1 the FSM can remain at state 1 and produce an output o_1. However, the FSM can also stay at state 1 while producing output o_2.

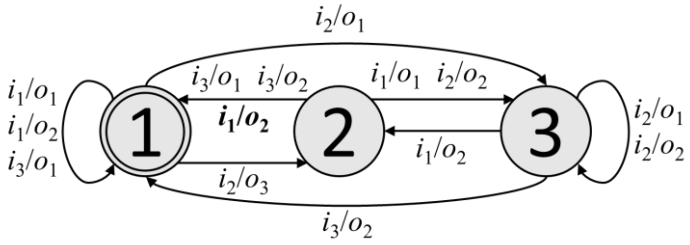

Fig. 1. FSM A

A *trace* of FSM S at state s is a sequence of input/output pairs $\alpha = i_1/o_1 \ldots i_l/o_l$ of consecutive transitions starting from state s. A sequence of inputs $i_1 \ldots i_l$ is an *input* sequence, a sequence of outputs $o_1 \ldots o_l$ is an *output* sequence.

Given an input alphabet I and an output alphabet O, a *test case* $\mathsf{TC}(I, O)$ is an initially connected observable FSM $\mathsf{T} = (T, I, O, h_T, t_0)$ with an acyclic transition graph such that at each state only one input with all possible outputs is defined. Given a complete FSM S over alphabets I and O, a test case $\mathsf{TC}(I, O)$ represents an adaptive experiment with the FSM S. If $|I| > 1$ then a test case is a partial FSM. A state $t \in T$ is a *deadlock* state of the FSM T if there are no defined inputs at this state. The notion of a test case can be used for representing an adaptive input sequence when the next input depends on the output to the previous input. In general, given a test case T, the *length* (*height*) of the test case T is defined as the length of a longest trace from the initial state to a deadlock state of T and it specifies the length of the longest input sequence that can be applied to an FSM S during the experiment.

A test case T is a *distinguishing* test case (DTC) for an FSM S if for every trace γ of T from the initial state to a deadlock state, γ is trace at most at one state of S. Sometimes, a distinguishing test case is called an *adaptive distinguishing sequence*. A distinguishing test

case for a submachine of FSM A in Figure 1 without the bold transition (2, i_1, o_2, 1) is shown in Figure 2.

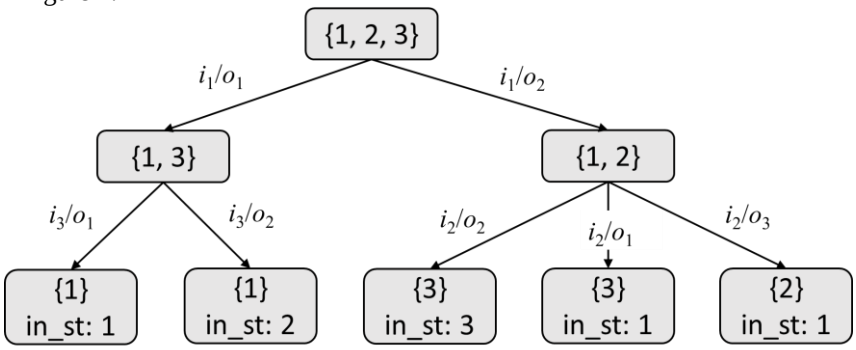

Fig. 2. A distinguishing test case for a submachine of FSM A without the bold transition (2, i_1, o_2, 1)

A test case T represents an (*adaptive*) *transfer* sequence from the initial state to state *s* if every trace of T from the initial state to a deadlock state takes the FSM from the initial state to state *s*. According to [5], if there exists an (*adaptive*) *transfer* sequence from the initial state to state *s* then state *s* is *definitely reachable* from the initial state and must be implemented in a conforming implementation [5]. Adaptive transfer sequences to states 2 and 3 for FSM A in Figure 1 is shown in Figure 3. For a deterministic FSM, a transfer sequence is simply an input sequence.

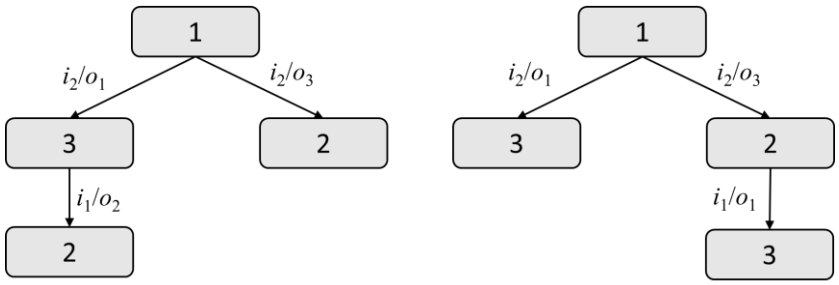

Fig. 3. Transfer test cases for FSM A

Test cases are (adaptive) input sequences which are derived against the given FSM specification to determine whether a given black-box implementation under test (IUT), which is also assumed to have the FSM behavior, conforms to the given specification. In this paper, an IUT *conforms* to the specification if an implementation FSM is a reduction of the specification FSM. In other words, an IUT conforms to the specification FSM if for each input sequence the output response of the IUT is contained in the set of output responses of the specification FSM to this input sequence. If the observed outputs do not match, then the implementation has a fault, i.e., it is a *nonconforming* implementation.

Given states *s* and *p* of complete FSMs S and P, state *p* is a *reduction* of *s*, $p \leq s$, if the set of *I*/*O* sequences of FSM P at state *p* is contained in the set of *I*/*O* sequences of FSM S at state *s*. FSM P is a *reduction* of FSM S if the reduction relation holds between the initial states of these machines.

We consider the fault domain $\mathfrak{I}_n$ which contains every deterministic complete FSM with at most n states and with the same input alphabet as the specification FSM S where n is number of states of the specification FSM. A *test suite* is a finite set of finite possibly adaptive input sequences of the specification. A test suite is *complete* with respect to the $\mathfrak{I}_n$ if for each FSM $P \in \mathfrak{I}_n$ such that P is not a reduction of S, the test suite has a sequence for which an output response is not in the set of output responses of S to this sequence. In the next section, based on the results of [4] we briefly remind how a complete test suite with respect to $\mathfrak{I}_n$ can be derived when the specification FSM has a distinguishing test case and each state is definitely reachable from the initial state.

3 Deriving a complete test

Methods for deriving tests with respect to the reduction relation are based on (adaptive) distinguishing and transfer sequences. A test suite of polynomial length can be derived under the following conditions: 1) an IUT is deterministic and does not have more states than the specification FSM; 2) the specification FSM has an (adaptive) distinguishing test case of polynomial length and each state is definitely reachable from the initial state.

An algorithm below returns a complete test suite with respect to the $\mathfrak{I}_n$ if the specification FSM has a distinguishing test case and each state is definitely reachable from the initial state. Since the length of an (adaptive) transfer sequence (if it exists) does not exceed n [5], the length of a returned test suite is polynomial with respect to the number of states of the specification FSM when a DTC possesses this feature. It is known [3] that a returned test suite can detect many other faults but the guarantee is only for FSMs with up to n states where n is the number of states of the specification FSM.

Procedure 1 **Deriving a complete test suite w.r.t. the fault domain** $\mathfrak{I}_n$

Input: A complete possibly nondeterministic observable specification FSM S with n states

Output: A complete test suite *TS* with respect to $\mathfrak{I}_n$

Step-1: **If** some state of S is not definitely reachable from the initial state or the FSM has no DTC

Then Return the message "The specification FSM does not possess the necessary features"

Else Step-2

Step-2: Derive a state cover of the FSM using (adaptive) transfer sequences;

Append every sequence of the state cover with a DTC;

Append every sequence of the state cover with all possible inputs which in turn, are appended with a DTC;

Denote *TS* the obtained set of (adaptive) input sequences;

Return the obtained set *TS* of input sequences.

According to the results in [4, 5], the following proposition holds.

Proposition 1. If the specification FSM S has a DTC and each state of S is definitely reachable from the initial state then Procedure 1 returns a complete test suite with respect to $\mathfrak{I}_n$; the length of a test suite is proportional to the product $|S|\ |I|\ |L_{\mathrm{DTC}}|$ where L_{DTC} is the length of the distinguishing test case DTC.

4 Refining the specification FSM

If the specification FSM S has no DTC or some state is not definitely reachable from the initial state then we could find a maximal submachine of S (i.e. a submachine with maximum number of transition) that possesses this feature, however, this is not always possible. For example, there is no such submachine if there are at least two states where transitions under the every input are deterministic and the FSM is taken to the same state with the same output. Nevertheless, if this is possible then we could delete some transitions from the specification FSM in order to have an FSM where each state is definitely reachable and there is a DTC of polynomial length. For example, a submachine of the FSM in Figure 1 without the bold transition (2, i_1, o_2, 1) has a DTC of length 2 and each state is definitely reachable from the initial state.

Given the specification FSM S, let S^{red} be its maximal complete submachine where each state is definitely reachable and there is a DTC of polynomial length. If $S = S^{red}$ then a test suite returned by Procedure 1 is *complete* with respect to the fault domain $\mathfrak{I}_n$. If S is not equal to S^{red} then Procedure 1 is used for deriving a test suite *TS* for FSM S^{red} and the following proposition holds.

Proposition 2. If for each input sequence of *TS* the output response of an IUT P is in the set of output responses of S^{red}, then the IUT is a reduction of the FSM S. If the output response of the IUT to some sequence of *TS* is not contained in the corresponding set of output responses of the specification FSM S to this input sequence, then the IUT is not a reduction of S.

If the output response of the IUT to some sequence of *TS* is not contained in the corresponding set of output responses of S^{red} but is contained in the set of output responses of S, then we cannot conclude whether the IUT conforms to its specification. i.e., the verdict is *inconclusive*.

When deriving distinguishing test cases, merging-free FSMs are often considered. An FSM S is *merging-free* if for every two transitions (s_1, *i*, *o*, *s*) and (s_2, *i*, *o*, *s*) it holds that states s_1 and s_2 coincide. In [6] it is shown, that a merging-free FSM S has a DTC if and only if for each pair of state of S there exists a DTC and moreover, if there exists a DTC then there exists a DTC with the length that is polynomial with respect to the number of states of S. However, in this paper, we do not derive a maximal merging-free submachine of the specification FSM. Another way to find a maximal submachine of S that possesses necessary features could be the enumeration of all submachines of the specification FSM. However, as this number is big enough, we further propose to consider only deterministic submachines of the specification FSM. Another reason for considering deterministic submachines is that if a complete deterministic FSM has an adaptive distinguishing sequence then the length of such sequence is $O(n^2)$ [7].

Procedure 2 **Refining the specification FSM**

Input: a complete possibly nondeterministic observable specification FSM S with *n* states

Output: a complete submachine S^{red} of FSM S where each state is definitely reachable and there is a DTC of reasonable length or the message "The specification FSM cannot be reduced"

Step-1: Find a set of all complete deterministic submachines of FSM S, determine a submachine S′ which has a DTC.

If such FSM does not exist

Then Return the message "The specification FSM cannot be reduced";

Else derive a DTC and a transfer sequence for each state of S'

Step-2: For each transfer sequence α from the initial state to state s of S', remove from S each transition tr such that tr is executed by S when α is applied, and transition tr breaks the property of α to be a transfer sequence to state s in S.

Step-3: For a DTC of S', remove from S each transition tr such that tr is executed by S when any sequence α from TC is applied and transition tr breaks the property of TC to be DTC for S.

Denote S^{red} the FSM obtained from S after removing transitions.

If Procedure 2 returns FSM S^{red} then S^{red} has DTC and each state is definitely reachable from the initial state, i.e., for each state of S^{red} there exists an (adaptive) transfer sequence, then a test suite returned by Procedure 1 for S^{red} has the fault coverage defined by Proposition 2.

6 Conclusions

In this paper, an approach for deriving test suites of reasonable length for nondeterministic FSMs with respect to the reduction relation has been proposed. A proposed method is based on deriving a submachine of the initial FSM specification that has a distinguishing test case of polynomial length and each state is definitely reachable from the initial state can be derived. Derived for refined specification test is not always complete, but can be used for checking an appropriate subset of implementations.

Acknowledgement

This work is partly supported by RSF Project No. 16-49-03012.

References

1. A. Gill. Introduction to the Theory of Finite-State Machines. 1964, 272 p.
2. Chow, T.S.: Test design modeled by finite-state machines. IEEE Transactions on Software Engineering, vol. 4, No 3, 1978, pp. 178-187.
3. R. Dorofeeva, K. El-Fakih, S. Maag, A.R. Cavalli, N. Yevtushenko. FSM-based conformance testing methods: a survey annotated with experimental evaluation. Information and Software Technology, 52, 2010, pp. 1286-1297.
4. A. Petrenko and N. Yevtushenko. Conformance Tests as Checking Experiments for Partial Nondeterministic FSM. Proceedings of the 5th International Workshop on Formal Approaches to Testing of Software (FATES 2005), LNCS 3997, 2005, pp. 118-133.
5. A. Petrenko and N. Yevtushenko. Adaptive Testing of Deterministic Implementations Specified by Nondeterministic FSMs, Proceedings of the 23d IFIP Int. Conference on Testing Software and Systems, Paris, France, Berlin Heidelberg: Springer-Verlag, LNCS 7019, 2011, pp. 162-178.
6. N. Yevtushenko, N. Kushik. Nondeterministic Merging-free Finite State Machines. In Proceedings of IEEE East-West Design & Test Symposium (EWDTS), 2015, pp. 338-341.
7. D. Lee, M. Yannakakis. Testing finite-state machines: state identification and verification, IEEE Transactions on Computers, vol. 43, No 3, 1994, pp. 306-320.