

Testing from Partial Deterministic FSM Specifications

Alexandre Petrenko and Nina Yevtushenko

Abstract—This paper addresses the problem of test generation from partially specified deterministic Finite State Machines (FSMs) that may have indistinguishable states and, thus, are not necessarily reduced (minimized). The known methods for checking experiments that are based on state identification are not applicable to unreduced machines. We propose the so-called State-Counting approach that is directly applicable to unreduced FSMs. The approach generalizes the idea of state identification in test generation methods for deterministic machines.

Index Terms—Finite State Machine, partially specified FSM, test generation, weak conformance testing, fault detection, state identification, checking experiment.



1 INTRODUCTION

THE problem of testing Finite State Machines (FSMs) is a long-standing research problem that can be traced back to the pioneering work of Moore [17] on machine identification and Hennie [7] on fault detection with checking experiments. It has various useful applications in automata theory, machine learning, hardware, and software testing, as well as protocol conformance testing, see, e.g., [23], [6], [9], [10], [29], [3], [25], [27], [8]. These and numerous other publications, see, e.g., [1], [30], [16], [18], [14], which provide extensive literature coverage, addressed the problem of fault detection, i.e., generate a test suite (or a single sequence if the implementation cannot be reset) from a given (specification) FSM that provides complete fault coverage of the set of all implementation FSMs within a given bound on the number of states. This is a classical checking experiment or checking sequence design problem.

The existing methods generate tests from the traditional Mealy model of FSM, where each transition is labeled by the pair of input and output. The testing literature almost universally assumes that FSM is completely specified, i.e., for each state-input combination, there is a transition, as well as that the machine is reduced, where any two states are not equivalent and, thus, can be distinguished by some input sequence. State distinguishability implies that an unknown state can always be identified; this is the fundamental idea of all state identification-based methods for test generation from FSMs.

On the other hand, it has been noticed that, in most practical situations, FSM specifications are not complete; some state-input combinations have no corresponding transitions. A partial FSM is an adequate model when, for

example, one considers a partial design or a restricted environment in which FSM operates [5]. The latter is often the case, when the design is modular and inputs of one FSM are outputs of another machine [11], [26], and [20]. At the same time, not much has yet been done to generalize test generation methods to partial FSMs.

The problem is that the classical state identification approach, as suggested in [7], may not be applicable to an arbitrary partial FSM since its states may no longer be distinguishable using only “core” or specified transitions. The core transitions have to be exclusively used to check for a *weak* conformance (quasi-equivalence [5]), as opposed to the *strong* conformance (equivalence) used for completely specified machines. A pragmatic solution that sometimes works is to complete a partial specification in one way or another. This is the so-called *completeness* assumption, stating how the missing transitions are to be treated. Sidhu and Leung [29], for example, suggest that “for unspecified state-input combinations, we assume that the (protocol) entity produces null output and remains in its present state.” Yannakakis and Lee [32] mention an alternative completeness assumption, namely, “if a transition is not in the core, then the machine makes a transition to an error state with error output symbol.” However, as [32] argues, “in any case, a partially specified machine augmented with a completeness assumption can be regarded as a fully specified machine; thus, strong conformance testing is essentially the same as testing the fully specified machine with the missing transitions included and does not present any new problem.” Similar observations are presented in [16]. On the other hand, when an FSM has to be tested via another (context) FSM, any completeness assumptions are actually useless, the context FSM may never excite noncore transitions in the FSM under test, see for details [11], [26], and [20]. Test generation from a reduced partial FSM whose core transitions are sufficient to distinguish states was addressed, as far as we know, only in [33] and [32]. However, we are not aware of any method for generating a complete test suite from unreduced partial FSMs whose

• A. Petrenko is with CRIM, Centre de Recherche Informatique de Montréal, 550 Sherbrooke Street West, Suite 100, Montreal, H3A 1B9 Canada. E-mail: petrenko@crim.ca.

• N. Yevtushenko is with Tomsk State University, 36 Lenin Street, Tomsk, 634050, Russia. E-mail: yevtushenko.RFF@elefot.tsu.ru.

Manuscript received 8 Jan. 2004; revised 15 Mar. 2005; accepted 5 May 2005; published online 15 July 2005.

For information on obtaining reprints of this article, please send e-mail to: tc@computer.org, and reference IEEECS Log Number TC-0007-0104.

states cannot be distinguished with core transitions. The goal of this paper is to propose such a method.

The rest of the paper is organized as follows: In Section 2, we give necessary basic notions. In Section 3, we discuss the structure of tests for various types of FSMs, which may have compatible, quasi-equivalent, and distinguishable states, and propose a method for test generation from partial FSMs with complete fault coverage. We also discuss the complexity of tests obtained with the proposed method. In Section 4, we relate the results with previous work and indicate possible future work. Section 5 concludes the paper.

2 PRELIMINARIES

A *Finite State Machine* (FSM), often simply called a machine throughout this paper, is a deterministic Mealy machine which can be formally defined as follows:

Definition 1. An FSM A is a 7-tuple $(S, s_0, X, Y, D_A, \delta, \lambda)$, where

- S is a finite set of states with the initial state s_0 ,
- X is a finite set of inputs,
- Y is a finite set of outputs,
- $D_A \subseteq S \times X$ is a specification domain,
- δ is a transition function $\delta: D_A \rightarrow S$, and
- λ is an output function $\lambda: D_A \rightarrow Y$.

An FSM A is said to be *completely specified* (a complete FSM) if $D_A = S \times X$. We will omit the specification domain D_A in the case of complete machines. If D_A is a proper subset of $S \times X$, then A is called a *partially specified machine* (a partial FSM). Given a string $\alpha = x_1 \dots x_k$ of the set X^* of all possible finite input sequences, α is said to be a *defined input sequence* at state $s \in S$ if there exist states s_1, \dots, s_k, s_{k+1} , where $s_1 = s$, such that $(s_i, x_i) \in D_A$ and $\delta(s_i, x_i) = s_{i+1}$ for all $i = 1, \dots, k$. In words, α is a defined input sequence at state s if the behavior of A for the sequence α is defined. We use $\Omega_A(s)$ to denote the set of all defined input sequences for state s and Ω_A for state s_0 , i.e., for A . We note that, for a complete machine, $\Omega_A(s) = X^*$ for any state $s \in S$, while a partial FSM can have states where the set of defined sequences is empty. Such states may occur in FSM modeling, e.g., a partial design, when testing has to be performed only on the defined part of a system.

We extend the transition and output functions from input symbols to defined input sequences, including the empty sequence ε , as usual. Let $\delta(s, \varepsilon) = s$ and $\lambda(s, \varepsilon) = \varepsilon$ for any $s \in S$. Suppose that β is a defined input sequence at state s and $\delta(s, \beta) = s'$. Then, for any $x \in X$ such that $(s', x) \in D_A$, we define $\delta(s, \beta x) = \delta(s', x)$ and $\lambda(s, \beta x) = \lambda(s, \beta)\lambda(s', x)$. For simplicity, we use the same notations δ and λ for the extended functions.

Given states $s, t \in S$, a sequence $\alpha \in \Omega_A(s)$ such that $\delta(s, \alpha) = t$ is a *transfer sequence* from s to t . For any state s , the empty sequence ε is a transfer sequence from s to s . A transfer sequence, if we do not specify from which state it is, starts, by default, from the initial state. We say that the sequence α , applied at state s , *traverses* state t if there exists a nonempty prefix β of α such that β is a transfer sequence from s to t . We also say that a set of sequences *traverses* a given state if it contains at least one sequence that traverses

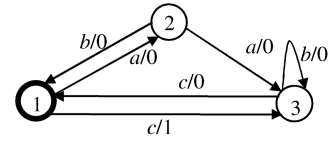


Fig. 1. The reduced partial FSM A ; the input set is $\{a, b, c\}$, the output set is $\{0, 1\}$, the initial state is depicted in bold.

the state. A transfer sequence from state s is said to be *acyclic* if it does not traverse state s and, in addition, it traverses each state at most once; otherwise, it is *cyclic*. FSM A is said to be *connected* if all its states are reachable from the initial state, i.e., for any state $s \in S$, there exists a transfer sequence $\alpha \in \Omega_A$ from s_0 to s . We consider here only connected FSMs (each FSM can be transformed into a connected FSM by removing states unreachable from the initial state).

A *state cover* of FSM A with n states is defined as a set of n transfer sequences, including the empty sequence, that take the machine A from the initial state to every state. An input sequence $\gamma \in \Omega_A$ is said to *cover* a transition from state s with input x if it can be represented as a concatenation $\gamma = \alpha x \beta$ such that α is a transfer sequence to s . We also say that a given set of sequences *covers* a given transition if it has at least one sequence that covers the transition. We will use several relations between states, defined in the following:

Definition 2. Given an FSM $A = (S, s_0, X, Y, D_A, \delta, \lambda)$ and states $s, t \in S$,

- s and t are compatible, written $s \sim t$, if $\Omega_A(s) \cap \Omega_A(t) = \emptyset$ or $\lambda(s, \alpha) = \lambda(t, \alpha)$ for all $\alpha \in \Omega_A(s) \cap \Omega_A(t)$;
- t is quasi-equivalent to s , written $t \supseteq s$, if $\Omega_A(t) \supseteq \Omega_A(s)$ and $\lambda(s, \alpha) = \lambda(t, \alpha)$ for all $\alpha \in \Omega_A(s)$;
- s and t are equivalent, written $s \cong t$, if $\Omega_A(s) = \Omega_A(t)$ and $\lambda(s, \alpha) = \lambda(t, \alpha)$ for all $\alpha \in \Omega_A(s)$;
- s and t are distinguishable (by γ), written $s \not\sim t$ ($s \not\sim_\gamma t$), if there exists an input sequence $\gamma \in \Omega_A(s) \cap \Omega_A(t)$, called a separating sequence, denoted $\gamma(s, t)$,¹ such that $\lambda(s, \gamma) \neq \lambda(t, \gamma)$; given a set of input sequences W , we also write $s \not\sim_W t$ if $s \not\sim_\gamma t$ and $\gamma \in W$.

An FSM is *reduced* if, for every pair s, s' of states, s and s' are distinguishable; a machine with compatible states is said to be *unreduced*. In the case of a complete FSM, two distinguishable states can be distinguished by input sequence of length at most $n - 1$, where n is the number of states in the given machine. For a partial machine, the corresponding bound is $n(n - 1)/2$. To demonstrate this, it is sufficient to consider a partial reduced FSM with n states, where each input defines two transitions such that, starting from a given two states, it takes $C_n^2 - 1 = n(n - 1)/2 - 1$ inputs to traverse each pair of states until a pair of states distinguishable by a single input is reached. The shortest sequence that distinguishes starting states has length $n(n - 1)/2$. Fig. 1 presents a reduced partial FSM with three states, where a shortest separating sequence for

1. We assume that $\gamma(s, t)$ is the empty sequence if $s \sim t$.

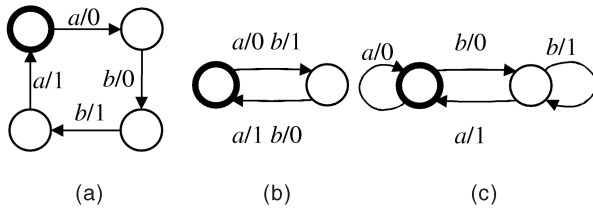


Fig. 2. The partial FSM (a) that has two distinguishable reduced forms (b) and (c).

states 1 and 2, abc , has length three. A method for determining separating sequences can be found in [15].

The notions of equivalence, quasi-equivalence, and distinguishability could also be applied to states from different machines. In particular, let $A = (S, s_0, X, Y, D_A, \delta, \lambda)$ and $B = (T, t_0, X, Y, D_B, \Delta, \Lambda)$ such that $S \cap T = \emptyset$. Considering the disjoint union of these machines, we say that B is *quasi-equivalent* to A , $B \sqsupseteq A$, if $t_0 \sqsupseteq s_0$; $A \cong B$, if $s_0 \cong t_0$; A and B are *distinguishable*, $B \not\sim A$, if $s_0 \not\sim t_0$. The quasi-equivalence relation, introduced in [5], is also called the *weak conformance* in [32] and other works.

Given $A = (S, s_0, X, Y, D_A, \delta, \lambda)$ and

$$B = (T, t_0, X, Y, D_B, \Delta, \Lambda),$$

if B is reduced and $B \sqsupseteq A$, then B is called a *reduced form* of A . Methods for constructing a reduced form of a partial machine, see, e.g., [12] and [4], are more involved than that for complete machines. As opposed to complete FSMs, partial FSMs may have several distinguishable reduced forms. The reason is that, in general, the compatibility relation is not transitive for partial FSMs.

Example 1. Consider FSM in Fig. 2a. It has two reduced forms shown in Fig. 2b and Fig. 2c. The two machines are complete FSMs, quasi-equivalent to the FSM in Fig. 2a, but they are distinguishable since, for example, input b causes different outputs when applied at the initial states.

We now define the notion of a complete test suite. Let A be a *specification* FSM and $\mathfrak{S}(A)$ be a set of *implementation* machines $\{B = (T, t_0, X, Y, D_B, \Delta, \Lambda) \mid \Omega_B \supseteq \Omega_A\}$. The set $\mathfrak{S}(A)$ is called a *fault domain*. FSM $B \in \mathfrak{S}(A)$ is a *conforming* implementation of the FSM A if $B \sqsupseteq A$ or it is a *nonconforming* implementation of the FSM A if $B \not\sim A$. A nonconforming FSM represents some implementation faults.

Definition 3. A defined input sequence of FSM A is called a *test case* (or simply a *test*) of A . A test suite for A is a finite set of tests of A . A test suite E is said to be *complete* for A w.r.t. the fault domain $\mathfrak{S}(A)$ if, for all $B \in \mathfrak{S}(A)$, $B \not\sim A$ implies $B \not\sim_E A$.

Our definition of a test suite allows multiple tests, so we assume that a reliable reset operation that brings a machine back to the initial state is available in any machine in the fault domain. A test suite is m -complete if it is complete w.r.t. the fault domain $\mathfrak{S}_m(A)$ that is the set that contains all implementation FSMs with at most m states. Clearly, an m -complete test suite is also k -complete for any $k < m$. If the specification FSM A with n states is reduced, then it is

usually assumed that $m \geq n$. The main reason is that the fault domain $\mathfrak{S}_m(A)$ for any $m < n$ does not contain any conforming implementation machine. This is not necessarily the case for an unreduced FSM A as its reduced form has fewer states, so we consider here that m is not smaller than a minimal number of states in reduced forms of A .

A naive approach for test generation from an unreduced partial FSM A would first minimize the machine, then use any existing method for test generation from a reduced FSM. As we discussed in Section 1, any completion of the original machine, including the one implied by minimization, is useless in testing FSM via context. The approach fails even when FSM is tested in isolation, in case it has several reduced forms. The partial FSM in Fig. 2a can serve as an example. If, e.g., a test suite generated from the reduced FSM shown in Fig. 2b contains a test that begins with input b , then the machine shown in Fig. 2c will be rejected as a nonconforming implementation, despite the fact that it is quasi-equivalent to the FSM in Fig. 2a.

In this paper, we propose the method for deriving an m -complete test suite directly from a given FSM which can be unreduced and partially specified. Earlier versions of the method were published in [34], [22].

3 TEST GENERATION

3.1 Complete Test Suite and Distinguishing Machine

Let A be a specification FSM and B be an implementation FSM, $B \in \mathfrak{S}_m(A)$. To check if the FSM B is quasi-equivalent to A , we use a designated FSM, called a *distinguishing machine* of A and B , whose states are pairs of states of A and B . Its initial state is the pair of initial states of the two machines; the remaining states are determined by performing reachability analysis. A designated output fail is used to signal when the two machines do not agree on a common input.

Definition 4. Given FSM $A = (S, s_0, X, Y, D_A, \delta, \lambda)$ and FSM $B = (T, t_0, X', Y', D_B, \Delta, \Lambda)$ such that $\Omega_A \subseteq \Omega_B$, the FSM $(Q, q_0, X, Y \cup \{\text{fail}\}, D, \psi, \varphi) = \nabla_{A,B}$ is the *distinguishing machine (DM)* of A and B , if $Q \subseteq S \times T$, the initial state $q_0 = (s_0, t_0)$, $\text{fail} \notin Y \cup Y'$,

$$D = \{((s, t), x) \mid (s, t) \in Q, (s, x) \in D_A\},$$

and the transition function ψ , and the output function φ are obtained using the following rules:

- $\psi((s, t), x) = (\delta(s, x), \Delta(s, x))$, if $(s, x) \in D_A$;
- $\varphi((s, t), x) = \lambda(s, x)$ if $(s, x) \in D_A$ and $\lambda(s, x) = \Lambda(t, x)$; and
- $\varphi((s, t), x) = \text{fail}$ if $(s, x) \in D_A$ and $\lambda(s, x) \neq \Lambda(t, x)$;

such that the state set Q is the smallest set obtained by application of the above rules. When it is clear from the context, we use ∇ instead of $\nabla_{A,B}$.

If FSM B is distinguishable from A , then any input sequence that, applied at the initial state of ∇ , covers a fail-transition (a transition with the output *fail*), distinguishes B from A , and can be used as a test case to detect faults modeled by B . The distinguishing machine has several

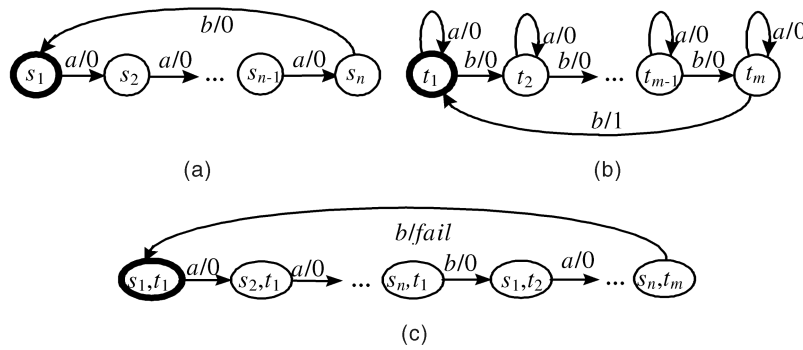


Fig. 3. Two machines distinguished by the input sequence of length nm : (a) the specification FSM, (b) implementation FSM, and (c) distinguishing machine.

other properties useful for test generation; some of them are collected in the following proposition.

Proposition 1. *Given two FSMs, $A = (S, s_0, X, Y, D_A, \delta, \lambda)$ with n states and $B = (T, t_0, X, Y', D_B, \Delta, \Lambda)$ with m states, $\Omega_A \subseteq \Omega_B$, let $\nabla = (Q, q_0, X, Y \cup \{fail\}, D, \psi, \varphi)$ be the distinguishing machine of A and B . Then,*

1. $\Omega_{\nabla}(s, t) = \Omega_A(s)$, for all $s \in S$ and $t \in T$ such that $(s, t) \in Q$.
2. $|Q| \leq nm$.

Proposition 1.1 says that any input sequence defined in ∇ is also defined in the specification machine A . The following statement, which is an immediate corollary to Proposition 1.2, states the upper bound on tests for partial FSMs.

Corollary 1. *For any FSM A with n states, the set $\Omega_A \cap X^{\leq nm}$ (that contains all defined input sequences of length up to nm) is an m -complete test suite.*

Example 2. Consider the specification FSM with n states and implementation FSM with m states shown in Fig. 3. The only fail-transition of DM is at state (s_n, t_m) . The shortest input sequence that covers this transition from the initial state is $(a^{n-1}b)^m$; its length is nm . Therefore, the bound nm is indeed tight for partial machines.

If the set of defined sequences of a specification FSM is finite, i.e., the machine has no cycling behavior, then the set is a complete test suite w.r.t. any fault domain, as stated in the following:

Proposition 2. *Given a specification FSM A such that the set Ω_A of defined input sequences is finite and an arbitrary fault domain $\mathfrak{S}(A)$, the test suite Ω_A is complete w.r.t. $\mathfrak{S}(A)$.*

Therefore, as opposed to complete FSMs, some partial FSMs have the same complete test suite regardless of the bound m on the number of states in implementation machines. A systematic method is required to generate m -complete test suites for an FSM with defined sequences that are longer than nm (e.g., FSMs with cycles). In the remainder of the paper, we present such a method.

3.2 State-Counting Approach

Discussions in the previous section indicate that a complete test suite can be obtained by pruning defined sequences according to the number of states traversed in specification

and implementation machines. The idea is formalized in the so-called state-counting approach to test generation.

3.2.1 Counting States in Test Generation

Let $A = (S, s_0, X, Y, D_A, \delta, \lambda)$ be the specification FSM. As mentioned in Section 3.1, given an implementation FSM $B \in \mathfrak{S}_m(A)$ such that $B \not\sim A$, any input sequence that, applied at the initial state of DM ∇ , covers a fail-transition, distinguishes B from A , and is a test that detects faults modeled by B . It is known that the length of a shortest input sequence that covers a transition of any FSM does not exceed the number of its states. On the other hand, states of distinguishing machines consist of states of FSMs A and B , so any state of A constitutes at most m states of ∇ . Therefore, a transfer sequence from some state to state (s, t) in ∇ is cyclic if it traverses the state s of A more than m times and can be shortened by removing a cycling part, i.e., can be replaced by a shorter, acyclic one that still reaches the state (s, t) . This observation leads us to the following statement.

Lemma 1. *Given a state cover V of FSM A and any state q of ∇ , there exist $\alpha \in V$ and a transfer sequence $\alpha\beta \in \Omega_{\nabla}$ to state q such that β , applied at state $\delta(s_0, \alpha)$, traverses any state of FSM A at most $m - 1$ times.*

We omit the proof of Lemma 1 since it is a particular case of Lemma 3.

The lemma determines the maximal length for input sequences needed to reach any state in an arbitrary $\nabla_{A,B}, B \in \mathfrak{S}_m(A)$. We find it convenient to express the sufficient conditions for an m -complete test suite based on Lemma 1 in terms of posets [28].

Given a sequence β over some alphabet, let $Pref(\beta)$ denote the set of all nonempty prefixes of β . Given states $s, p \in S$ and an input sequence $\beta \in \Omega_A(s)$, state p induces a poset $(Pref_{s,p}(\beta), \leq_{s,p})$, where

$$Pref_{s,p}(\beta) = \{\omega \in Pref(\beta) \mid \delta(s, \omega) = p\}$$

and $\omega \leq_{s,p} \omega', \omega, \omega' \in Pref_{s,p}(\beta)$, if $|\omega| \leq |\omega'|$. Let

$$l(Pref_{s,p}(\beta), \leq_{s,p})$$

be the length of the poset $(Pref_{s,p}(\beta), \leq_{s,p})$. Recall that the length of a poset is the length of the longest chain [28] in the poset, while the length of a chain C is the number

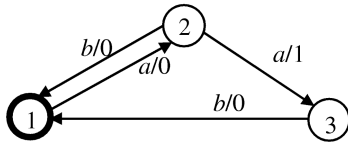


Fig. 4. The unreduced partial FSM A .

$l(C) = |C|$. Note that, in the above case, a poset is a total order, so it is a chain.

The sufficient conditions are stated in the following:

Proposition 3. Let V be a state cover of FSM A and $E = \cup_{\alpha_i \in V} \alpha_i L_i$, where

$L_i = \{ \beta \in \Omega_A(\delta(s_0, \alpha_i)) \mid l(\text{Pref}_{\delta(s_0, \alpha_i), \delta(s_0, \alpha_i \nu)}(\nu), \leq_{\delta(s_0, \alpha_i), \delta(s_0, \alpha_i \nu)}) < m$ for each proper prefix ν of β and $(\Omega_A(\delta(s_0, \alpha_i \beta)) = \emptyset$ or $l(\text{Pref}_{\delta(s_0, \alpha_i), \delta(s_0, \alpha_i \beta)}(\beta), \leq_{\delta(s_0, \alpha_i), \delta(s_0, \alpha_i \beta)}) = m) \}$. The set E is an m -complete test suite of A .

In other words, for each $\alpha_i \in V$, the set L_i is comprised of each shortest sequence $\beta \in \Omega_A(\delta(s_0, \alpha_i))$ such that state $p = \delta(s_0, \alpha_i \beta)$ induces a poset $(\text{Pref}_{\delta(s_0, \alpha_i), p}(\beta), \leq_{(s_0, \alpha_i), p})$ of length m or no input is defined in p .

Proof. Let $B \in \mathfrak{S}_m(A)$ and $B \not\sim A$. If $B \not\sim A$, then, in ∇ , there exists state q with an emanating fail-transition. Let x be the input that labels the fail-transition. By Lemma 1, there exists a transfer sequence $\alpha_i \beta \in \Omega_{\nabla}$ to state q such that $\alpha_i \in V$, $\beta \in \Omega_A(\delta(s_0, \alpha_i))$, and β traverses any state of FSM A including state $p = \delta(s_0, \alpha_i \beta)$ at most $m - 1$ times. Then, the sequence β is a proper prefix of some sequence in the traversal set L_i . Hence, the sequence βx that distinguishes FSMs A and B is a prefix of some sequence in the set L_i . Thus, $B \not\sim_E A$. \square

Proposition 3 gives the method for generating an m -complete test suite from an FSM. It is sufficient to expand each transfer sequence in a chosen state cover (to obtain traversal sets L_i) in all possible ways until, for each obtained sequence, the length of a poset induced by some state reaches m . We illustrate the method using the FSM in Fig. 4.

Example 3. A state cover V of the FSM A in Fig. 4 is $\{\varepsilon, a, aa\}$. Assume $m = 3$. Fig. 5a illustrates the construction of the traversal set L_1 for state 1 and $\alpha_1 = \varepsilon$. The input sequence $aabaaba$ is terminated since, there, state 2 is traversed three times ($m = 3$), in other words, state 2 induces a poset $a \leq_{1,2} aaba \leq_{1,2} aabaaba$ of length three (m). The remaining three input sequences are terminated using the same rule. Fig. 5b illustrates the construction of the traversal set L_2 for state 2 and $\alpha_2 = a$. Here, e.g., the sequence $abaabaa$ is terminated since state 3 induces a poset of length three, while the sequence $abaabab$ is terminated since it is state 1 that induces a poset of length three. The traversal set L_3 for state 3 and $\alpha_3 = aa$ is presented in Fig. 5c. The 3-complete test suite

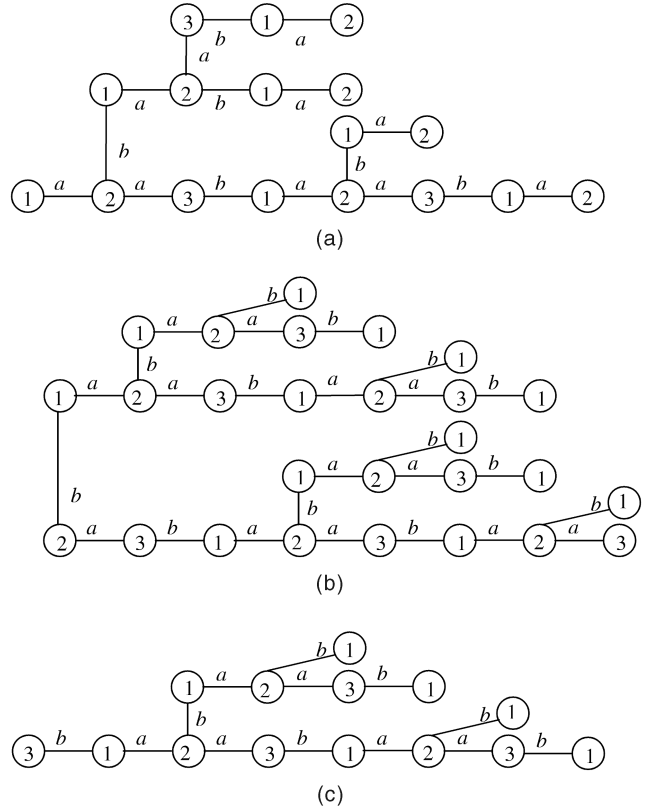


Fig. 5. The traversal sets (a) L_1 , (b) L_2 , and (c) L_3 .

$$E = \cup_{\alpha_i \in V} \alpha_i L_i = \{raabaabaab, raabaabab, raababaab, raababab, rabaabaab, rabaabab, rababaab, rababab\},$$

where r is the reset symbol.² The total length is 68.

It is worth noting that the method does not make use of any relation (quasi-equivalence or distinguishability) between states in the specification FSM. This indicates that, in a general case, the length of sequences in the resulting test suite cannot be further reduced if a specification machine has neither distinguishable nor quasi-equivalent states. The specification FSM in Fig. 3 illustrates this.

3.2.2 Counting Quasi-Equivalent States

Assume now that a given specification machine A has quasi-equivalent states. The presence of quasi-equivalent states may shorten sequences that are needed to cover fail-transitions in distinguishing machines, compared to Lemma 1. The idea is based on the fact that successors of quasi-equivalent states are also quasi-equivalent, as stated in the following:

Lemma 2. If state q' is quasi-equivalent to state q in ∇ and β is a defined input sequence for state q , then state $\psi(q', \beta)$ is quasi-equivalent to state $\psi(q, \beta)$ in ∇ .

Example 4. Consider the FSM A in Fig. 4. By direct inspection, one can verify that state 2 is quasi-equivalent to state 3, $2 \sqsupseteq 3$. Consider also the implementation FSM B , shown in Fig. 6a. The DM $\nabla_{A,B}$ is presented in Fig. 6b. The input sequence aab covers a fail-transition

2. We prefix each string with the reset symbol to account for the number of tests in the test suite.

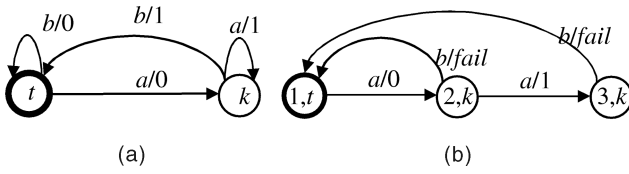


Fig. 6. (a) The implementation FSM B and (b) the distinguishing machine $\nabla_{A,B}$ for FSM A in Fig. 4.

from state $(3, k)$ in ∇ . Since $2 \sqsupseteq 3$, we also have $(2, k) \sqsupseteq (3, k)$, so state $(2, k)$ has a fail-transition as well and a shorter sequence ab also covers a fail-transition in ∇ .

Therefore, given a state q in ∇ with an emanating fail-transition, if a transfer sequence from some state to state q in ∇ traverses states (s', t) and (s, t) such that $s' \sqsupseteq s$, then the transfer sequence can be shortened by removing the part between states (s', t) and (s, t) . The shortened sequence reaches the state that is quasi-equivalent to state q , i.e., the sequence reaches a state that has an emanating fail-transition for the same input. We again use posets to express the sufficient conditions for an m -complete test suite for an FSM with quasi-equivalent states.

Given a specification FSM A , states $s, p \in S$, and input sequence $\beta \in \Omega_A(s)$, state p induces a poset $(Pref_{s,p}(\beta), \sqsupseteq_{s,p})$, where $Pref_{s,p}(\beta) = \{\omega \in Pref(\beta) \mid \delta(s, \omega) \sqsupseteq p\}$ and $\omega \sqsupseteq_{s,p} \omega'$; $\omega, \omega' \in Pref_{s,p}(\beta)$, if $|\omega| \leq |\omega'|$ and $\delta(s, \omega) \sqsupseteq \delta(s, \omega')$. Clearly, $|\beta| \geq l(Pref_{s,p}(\beta), \sqsupseteq_{s,p}) \geq 0$ and $l(Pref_{s,p}(\beta), \sqsupseteq_{s,p}) = 0$ if none of the states traversed by a nonempty sequence β is quasi-equivalent to p ; moreover, $\omega \sqsubseteq_{s,p} \omega'$ implies $\omega \sqsupseteq_{s,p} \omega'$, thus $l(Pref_{s,p}(\beta), \sqsupseteq_{s,p}) \geq l(Pref_{s,p}(\beta), \sqsubseteq_{s,p})$.

We also introduce the concept of a core of an FSM.

Definition 5. Given FSM A , a minimal (with respect to the inclusion ordering) set of states of A that contains the initial state and, for each state $s \in S$, a state quasi-equivalent to s is a core of FSM A .

In the case when the machine A has no quasi-equivalent states, the core coincides with the state set S . A subset $K \subseteq A$ containing the empty sequence is said to be a *core cover* of A if, for each state in the core of A , it has exactly one transfer sequence.

Example 5. Consider the partial FSM A shown in Fig. 4. We have $2 \sqsupseteq 3$. Thus, the core of FSM A contains states 1 and 2. The core cover $K = \{\varepsilon, a\}$.

Using the above notions, Lemma 1 can be generalized to the case of quasi-equivalent states as follows:

Lemma 3. Given a core cover K of FSM A and any state q of ∇ , there exist $\alpha \in K$ and a transfer sequence $\alpha\beta \in \Omega_\nabla$ to state $q' \sqsupseteq q$ such that $l(Pref_{\delta(s_0, \alpha), p}(\beta), \sqsupseteq_{\delta(s_0, \alpha), p}) \leq m - 1$ for all $p \in S$.

Proof. Let P be a set of states that are reachable in ∇ via sequences of the core cover K and β be a shortest transfer sequence from a state of the set P to a state that is quasi-equivalent to state q . In other words, there exists $\alpha \in K$ such that $\psi(q_0, \alpha\beta) \sqsupseteq q$ and, for each sequence $\omega\gamma, \omega \in K$, it holds that $\psi(q_0, \omega\gamma) \sqsupseteq q$ implies $|\gamma| \geq |\beta|$. Since the ∇ is

connected and the set K contains the empty sequence, such a transfer sequence β exists.

Suppose there exists state p of A such that $l(Pref_{\delta(s_0, \alpha), p}(\beta), \sqsupseteq_{\delta(s_0, \alpha), p}) = m$. We have m prefixes β_1, \dots, β_m of β such that $0 < |\beta_1| < \dots < |\beta_m|$ and $\delta(s_0, \alpha\beta_1) \sqsupseteq \dots \sqsupseteq \delta(s_0, \alpha\beta_m) \sqsupseteq p$. All component states of FSM B in

$$(\delta(s_0, \alpha\beta_1), \Delta(t_0, \alpha\beta_1)), \dots, (\delta(s_0, \alpha\beta_m), \Delta(t_0, \alpha\beta_m))$$

are distinct states, denoted t_1, \dots, t_m . Otherwise, if $\Delta(t_0, \alpha\beta_i) = \Delta(t_0, \alpha\beta_j) = t$, the transfer sequence β could be shortened by deleting the part between two states $(\delta(s_0, \alpha\beta_i), \Delta(t_0, \alpha\beta_i))$ and $(\delta(s_0, \alpha\beta_j), \Delta(t_0, \alpha\beta_j))$ of FSM ∇ as $(\delta(s_0, \alpha\beta_i), t) \sqsupseteq (\delta(s_0, \alpha\beta_j), t)$ (Lemma 2). By the definition of a core cover, there exists $\omega \in K$ such that $\delta(s_0, \omega) = s'$ and $s' \sqsupseteq \delta(s_0, \alpha\beta_1) \sqsupseteq \dots \sqsupseteq (\delta(s_0, \alpha\beta_m) \sqsupseteq p$. The sequence ω takes ∇ into state $\psi(q_0, \omega) = (s', \Delta(t_0, \omega))$ such that $\Delta(t_0, \omega) = \Delta(t_0, \alpha\beta_j)$ for some nonempty prefix β_j of β as FSM B has at most m states. As $\delta(s_0, \omega) \sqsupseteq \delta(s_0, \alpha\beta_j)$, $\psi(q_0, \omega) \sqsupseteq \psi(q_0, \alpha\beta_j)$ and the transfer sequence $\alpha\beta'$, where β' is obtained from β by deleting the prefix β_j , takes the FSM ∇ to a state that is quasi-equivalent to state q (Lemma 2). The latter contradicts the fact that the sequence β is a shortest transfer sequence with such feature. Thus, it holds that

$$l(Pref_{\delta(s_0, \alpha), p}(\beta), \sqsupseteq_{\delta(s_0, \alpha), p}) \leq m - 1$$

for all $p \in S$. \square

The lemma leads us to the following statement that includes Proposition 3 as a special case.

Proposition 4. Let K be a core cover of FSM A and $F = \cup_{\alpha_i \in K} \alpha_i M_i$, where

$$M_i = \{\beta \in \Omega_A(\delta(s_0, \alpha_i)) \mid l(Pref_{\delta(s_0, \alpha_i), \delta(s_0, \alpha_i)\nu}(\nu), \sqsupseteq_{\delta(s_0, \alpha_i), \delta(s_0, \alpha_i)\beta}) < m$$

for each proper prefix ν of β and $(\Omega_A(\delta(s_0, \alpha_i)\beta) = \emptyset$ or $l(Pref_{\delta(s_0, \alpha_i), \delta(s_0, \alpha_i)\beta}(\beta), \sqsupseteq_{\delta(s_0, \alpha_i), \delta(s_0, \alpha_i)\beta}) = m)\}$. The set F is an m -complete test suite of A .

In other words, for each $\alpha_i \in V$, the set M_i is comprised of each shortest sequence $\beta \in \Omega_A(\delta(s_0, \alpha_i))$ such that state $p = \delta(s_0, \alpha_i)\beta$ induces a poset $(Pref_{\delta(s_0, \alpha_i), p}(\beta), \sqsupseteq_{\delta(s_0, \alpha_i), p})$ of length m or no input is defined in p .

Proof. Let $B \in \mathfrak{S}_m(A)$ and $B \not\sim A$. If $B \not\sim A$, then, in ∇ , there exists state q with an emanating fail-transition. Let x be the input that labels a fail-transition from state q . By Lemma 3, there exists a transfer sequence $\alpha_i\beta \in \Omega_\nabla$ to state $q' \sqsupseteq q$ such that $\alpha_i \in K$ and $l(Pref_{\delta(s_0, \alpha_i), s}(\beta), \sqsupseteq_{\delta(s_0, \alpha_i), s}) \leq m - 1$ for all $s \in S$. Then, the sequence β is a proper prefix of some sequence in the set M_i . $q' \sqsupseteq q$ implies that $\Omega_\nabla(q') \sqsupseteq \Omega_\nabla(q)$ and x labels a fail-transition from state q' . Hence, the input x is also defined at state q' and sequence βx is a prefix of some sequence in the set M_i that covers a fail-transition. Thus, $B \not\sim_F A$. \square

Example 6. Consider the specification FSM in Fig. 4 with the core cover $\{\varepsilon, a\}$ (see Example 5). Since a test suite E (Proposition 3) is a superset of a test suite F (Proposition 4), we use Fig. 5 to illustrate the method. We notice first of all

that the traversal set in Fig. 5c is irrelevant since state 3 is not in the core. Sequences in the traversal set in Fig. 5a are now reduced as follows: Instead of $aabaaba$, it is sufficient to include into the traversal set M_1 the sequence $aabaa$ since $a \sqsubseteq_{1,3} aaba \sqsubseteq_{1,3} aabaa$, so $l(\text{Pref}_{1,3}(aabaa), \sqsubseteq_{1,3}) = 3$ (notice that there is also another chain of the same length $a \sqsubseteq_{1,3} aa \sqsubseteq_{1,3} aabaa$); the sequence $abaaba$ is shortened to $abaa$ since $a \sqsubseteq_{1,3} aba \sqsubseteq_{1,3} abaa$, i.e.,

$$l(\text{Pref}_{1,3}(abaa), \sqsubseteq_{1,3}) = 3.$$

The other two sequences in L_1 remain the same. Similarly, we determine the traversal set E_2 and, finally, obtain the 3-complete test suite

$$F = \{raabaabaa, raabaabab, raababaa, raababab, rabaabaa, rabaabab, rababaa, rababab\}$$

of total length 64. Recall that, in Example 3, the test suite has 68 symbols.

For the class of FSMs without distinguishable states, Example 2 again indicates that the result of Proposition 4 cannot be improved in a general case. On the other hand, if some states are distinguishable, then a test suite produced by the above method may be further reduced since the method may give a longer test suite compared to state identification-based methods in case of reduced FSMs. This observation points to a test generation approach that includes both state counting and identification approaches as particular cases.

3.3 Test Generation Method

Assume now that some states of the specification FSM A are distinguishable. Such states can be determined, using standard algorithms for state minimization of partial FSMs [12], [15].

Distinguishable states of FSM A may create conflicting states in $\nabla_{A,B}$, $B \in \mathfrak{S}_m(A)$. In particular, states (s, t) and (s', t) of ∇ are said to be *conflicting* if $s \not\sim s'$. Let $s \not\sim s'$, the FSM B is deterministic, so state t cannot agree on the output sequence in response to γ from both states s and s' of FSM A . This implies that a fail-transition can already be covered by the separating sequence γ , applied at (s, t) and (s', t) of ∇ . Intuitively, shortening of tests is achieved as follows: Proposition 4 says that, to obtain an m -complete test suite, it is sufficient to prune each defined sequence to obtain $\alpha_i\beta$ as soon as $l(\text{Pref}_{\delta(s_0, \alpha_i), \delta(s_0, \alpha_i\beta)}(\beta), \sqsubseteq_{\delta(s_0, \alpha_i), \delta(s_0, \alpha_i\beta)})$, the length of a poset induced by the reached state on the set of prefixes of β reaches m . If, however, the sequence β traverses a state s distinguishable from state $\delta(s_0, \alpha_i\beta)$, the latter cannot constitute m states in any $\nabla_{A,B}$, $B \in \mathfrak{S}_m(A)$ since at least one state of B should occur with the state s . Therefore, the bound m has to be lowered to account for distinguishable states. Thus, counting distinguishable states traversed by defined input sequences could allow us to prune them earlier at the price of adding separating sequences (needed to cover fail-transitions).

Lemma 4. Given $B \in \mathfrak{S}_m(A)$, let (s', t) and (s, t) be conflicting states of $\nabla_{A,B}$ such that $s' \not\sim_\gamma s$, then the input sequence γ covers a fail-transition in $\nabla_{A,B}$ from (s', t) or (s, t) .

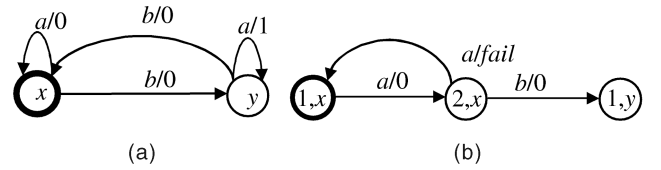


Fig. 7. (a) The implementation FSM B and (b) a fragment of the distinguishing machine $\nabla_{A,B}$ for FSM A in Fig. 4.

Proof. By definition, given state (s, t) of the DM $\nabla_{A,B}$, an input sequence γ applied at state (s, t) covers a fail-transition if and only if $s \not\sim_\gamma t$. As FSM B is deterministic and $s' \not\sim_\gamma s$, the output response of FSM B to γ at state t is different from that of FSM A at state s' or at state s . Thus, $s' \not\sim_\gamma t$ or $s \not\sim_\gamma t$, i.e., the input sequence γ covers a fail-transition in $\nabla_{A,B}$ from (s', t) or (s, t) . \square

Example 7. Consider the specification FSM A in Fig. 4 and the implementation FSM B in Fig. 7a. A fragment of the $\nabla_{A,B}$ is shown in Fig. 7b. Conflicting states are $(1, x)$ and $(2, x)$ since states 1 and 2 are distinguishable by a , $1 \not\sim_a 2$. The latter covers a fail-transition out of state $(2, x)$ in FSM $\nabla_{A,B}$ after the input sequence a of the core cover $\{\varepsilon, a\}$ of A . The FSM B is distinguished from the specification FSM by the input sequence aa .

To generalize Proposition 4 for the case of distinguishable states, we now consider all the posets induced by pairwise distinguishable states traversed by traversal sequences. We first introduce the notion of a set of pairwise distinguishable states.

A set $R \subseteq S$ of states of A is called a *set of pairwise distinguishable states* of A , if $p_1 \neq p_2$ implies $p_1 \not\sim p_2$ for all $p_1, p_2 \in R$; we use R_A to denote the set of all such sets of states of A .

The following proposition states conditions under which traversal sequences are guaranteed either to cover fail-transition for any implementation machine or to traverse conflicting states (from which fail-transition can always be covered, by Lemma 4). Let $C_{s,p}(\beta)$ denote a longest chain of the poset $(\text{Pref}_{s,p}(\beta), \sqsubseteq_{s,p})$.

Proposition 5. Let K be a core cover of FSM A and $G = \cup_{\alpha_i \in K} \alpha_i N_i$, where

$$N_i = \{\beta \in \Omega_A(\delta(s_0, \alpha_i)) \mid \sum_{p \in R} l(\text{Pref}_{\delta(s_0, \alpha_i), p}(\beta), \sqsubseteq_{\delta(s_0, \alpha_i), p}) \leq m$$

for each proper prefix ν of β and for $\forall R \in R_A$ and $(\Omega_A(\delta(s_0, \alpha_i\beta)) = \emptyset$ or $\exists R \in R_A$ s.t. $\delta(s_0, \alpha_i\beta) \in R$ and $\sum_{p \in R} l(\text{Pref}_{\delta(s_0, \alpha_i), p}(\beta), \sqsubseteq_{\delta(s_0, \alpha_i), p}) + |R| = m + 1)$. The following statements hold:

- If FSM A has no distinguishable states, then the set G is an m -complete test suite of A .
- Let $B \in \mathfrak{S}_m(A)$, $B \not\sim A$ and $B \not\sim_G A$. Then, there exist $\alpha_i \in K$, $\beta \in N_i$, such that

$$\sum_{p \in R} l(\text{Pref}_{\delta(s_0, \alpha_i), p}(\beta), \sqsubseteq_{\delta(s_0, \alpha_i), p}) + |R| = m + 1$$

for some set $R \in R_A$, $\delta(s_0, \alpha_i\beta) \in R$. Moreover, for each such set and for each set of the longest chains $C_{\delta(s_0, \alpha_i), p}(\beta)$ of the posets $(\text{Pref}_{\delta(s_0, \alpha_i), p}(\beta), \sqsubseteq_{\delta(s_0, \alpha_i), p})$,

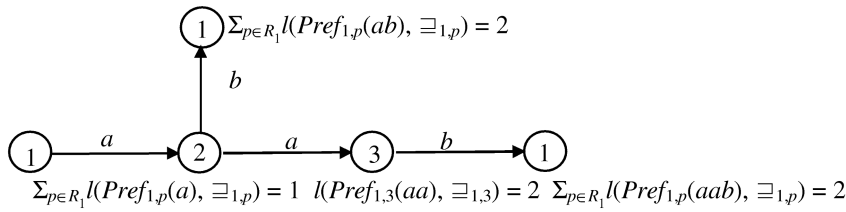


Fig. 8. Constructing the set N_i of traversal sequences.

$p \in R$, the set $\{\psi(q_0, \alpha) \mid (\alpha \in K_i(\beta) \text{ or } \alpha = \alpha_i \gamma, \gamma \in C_{\delta(s_0, \alpha_i), p}(\beta), p \in R)\}$ contains conflicting states, where $K_i(\beta)$ is a subset of K such that, for each state $s \in (R \cup \{\delta(s_0, \alpha_i \gamma) \mid \gamma \in C_{\delta(s_0, \alpha_i), p}(\beta), p \in R\})$, there exists $\alpha \in K_i(\beta)$ such that $\delta(s_0, \alpha) \sqsupseteq s$.

In other words, for each $\alpha_i \in K$, the set N_i is comprised of each shortest sequence $\beta \in \Omega_A(\delta(s_0, \alpha_i))$ such that state $\delta(s_0, \alpha_i \beta)$ has no defined inputs or there exists a set R of pairwise distinguishable states that includes the state $\delta(s_0, \alpha_i \beta)$ and induces a family of posets

$$(Pref_{\delta(s_0, \alpha_i), p}(\beta), \sqsupseteq_{\delta(s_0, \alpha_i), p}),$$

$p \in R$, of total length $m - |R| + 1$. A proof of Proposition 5 is given in the Appendix.

Example 8. Consider the specification FSM A in Fig. 4. The core cover $K = \{\varepsilon, a\}$. As already mentioned, state 2 is quasi-equivalent to state 3, $(1, 2)$ is the only pair of distinguishable states, hence, when pruning defined sequences, it is sufficient to consider two sets of states $R_1 = \{1, 2\}$ and $R_2 = \{3\}$.

The process of determining the set of input sequences N_1 (for $\alpha_1 = \varepsilon$) is illustrated in Fig. 8. A defined sequence β is included in N_1 as soon as either state 3 is reached and $\sum_{p \in R_2} l(\text{Pref}_{1,p}(\beta), \sqsupseteq_{1,p}) = m - |\{3\}| + 1 = 3 - 1 + 1 = 3$ or either state 1 or state 2 is reached and $\sum_{p \in R_1} l(\text{Pref}_{1,p}(\beta), \sqsupseteq_{1,p}) = m - |\{1, 2\}| + 1 = 3 - 2 + 1 = 2$ (as before, $m = 3$). The input sequence a cannot be terminated as state 2 is reached and $l(\text{Pref}_{1,2}(a), \sqsupseteq_{1,2}) = 1$. We expand it by a and b as both are defined in state 2. For input b , we obtain $\sum_{p \in R_1} l(\text{Pref}_{1,p}(ab), \sqsupseteq_{1,p}) = 2$ and terminate the sequence ab . For input a , $l(\text{Pref}_{1,3}(aa), \sqsupseteq_{1,3}) = 2$ (it is because aa first traverses state 2 that is quasi-equivalent to state 3), thus we continue with the input b defined at state 3. This is the last step, as $\sum_{p \in R_1} l(\text{Pref}_{1,p}(aab), \sqsupseteq_{1,p}) = 2$. We obtain $N_1 = \{aab, ab\}$.

Consider now the implementation FSM C in Fig. 9. As can be seen from the fragment of $\nabla_{A,C}$, $C \not\sim_{N_1} A$ holds for machines C and A . However, it can also be seen that the sequence aab traverses two conflicting states $(2, y)$ and $(1, y)$ in $\nabla_{A,C}$, as stated in Proposition 5.

The set G from Proposition 5 can easily be extended to cover fail-transitions missed by G , using Lemma 4. Thus, the presence of distinguishable states in FSM A allows us to decrease the length (and, therefore, the number) of sequences needed to traverse states (cf. Propositions 4 and 5) at a price of including additional input sequences separating states in A .

We are now ready to present the detailed description of the State-Counting (SC-) method.

The SC-method for deriving an m -complete test suite.

Input. FSM $A = (S, s_0, X, Y, D_A, \delta, \lambda)$, all pairs of distinguishable states, and an integer m .

Output. An m -complete test suite for FSM A .

Step 1. Determine a core cover K of A .

Step 2. For each $\alpha_i \in K$, determine the set N_i that comprises each shortest sequence $\beta \in \Omega_A(\delta(s_0, \alpha_i))$ such that $\sum_{p \in R} l(\text{Pref}_{\delta(s_0, \alpha_i), p}(\beta), \sqsupseteq_{\delta(s_0, \alpha_i), p}) + |R| \leq m$ for all $R \in R_A$ and $(\Omega_A(\delta(s_0, \alpha_i \beta)) = \emptyset$ or there exists a set $R \in R_A$ such that $\delta(s_0, \alpha_i \beta) \in R$ and

$$\sum_{p \in R} l(\text{Pref}_{\delta(s_0, \alpha_i), p}(\beta), \sqsupseteq_{\delta(s_0, \alpha_i), p}) + |R| = m + 1).$$

For each sequence β , select one set R of such sets, denoted R_β , and, for each $p \in R_\beta$, determine a longest chain in the poset $(Pref_{\delta(s_0, \alpha_i), p}(\beta), \sqsupseteq_{\delta(s_0, \alpha_i), p})$, denoted $C_{\delta(s_0, \alpha_i), p}(\beta)$. For each R_β , determine a minimal subset $K_i(\beta) \subseteq K$ such that, for each $s \in (R_\beta \cup \{\delta(s_0, \omega) \mid \omega \in \cup_{p \in R_\beta} \alpha_i C_{\delta(s_0, \alpha_i), p}(\beta)\})$, there exists $\alpha \in K_i(\beta)$ such that $\delta(s_0, \alpha) \sqsupseteq s$.

Step 3. For each pair of distinguishable states choose an input sequence γ that separates them. For each $\alpha_i \in K$ and $\beta \in N_i$, determine the following four sets:

$$\begin{aligned} & \{\omega \gamma (\delta(s_0, \omega), \delta(s_0, \mu)) \mid \omega, \mu \in K_i(\beta)\}; \\ & \{\mu \gamma (\delta(s_0, \mu), \delta(s_0, \alpha_i \omega)) \mid \mu \in K_i(\beta), \omega \in \cup_{p \in R_\beta} C_{\delta(s_0, \alpha_i), p}(\beta)\}; \\ & \{\alpha_i \omega \gamma (\delta(s_0, \alpha_i \omega), \delta(s_0, \mu)) \mid \mu \in K_i(\beta), \omega \in \cup_{p \in R_\beta} C_{\delta(s_0, \alpha_i), p}(\beta)\}; \end{aligned}$$

and

$$\{\alpha_i \mu \gamma (\delta(s_0, \alpha_i \mu), \delta(s_0, \alpha_i \omega)) \mid \mu, \omega \in \cup_{p \in R_\beta} C_{\delta(s_0, \alpha_i), p}(\beta)\}.$$

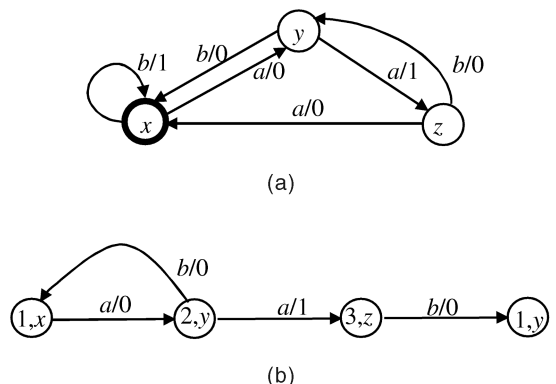


Fig. 9. (a) The implementation FSM C and (b) a fragment of the distinguishing machine.

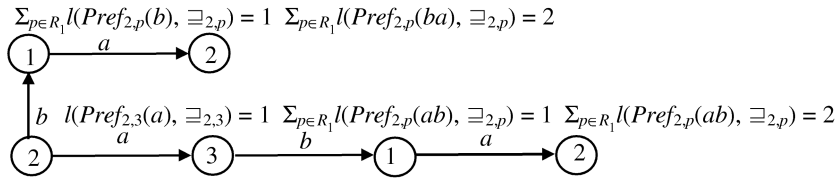


Fig. 10. Constructing the set N_2 of traversal sequences.

The union of the obtained sets over all $\beta \in N_i$ and $\alpha_i \in K$ is an m -complete test suite.

We provide a few comments on the steps of this method. Determining a core of a given FSM in Step 1 is a typical combinatorial, set cover problem, which is finding a minimal number of states that “cover” all the states of a given machine. We say that one state “covers” another state if the former is quasi-equivalent to the latter. Once a core is determined, a core cover can be obtained from a spanning tree similar to the one used to find a state cover, see, e.g., [2]. Step 2 corresponds to constructing the set of traversal sequences in the W -, Wp -, HSI-methods, and other methods for reduced machines [31], [2], [33], [3], [32]. The step can be performed as follows: We start with the empty input sequence and extend it in all the possible ways (obtaining defined sequences) until a current input sequence reaches a state where no input is defined or satisfies the conditions of Proposition 5, namely, the total length of posets induced by some set of pairwise distinguishable states reaches the value $m - |R| + 1$. Step 3 is performed only when the FSM A has distinguishable states. In this step, according to Lemma 4, separating sequences³ are appended to traversal sequences to cover fail-transitions from all conflicting states, identified in Proposition 5. Fail-transitions can be covered from any state reached by input sequences in any of the four types of sets. In the sets $\{\omega\gamma(\delta(s_0, \omega), \delta(s_0, \mu)) \mid \omega, \mu \in K_i(\beta)\}$ we use separating sequences distinguishing states in the core. In the sets

$$\{\mu\gamma(\delta(s_0, \mu), \delta(s_0, \alpha_i\omega)) \mid \mu \in K_i(\beta), \omega \in \cup_{p \in R_\beta} C_{\delta(s_0, \alpha_i), p}(\beta)\}$$

and

$$\{\alpha_i\omega\gamma(\delta(s_0, \alpha_i\omega), \delta(s_0, \mu)) \mid \mu \in K_i(\beta), \omega \in \cup_{p \in R_\beta} C_{\delta(s_0, \alpha_i), p}(\beta)\},$$

we add separating sequences to states in the core as well as to states reached by sequences in all the relevant chains. In the last sets

$$\{\alpha_i\mu\gamma(\delta(s_0, \alpha_\mu), \delta(s_0, \alpha_i\omega)) \mid \mu, \omega \in \cup_{p \in R_\beta} C_{\delta(s_0, \alpha_i), p}(\beta)\},$$

we add separating sequences to all states reached by sequences in all the chains.

Example 9. We complete our working example of the partial FSM A in Fig. 4 by using the SC-method to generate an m -complete test suite for $m = 3$.

Step 1. We determine the core of FSM A with states 1 and 2; a core cover $K = \{\varepsilon, a\}$ (see Example 5).

Step 2. In Example 8, we determined $N_1 = \{aab, ab\}$ for $\alpha_1 = \varepsilon$ (Fig. 8). Similarly, for $\alpha_2 = a$, the traversal set

3. We leave open the question of how these sequences are determined as we believe it is more related to optimization than to completeness of tests.

$N_2 = \{aba, ba\}$ is determined. Fig. 10 illustrates the construction.

We first consider sequences in $N_1 = \{aab, ab\}$, $\alpha_1 = \varepsilon$. For aab , the set $R_{aab} = R_1 = \{1, 2\}$, see Fig. 8, $C_{1,2}(aab) = \{a\}$, while $C_{1,1}(aab) = \{aab\}$.

$$K_1(aab) = K = \{\varepsilon, a\}.$$

For ab , the set $R_{ab} = R_1 = \{1, 2\}$ is also used, see Fig. 7, $C_{1,2}(ab) = \{a\}$, while $C_{1,1}(ab) = \{ab\}$. $K_1(ab) = K = \{\varepsilon, a\}$. Similarly, we proceed with sequences in $N_2 = \{aba, ba\}$, $\alpha_2 = a$. For both sequences, we use

$$R_{aba} = R_{ba} = R_1 = \{1, 2\},$$

see Fig. 10. For aba , the set $C_{2,1}(aba) = \{ab\}$, while $C_{2,2}(aba) = \{aba\}$. $K_2(aba) = K = \{\varepsilon, a\}$. Finally, for ba , the set $C_{2,1}(ba) = \{b\}$, while $C_{2,2}(ba) = \{ba\}$.

$$K_2(ba) = K = \{\varepsilon, a\}.$$

Step 3. States 1 and 2 are the only distinguishable states, they are distinguished by input a , $1 \not\sim_a 2$ (see Example 7). For $\alpha_1 = \varepsilon$ and $\beta = aab$, we have:

$$\gamma(\delta(s_0, \varepsilon), \delta(s_0, a)) = a,$$

$$\{\omega\gamma(\delta(s_0, \omega), \delta(s_0, \mu)) \mid \omega, \mu \in K_i(\beta)\} = \{a, aa\}.$$

$$\{\mu\gamma(\delta(s_0, \mu), \delta(s_0, \alpha_i\omega)) \mid \mu \in K_i(\beta), \omega \in \cup_{p \in R_\beta} C_{\delta(s_0, \alpha_i), p}(\beta)\} = \{a, aa\}.$$

$$\{\alpha_i\omega\gamma(\delta(s_0, \alpha_i\omega), \delta(s_0, \mu)) \mid \mu \in K_i(\beta), \omega \in \cup_{p \in R_\beta} C_{\delta(s_0, \alpha_i), p}(\beta)\} = \{aa, aaba\}.$$

$$\{\alpha_i\omega\gamma(\delta(s_0, \alpha_i\mu), \delta(s_0, \alpha_i\omega)) \mid \mu, \omega \in \cup_{p \in R_\beta} C_{\delta(s_0, \alpha_i), p}(\beta)\} = \{aa, aaba\}.$$

For $\alpha_1 = \varepsilon$ and $\beta = ab$, we obtain

$$\{\omega\gamma(\delta(s_0, \omega), \delta(s_0, \mu)) \mid \omega, \mu \in K_i(\beta)\} = \{a, aa\}.$$

$$\{\mu\gamma(\delta(s_0, \mu), \delta(s_0, \alpha_i\omega)) \mid \mu \in K_i(\beta), \omega \in \cup_{p \in R_\beta} C_{\delta(s_0, \alpha_i), p}(\beta)\} = \{a, aa\}.$$

$$\{\alpha_i\omega\gamma(\delta(s_0, \alpha_i\omega), \delta(s_0, \mu)) \mid \mu \in K_i(\beta), \omega \in \cup_{p \in R_\beta} C_{\delta(s_0, \alpha_i), p}(\beta)\} = \{aa, aba\}.$$

$$\{\alpha_i\omega\gamma(\delta(s_0, \alpha_i\mu), \delta(s_0, \alpha_i\omega)) \mid \mu, \omega \in \cup_{p \in R_\beta} C_{\delta(s_0, \alpha_i), p}(\beta)\} = \{aa, aba\}.$$

We proceed similarly for $\alpha_1 = a$ and $\beta = aba$ and determine the four sets, $\{a, aa\}$, $\{a, aa\}$, $\{aaba, aabaa\}$, and $\{aaba, aabaa\}$. Finally, for $\alpha_1 = a$ and $\beta = ba$, we determine the four sets, $\{a, aa\}$, $\{a, aa\}$, $\{aabaa, abaa\}$, and $\{aabaa, abaa\}$. Merging the determined sets and removing strings that are prefixes of others, we obtain just two words, $aabaa$ and $abaa$. The determined m -complete test suite for $m = 3$ is $\{raabaa, raaaa\}$. The total length is 11 (cf. Examples 3 and 4).

Theorem. *The set of input sequences obtained by the SC-method is an m -complete test suite for a given FSM A .*

The statement follows from Proposition 5 and Lemma 4.

Finally, we discuss the length of a test suite obtained with the proposed method. As Fig. 3 illustrates, there exist pathological FSMs for which the smallest m -complete test suite coincides with the set $\Omega_A \cap X^{\leq nm}$ of all defined sequences of length up to nm . On the other hand, in the class of completely specified reduced FSMs, the length of the resulting test cases is bounded by $n + m - 1$. Actually, for such FSMs, the length of transfer sequences is at most $n - 1$, each traversal sequence is $m - n + 1$ long (see Section 4 for more detail) and each separating sequence does not exceed $n - 1$, so the length is $n - 1 + m - n + 1 + n - 1 = n + m - 1$. This is the tight bound to distinguish complete FSM with n states from complete FSM with m states [17].

The total length of an m -complete test suite for the specification FSM with distinguishable states generated by the SC-method depends on the choice of separating sequences, constituting a characterization set W , since these sequences are a part of the input to our method. The tight upper bound for the length of such sequences for partial FSMs is known, but it is unknown whether a shorter sequence always leads to shorter tests. “Optimal” separating sequences should yield a test suite of minimal complexity. The problem of determining such sequences remains open for all existing test generation methods with fault coverage, our method included. Treating separating sequences as the input to our method, we separate this combinatorial problem from the test completeness problem for arbitrary deterministic machines, which we solve with the SC-method. It seems that choosing separating sequences after all the traversal sequences have been determined [13] might be a better strategy than fixing them in advance, but more research is needed to solve the optimization problem.

4 RELATED AND FUTURE WORK

We are not aware of any other method that generates an m -complete test suite from a partially specified unreduced FSM. To the best of our knowledge, all the existing methods require the FSM be reduced since checking experiments are only defined for such machines—the whole idea of state identification comes from the assumption that all states are distinguishable. As opposed to state identification-based testing methods, the SC-method applies to any deterministic FSM, even if its states are not distinguishable. At the same time, the proposed method is also applicable to

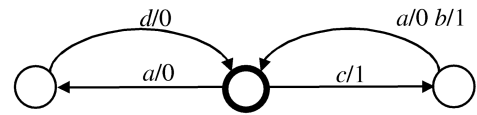


Fig. 11. Unreduced FSM C .

reduced machines, so it can be compared to the existing methods. Informally speaking, the more distinguishable states the specification FSM has, the more the structure of a complete test suite returned by our method resembles that for reduced machines (assuming reset operation). To demonstrate the point more formally, we consider the set G of Proposition 5 for the case of a completely specified reduced specification FSM A with n states.

The core cover K of such a machine coincides with a state cover V of the given machine. All n states of A are pairwise distinguishable. Then, each set N_i is nothing more than the set of all traversal input sequences of length up to $m - n + 1$, i.e., $N_i = X^{\leq m-n+1}$. Thus, $G = VX^{\leq m-n+1}$. Let W denote a characterization set of FSM A , i.e., a set of input sequences of A that, for any two states, has a separating sequence. According to Lemma 4, the sequences of W , concatenated to each sequence in the set G , cover a fail-transition in the distinguishing machine for any implementation FSM that is not distinguished from A by sequences of the set G . If V is prefix-closed while W is suffix-closed, then the set $VX^{\leq m-n+1}W$ is exactly the one generated by the W -method [31], [2], which is m -complete. The latter needs the completely specified FSM to be first reduced in case it has equivalent states. Notice that the SC-method directly treats such FSMs and delivers the same test suite without minimizing it. A similar observation applies to the test generation methods developed for reduced partial FSMs in [33] and [32]. This indicates that the SC-method includes the existing methods such as [31], [2], [33], [32] as special cases. Similarly to those methods, the proposed method does not guarantee that the resulting test suite is the shortest possible; in other words, the problem of finding a complete test suite of a minimal total length remains open and requires further research. Here, we discuss some possibilities for further improvement of the SC-method.

When a specification FSM has no distinguishable states, it might be useful to count not only states of the specification traversed by an input sequence, but also the traversed transitions in order to terminate input sequences earlier.

Example 10. We illustrate the above point with the specification FSM C in Fig. 11. If each implementation FSM has a single state ($m = 1$) and $\{\varepsilon, a, c\}$ is a state cover, then Proposition 3 yields the test suite $\{rad, rca, rcb\}$.

However, to verify whether an implementation FSM with only one state is distinguishable from the specification FSM, it is sufficient to apply each input at this state and check the response of the implementation machine. Therefore, input a in the test rca is redundant since an implementation can only produce the same output as in the test rad ; thus, the test suite $\{rad, rcb\}$ is also 1-complete. It is shorter by three symbols than the previous test suite.

Finally, we notice that the use of a prefix-closed state cover generally facilitates merging sequences in sets $\alpha_i L_i$ (Proposition 3), shortening a test suite. However, for the machine in Fig. 11, the state cover $\{\varepsilon, a, adc\}$ that is not prefix-closed and longer than $\{\varepsilon, a, c\}$ leads to the 1-complete test suite $\{radcb\}$ that is shorter than the above considered test suites. More research is needed to investigate how these observations can be used to improve the SC-method.

There are other research directions that could be pursued based on this work. One can try, for example, to construct a single checking sequence from the resulting complete test suite for a partial machine, similar to what was done in [25] for complete machines (without reset). Our current work is to generalize the SC-method to nondeterministic FSMs, improving earlier methods [21], [19].

5 CONCLUSION

We addressed in this paper the problem of test generation with complete fault coverage from partially specified deterministic FSMs and proposed the so-called state-counting approach. This approach could be viewed as a generalization of the classical state identification approach for test generation to unreduced partially specified deterministic FSMs. We demonstrated that, applied to reduced deterministic machines, the resulting tests coincide with that obtained by known methods. The tight upper bound for the length of test cases was established for unreduced machines.

Since the proposed SC-method does not guarantee that the resulting test suite is the shortest possible, we discussed several possibilities for further improvements of the method and indicated other possible future work.

APPENDIX

PROOF OF PROPOSITION 5

1. If FSM A has no distinguishable states, then the set G coincides with the set F in Proposition 4, i.e., G is an m -complete test suite.
2. We consider an FSM B that is distinguishable from A , but $B \sim_G A$. We have to demonstrate that the set of states in the FSM ∇ traversed by an input sequence β from some N_i and the input sequences of the set $K_i(\beta)$ contains two conflicting states. Moreover, in these two conflicting states, states of FSM A are quasi-equivalent to states of the set R that bounds the input sequence β , in other words, $\sum_{p \in R} l(\text{Pref}_{\delta(s_0, \alpha_i), p}(\beta), \exists_{\delta(s_0, \alpha_i), p}) = m - |R| + 1$.

Let $B \not\sim A$ and P be a set of states that are reachable in ∇ via sequences of the core cover K and ν be a shortest transfer sequence from a state of the set P to a state q with a fail-transition. In other words, there exists $\alpha \in K$ such that $\psi(q_0, \alpha\nu) = q$ and, for each sequence $\omega\gamma$, $\omega \in K$, such that $\psi(q_0, \omega\gamma)$ has a fail-transition, it holds that $|\gamma| \geq |\nu|$. Since $B \not\sim A$, the ∇ is connected, and the set K contains the empty sequence, such a transfer sequence ν exists. By definition of sets N_i and as $B \sim_G A$, the set G has a prefix β of ν with the property $\sum_{p \in R} l(\text{Pref}_{\delta(s_0, \alpha), p}(\beta), \exists_{\delta(s_0, \alpha), p}) = m - |R| + 1$ for some set R of pairwise distinguishable states of A such that $\delta(s_0, \alpha\beta) \in R$.

Consider the union $\{\beta_1, \dots, \beta_{m-|R|+1} = \beta\}$ of longest chains $C_{\delta(s_0, \alpha), p}(\beta)$ of the posets $(\text{Pref}_{\delta(s_0, \alpha), p}(\beta), \exists_{\delta(s_0, \alpha), p})$, $p \in R$, where $1 \leq |\beta_1| < \dots < |\beta_{m-|R|+1}|$ and the corresponding $m - |R| + 1$ states of FSM B ,

$$\Delta(t_0, \alpha\beta_1), \dots, \Delta(t_0, \alpha\beta_{m-|R|+1}).$$

By definition, for any $j, k = 1, \dots, m - |R| + 1$, either there exists $p \in R$ such that $\delta(s, \alpha\beta_j) \supseteq \delta(s, \alpha\beta_k) \supseteq p$ or there exist distinguishable states $p_1, p_2 \in R$ such that $\delta(s, \alpha\beta_j) \supseteq p_1$ and $\delta(s, \alpha\beta_k) \supseteq p_2$. Thus, if there exist $j < k$, such that $\Delta(t_0, \alpha\beta_j) = \Delta(t_0, \alpha\beta_k)$, then $\delta(s_0, \alpha\beta_j) \not\sim (s_0, \alpha\beta_k)$. The reason is that if $\delta(s, \alpha\beta_j) \supseteq \delta(s, \alpha\beta_k)$ and $\beta_k = \beta_j\rho$, then a shortest transfer sequence v could be further shortened by deleting the part ρ that takes the FSM ∇ from state $(\delta(s_0, \alpha\beta_j), \Delta(t_0, \alpha\beta_j))$ to state $(\delta(s_0, \alpha\beta_k), \Delta(t_0, \alpha\beta_k))$ (Lemma 2). Thus, if $m - |R| + 1$ states of FSM B , $\Delta(t_0, \alpha\beta_1), \dots, \Delta(t_0, \alpha\beta_{m-|R|+1})$, are not distinct, then the FSM ∇ has two conflicting states $(\delta(s_0, \alpha\beta_j), \Delta(t_0, \alpha\beta_j))$, and $(\delta(s_0, \alpha\beta_k), \Delta(t_0, \alpha\beta_k))$.

Assume, therefore, that the states

$$\Delta(t_0, \alpha\beta_1), \dots, \Delta(t_0, \alpha\beta_{m-|R|+1})$$

are distinct. Let $K(\beta)$ be a subset of the core cover K of A that has exactly $|R|$ transfer sequences such that, for any state s of the set $\{\delta(t_0, \alpha\beta_1), \dots, \delta(t_0, \alpha\beta_{m-|R|+1})\} \cup R$, there exists $\mu \in K(\beta)$ for which $\delta(s_0, \mu) \supseteq s$ holds. The corresponding $|R|$ states of FSM B are distinct if the set $K(\beta)$ does not traverse conflicting states. Then, the two subsets of states of FSM B , $\{\Delta(t_0, \alpha\beta_1), \dots, \Delta(t_0, \alpha\beta_{m-|R|+1})\}$ and $\{\Delta(t_0, \gamma) \mid \gamma \in K(\beta)\}$, intersect as, together, they have $m - |R| + 1 + |R| = m + 1$ states, while B has only m states. Thus, there exist a sequence $\mu \in K(\beta)$ and nonempty prefix β_j of β such that $\Delta(t_0, \mu) = \Delta(t_0, \alpha\beta_j)$, while $\delta(s_0, \mu)$ and $\delta(s_0, \alpha\beta_j)$ are quasi-equivalent to some states in the set R . If states $\delta(s_0, \mu)$ and $\delta(s_0, \alpha\beta_j)$ are quasi-equivalent to the same state in the set R , then $\delta(s_0, \mu) \supseteq \delta(s_0, \alpha\beta_j)$ by definition of the set $K(\beta)$. In this case, the sequence $\mu\nu'$, where ν' is obtained from ν by deleting the prefix β_j , takes the FSM ∇ to a state that is quasi-equivalent to state q (Lemma 2), i.e., to a state with a fail-transition. The latter contradicts the fact that it is a shortest sequence with such a property. Thus, $\delta(s_0, \mu)$ and $\delta(s_0, \alpha\beta_j)$ are quasi-equivalent to two different states of the set R , i.e., $\delta(s_0, \mu) \not\sim \delta(s_0, \alpha\beta_j)$. Hence, the set of states $\{\psi(q_0, \omega) \mid \omega \in K(\beta) \text{ or } \omega = \alpha\gamma, \gamma \in C_{\delta(s_0, \alpha), p}(\beta), p \in R\}$ contains conflicting states. \square

ACKNOWLEDGMENTS

This work was in part supported by the Natural Sciences and Engineering Research Council of Canada under discovery grant OGP0194381 and the Ministry of High Education of Russia.

REFERENCES

- [1] G. v. Bochmann and A. Petrenko, "Protocol Testing: Review of Methods and Relevance for Software Testing," *Proc. ACM Int'l Symp. Software Testing and Analysis (ISSTA '94)* pp. 109-124, 1994.
- [2] T.S. Chow, "Test Software Design Modeled by Finite State Machines," *IEEE Trans. Software Eng.*, vol. 4, no. 3, pp. 178-187, 1978.

- [3] S. Fujiwara, G. v. Bochmann, F. Khendek, M. Amalou, and A. Ghedamsi, "Test Selection Based on Finite State Models," *IEEE Trans. Software Eng.*, vol. 17, no. 6, pp. 591-603, June 1991.
- [4] S. Goren and F.J. Ferguson, "CHESMIN: A Heuristic for State Reduction in Incompletely Specified Finite State Machines," *Proc. 2002 Design, Automation, and Test in Europe Conf. and Exhibition*, 2002.
- [5] A. Gill, *Introduction to the Theory of Finite-State Machines*. New York: McGraw-Hill, 1962.
- [6] G. Gonenc, "A Method for the Design of Fault Detection Experiments," *IEEE Trans. Computers*, vol. 19, no. 6, pp. 551-558, June 1970.
- [7] F.C. Hennie, "Fault Detecting Experiments for Sequential Circuits," *Proc. IEEE Fifth Ann. Symp. Switching Circuit Theory and Logical Design*, pp. 95-110, 1964.
- [8] R.M. Hierons and H. Ural, "Reduced Length Checking Sequences," *IEEE Trans. Computers*, vol. 51, no. 9, pp. 1111-1117, Sept. 2002.
- [9] E.P. Hsieh, "Checking Experiments for Sequential Machines," *IEEE Trans. Computers*, vol. 20, no. 10, pp. 1152-1166, Oct. 1971.
- [10] J. Kella, "Sequential Machine Identification," *IEEE Trans. Computers*, vol. 20, no. 3, pp. 332-338, Mar. 1971.
- [11] J. Kim and M.M. Newborn, "The Simplification of Sequential Machines with Input Restrictions," *IEEE Trans. Computers*, vol. 21, no. 12, pp. 1440-1443, Dec. 1972.
- [12] Z. Kohavi, *Switching and Finite Automata Theory*. New York: McGraw-Hill, 1970.
- [13] I. Koufareva and M. Dorofeeva, "A Novel Modification of W-Method," *Joint Bull. Novosibirsk Computing Center and A.P. Ershov Inst. of Informatics Systems, Series: Computing Science*, no. 18, pp. 69-81, NCC Publisher, Novosibirsk, 2002.
- [14] R. Lai, "A Survey of Communication Protocol Testing," *J. Systems and Software*, vol. 62, pp. 21-46, 2002.
- [15] D. Lee and M. Yannakakis, "Testing Finite-State Machines: State Identification and Verification," *IEEE Trans. Computers*, vol. 43, no. 3, pp. 306-320, Mar. 1994.
- [16] D. Lee and M. Yannakakis, "Principles and Methods of Testing Finite State Machines, a Survey," *Proc. IEEE*, vol. 84, no. 8, pp. 1090-1123, 1996.
- [17] E. Moore, "Gedanken—Experiments on Sequential Machines," *Automata Studies*, Princeton, N.J.: Princeton Univ. Press, pp. 129-153, 1956.
- [18] A. Petrenko, "Fault Model-Driven Test Derivation from Finite State Models: Annotated Bibliography," *Proc. Modeling and Verification of Parallel Processes (MOVEP 2000)*, pp. 196-205, 2000.
- [19] A. Petrenko, N. Yevtushenko, and G. v. Bochmann, "Testing Deterministic Implementations from Their Nondeterministic Specifications," *Proc. IFIP Ninth Int'l Workshop Testing of Communicating Systems*, pp. 125-140, 1996.
- [20] A. Petrenko, N. Yevtushenko, and R. Dssouli, "Testing Strategies for Communicating FSMs," *Proc. IFIP Seventh Int'l Workshop Protocol Test Systems*, pp. 193-208, 1994.
- [21] A. Petrenko, N. Yevtushenko, A. Lebedev, and A. Das, "Nondeterministic State Machines in Protocol Conformance Testing," *Proc. IFIP Sixth Int'l Workshop Protocol Test Systems*, pp. 363-378, 1993.
- [22] A. Petrenko and N. Yevtushenko, "On Test Derivation from Partial Specifications," *Proc. IFIP Joint Int'l Conf. Formal Description Techniques for Distributed Systems and Comm. Protocols and Protocol Specification, Testing, and Verification (FORTE/PSTV 2000)* pp. 85-102, 2000.
- [23] J.F. Poage and E.J. McCluskey, "Derivation of Optimum Test Sequences for Sequential Machines," *Proc. Fifth Ann. Symp. Switching Theory and Logical Design*, pp. 121-132, 1964.
- [24] I. Pomeranz and S.M. Reddy, "Test Generation for Multiple State-Table Faults in Finite-State Machines," *IEEE Trans. Computers*, vol. 46, no. 7, pp. 783-794, July 1997.
- [25] A. Rezaki and H. Ural, "Construction of Checking Sequences Based on Characterization Sets," *Computer Comm.*, vol. 18, pp. 911-920, Dec. 1995.
- [26] J.-K. Rho, G. Hachtel, and F. Somentzi, "Don't Care Sequences and the Optimization of Interacting Finite State Machines," *Proc. IEEE Conf. Computer-Aided Design*, pp. 414-421, 1991.
- [27] R. Shehady and D.P. Siewiorek, "A Method to Automate User Interface Testing Using Variable Finite State Machines," *Proc. 27th Int'l Symp. Fault Tolerant Computing* pp. 80-88, 1997.
- [28] B.S.W. Schröder, *Ordered Sets: An Introduction*. Boston: Birkhäuser, 2003.
- [29] D.P. Sidhu and T.K. Leung, "Formal Methods for Protocol Testing: A Detailed Study," *IEEE Trans. Software Eng.*, vol. 15, no. 4, pp. 413-426, Apr. 1989.
- [30] H. Ural, "Formal Methods for Test Sequence Generation," *Computer Comm.*, vol. 15, no. 5, pp. 311-325, 1992.
- [31] M.P. Vasilevskii, "Failure Diagnosis of Automata," *Cybernetics*, no. 4, pp. 653-665, 1973.
- [32] M. Yannakakis and D. Lee, "Testing Finite State Machines: Fault Detection," *J. Computer and System Sciences*, vol. 50, pp. 209-227, 1995.
- [33] N. Yevtushenko and A. Petrenko, "Synthesis of Test Experiments in Some Classes of Automata," *Automatic Control and Computer Sciences*, no. 4, pp. 50-55, 1990.
- [34] N. Yevtushenko and A. Petrenko, "Test Derivation Method for an Arbitrary Deterministic Automaton," *Automatic Control and Computer Sciences*, no. 5, pp. 65-68, 1990.



Alexandre Petrenko received the diploma degree in electrical and computer engineering from Riga Polytechnic Institute in 1970 and the PhD degree in computer science from the Institute of Electronics and Computer Science, Riga, USSR, in 1974. He was also awarded other degrees and titles, namely, "Doctor of Technical Sciences" and "Senior Research Fellow in Technical Cybernetics and Information Theory" from the Supreme Attestation Committee, Moscow, USSR, and "Doctor Habil. of Computer Science" from the Latvian Scientific Council, Riga, Latvia. Until 1992, he was head of a computer network research lab of the Institute of Electronics and Computer Science in Riga. From 1979 to 1982, he was with the Computer Network Task Force of the International Institute for Applied Systems Analysis (IIASA), Vienna, Austria. From 1992 to 1996, he was a visiting professor/researcher of the Université de Montréal. He joined the Centre de Recherche Informatique de Montréal (CRIM) in 1996, where he is currently a senior researcher and team leader. In 2005, along with C. Campbell, M. Veanes, and J. Huo, he received the best paper award from the 17th IFIP International Conference on Testing of Communicating Systems. He has published more than 150 research papers and has given numerous invited lectures worldwide. He is a member of the IFIP TC6 Working Group 6.1 "Architectures and Protocols for Distributed Systems" and serves as a member of the program committee for a number of international conferences and workshops. He is a member of the steering committee of the IFIP International Conference on Testing of Communicating Systems (TestCom). His current research interests include formal methods and their application in distributed systems and computer networks.



Nina Yevtushenko received the diploma degree in electrical engineering from Tomsk State University in 1971 and the PhD degree in computer science from Saratov State University in 1983. She received the "Doctor of Technical Sciences" degree and a professorship title from the Supreme Attestation Committee in Moscow. From 1971 to 1991, she worked as a researcher with the Siberian Scientific Institute of Physics and Technology. In 1991, she joined Tomsk State University as a professor, where she currently leads a research team working on the synthesis and analysis of discrete event systems. She was also a visiting researcher/professor at Moscow State University, the Université de Montréal, Centre de Recherche Informatique de Montréal, University of Ottawa, and Institut National des Telecommunications in Evry, France. She has published approximately 100 research papers. Her research interests include formal methods, automata theory, distributed systems, protocol, and discrete event systems testing. She has received two NATO linkage grants in FSM analysis and synthesis along with the scientific team of Professor R. Brayton (University of California at Berkeley).