

Bounded Sequence Testing from Non-deterministic Finite State Machines

Florentin Ipate

Department of Computer Science and Mathematics
University of Pitesti, Romania
fipate@ifsoft.ro

Abstract. The widespread use of *finite state machines (FSMs)* in modeling of communication protocols has led to much interest in testing from (deterministic and non-deterministic) FSMs. Most approaches for selecting a test suite from a non-deterministic FSM are based on *state counting*. Generally, the existing methods of testing from FSMs check that the implementation under test behaves as specified for *all* input sequences. On the other hand, in many applications, only input sequences of limited length are used. In such cases, the test suite needs only to establish that the IUT produces the specified results in response to input sequences whose length does not exceed an upper bound l . A recent paper devises methods for *bounded sequence testing* from *deterministic* FSM specifications. This paper considers the, more general, situation where the specification may be a *non-deterministic FSM* and extends state counting to the case of bounded sequences. The extension is not trivial and has practical value since the test suite produced may contain only a small fraction of all sequences of length less than or equal to the upper bound.

1 Introduction

Finite state machines (FSMs) are widely used in modeling of communication protocols. As testing is a vital part of system development, this has led to much interest in testing from FSMs [13], [9]. Given a FSM specification, for which we have its transition diagram, and an implementation, which is a “black box” for which we can only observe its input/output behavior, we want to test whether the implementation under test (IUT) conforms to the specification. This is called *conformance testing* or *fault detection* and a set of sequences that solves this problem is called a *test suite*.

Many test selection methods have been developed for the case where the specification is a *deterministic* FSM. The best known methods are: Transition Tour [13], Unique Input Output (UIO) [13], Distinguishing Sequence [13], the W method [2], [13] and its variant, the “partial W ” (Wp) method [3]. The W and Wp methods will find all the faults in the IUT provided that the number of states of the IUT remain below a known upper bound.

When the specification is deterministic, equivalence is the natural notion of correctness. On the other hand, when the specification is non-deterministic,

equivalence may often be too restrictive. Usually, a non-deterministic FSM specification provides a set of alternative output sequences that are valid responses to some input sequence and the IUT may choose from these (when the IUT is deterministic only one choice is allowed, otherwise multiple choices can be made). Consequently, the IUT is correct if and only if every input/output sequence that is possible in the IUT is also present in the specification; we say that the IUT is a *reduction* of the specification. Obviously, equivalence is a particular case of reduction, where all specified choices are implemented. Most approaches for selecting a test suite from a non-deterministic FSM are based on *state counting* [11], [12], [14].

Generally, the existing methods of testing from FSMs check that the IUT behaves as specified for *all* input sequences. On the other hand, in many applications, only input sequences of limited length are used. In such cases, the test suite needs only to establish that the IUT produces the specified results in response to input sequences whose length does not exceed an upper bound l . A recent paper extends the *W* and *Wp* methods to the case of *bounded sequences* [8].

This paper considers the, more general, situation where the specification may be a *non-deterministic FSMs* and extends the state counting based test selection method to the case of bounded sequences. The extension is not straightforward since it is not sufficient to extract the prefixes of length at most l from the test suite produced in the unbounded case. Furthermore, the test suite produced may contain only a small fraction of all sequences of length less than or equal to the upper bound.

The paper is structured as follows. Section 2 introduces FSM related concepts and results that are used later in the paper, while section 3 reviews the use of state counting in testing from non-deterministic FSMs. Section 4 presents the testing method for bounded sequences, while the following two sections provide its theoretical basis: the l -bounded product FSM is defined in section 5, while in section 6, state counting is used to validate the test suite given earlier. Conclusions are drawn in section 7.

2 Finite state machines

This section introduces the finite state machine and related concepts and results that will be used later in the paper.

First, the notation used is introduced. For a finite set A , we use A^* to denote the set of finite sequences with members in A ; ϵ denotes the empty sequence. For $a, b \in A^*$, ab denotes the concatenation of sequences a and b . a^n is defined by $a^0 = \epsilon$ and $a^n = a^{n-1}a$ for $n \geq 1$. For $U, V \subseteq A^*$, $UV = \{ab \mid a \in U, b \in V\}$; U^n is defined by $U^0 = \{\epsilon\}$ and $U^n = U^{n-1}U$ for $n \geq 1$. Also, $U[n] = \bigcup_{0 \leq k \leq n} U^k$. For a sequence $a \in A^*$, $\|a\|$ denotes the length (number of elements) of a ; in particular $\|\epsilon\| = 0$. For a sequence $a \in A^*$, $b \in A^*$ is said to be a *prefix* of a if there exists a sequence $c \in A^*$ such that $a = bc$. The set of all prefixes of a is

denoted by $\text{pref}(a)$. For $U \subseteq A^*$, $\text{pref}(U)$ denotes the set of all prefixes of the elements in U . For a finite set A , $|A|$ denotes the number of elements of A .

A *finite state machine (FSM)* M is a tuple $(\Sigma, \Gamma, Q, h, q_0)$, where Σ is the finite *input alphabet*, Γ is the finite *output alphabet*, Q is the finite *set of states*, $h : Q \times \Sigma \rightarrow 2^{Q \times \Gamma}$ is the *transition function* and $q_0 \in Q$ is the *initial state*. A FSM is usually described by a state-transition diagram. Given $q, q' \in Q$, $\sigma \in \Sigma$ and $\gamma \in \Gamma$, the application of input σ when M is in state q may result in M moving to state q' and outputting γ if and only if $(q', \gamma) \in h(q, \sigma)$.

M is said to be *completely specified* if for all $q \in Q$ and $\sigma \in \Sigma$, $|h(q, \sigma)| \geq 1$. If M is not completely specified, it may be transformed to form a completely specified FSM by assuming that the “refused” inputs produce a designated error output, which is not in the output alphabet of M ; this behavior can be represented as self-looping transitions or transitions to an extra (error) state. M is said to be *deterministic* if for all $q \in Q$ and $\sigma \in \Sigma$, $|h(q, \sigma)| \leq 1$.

The function h may be extended to take input sequences and produce output sequences, i.e. $h : Q \times \Sigma^* \rightarrow 2^{Q \times \Gamma^*}$. The projections $h_1 : Q \times \Sigma^* \rightarrow 2^Q$ and $h_2 : Q \times \Sigma^* \rightarrow 2^{\Gamma^*}$ of h give the states reached (h_1) and the output sequences produced (h_2) from a state, given an input.

A FSM M is said to be *initially connected* if every state q is reachable from the initial state of M , i.e. there exists $s \in \Sigma^*$ such that $q \in h_1(q_0, s)$. If M is not initially connected it may be transformed into an initially connected FSM by removing the unreachable states.

Given a state q , the associated language of q , $L_M(q)$, contains the input/output sequences allowed by M from q . More formally, $L_M(q) = \{(s, g) \mid s \in \Sigma^*, g \in h_2(q, s)\}$. The input/output sequences allowed by M from q_0 make up the associated language of M , denoted by $L(M)$.

States q of M and q' of M' are said to be *equivalent* if $L_M(q) = L_{M'}(q')$. FSMs M and M' are said to be *equivalent* if their initial states are equivalent, i.e. $L(M) = L(M')$. The equivalence relation can be restricted to a set of input sequences $Y \subseteq \Sigma^*$; this is called Y -equivalence.

M is said to be *observable* if for every state q , input σ and output γ , M has at most one transition leaving q with input σ and output γ , i.e. $|\{q' \mid (q', \gamma) \in h(q, \sigma)\}| \leq 1$. In such a FSM, given $q \in Q$, $s \in \Sigma^*$ and $g \in \Gamma^*$, $h_g(q, s)$ is used to denote the state (if exists) where input sequence s takes M from state q while outputting sequence g . Every FSM is equivalent to an observable FSM [10]. It will thus be assumed that any FSM considered is observable.

Suppose M and M' are two completely specified FSMs. Given states q of M and q' of M' , q is said to be a *reduction* of q' , written $q \leq q'$, if $L_M(q) \subseteq L_{M'}(q')$. Obviously, q and q' are equivalent if and only if $q \leq q'$ and $q' \leq q$. On the class of deterministic FSMs, the two relations coincide. The FSM M is said to be a *reduction* of the FSM M' , written $M \leq M'$, if $q_0 \leq q'_0$. Given a set of input sequences $Y \subseteq \Sigma^*$, weaker reduction relations, denoted by \leq_Y , can be obtained by restricting the above definitions to Y .

3 Testing from non-deterministic FSMs

This section briefly reviews the use of state counting in testing from (possibly) non-deterministic FSMs [5]. One important case is where the IUT is known to be deterministic [12]. However, the general case where the IUT may also be non-deterministic, is considered. All FSMs referred to are assumed to be initially connected, completely specified and observable.

3.1 Prerequisites

When testing from a formal specification, it is usual to assume that the IUT behaves like some unknown element from a *fault domain*. In the case of a FSM specification $M = (\Sigma, \Gamma, Q, h, q_0)$, the fault domain consists of all initially connected, completely specified and observable FSMs $M' = (\Sigma, \Gamma, Q', h', q'_0)$ with the same input and output alphabets as M and at most m' states, where m' is a predetermined integer greater than or equal to the number m of states of M . Furthermore, it will be assumed that the IUT has a reliable reset. A FSM has a reset operation if there is some input r that takes every state to the initial state. A reliable reset is a reset that is known to have been implemented correctly and might be implemented through the system being switched off and then on again. The reset will not be included in the input alphabet.

A *test suite* is a finite set of input sequences that, for every M' in the fault model that is not a reduction of M , shows that M' is erroneous. More formally, $Y \subseteq \Sigma^*$ is a test suite if and only if for every M' in the fault model, $M' \leq M$ if and only if $M' \leq_Y M$. Naturally, when the specification M is deterministic, testing for $M' \leq M$ reduces to testing for the equivalence of the two FSMs

When testing a non-deterministic implementation, it is normal to make a fairness assumption, called the *complete testing assumption* [10], that there is some known N such that if an input sequence is applied N times then every possible response is observed at least once. Naturally, this assumption automatically holds when the implementation is deterministic. This paper will assume that the complete testing assumption can be made.

When testing from a FSM M , sequences that *reach* and *distinguish* the states of M are normally selected. These issues are now discussed.

3.2 Reaching states

Input sequence $s \in \Sigma^*$ is said to *deterministically-reach* (*d-reach*) state q if $h_1(q_0, s) = \{q\}$. That is, q is the only state reached by s . q is said to be *d-reachable* [12]. The initial state is always d-reachable since it is d-reached by the empty sequence ϵ . Naturally, all reachable states of a deterministic FSM are also d-reachable.

A set $S \subseteq \Sigma^*$ of input sequences is called a *state cover* of M if $\epsilon \in S$ and S is a minimal set such that every d-reachable state of M is d-reached by some sequence from S .

Consider, for example, M as represented in Fig. 1. States 0, 1 and 2 are d-reached by ϵ , a and aa , respectively. On the other hand, state 3 is not d-reachable. Thus $S = \{\epsilon, a, aa\}$ is a state cover of M .

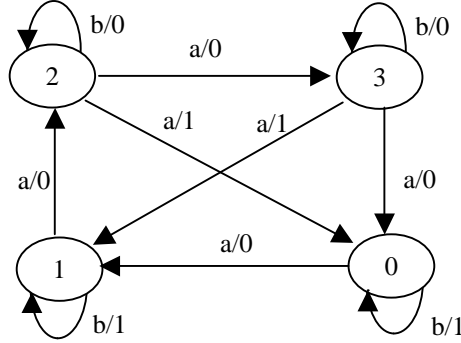


Fig. 1. The state-transition diagram of M

3.3 Distinguishing states

In order for an input sequence s to distinguish two states q and q' of M , it is sufficient that the corresponding sets of output sequences do not intersect, i.e. $h_2(q, s) \cap h_2(q', s) = \emptyset$. Two states for which there exists an input sequence with this property are said to be *separable*.

In the case of a non-deterministic FSM, however, it may be possible that there is no single input sequence that distinguishes between two states, rather these can be distinguished by a set of sequences. This idea leads to the, more general, concept of r -distinguishable states, formally defined in an inductive manner as follows [12]. States q and q' are said to be $r(1)$ -distinguishable if there exists $\sigma \in \Sigma$ such that $h_2(q, \sigma) \cap h_2(q', \sigma) = \emptyset$. States q and q' are said to be $r(k)$ -distinguishable, $k > 1$, if either q and q' are $r(j)$ -distinguishable for some j , $1 \leq j < k$, or there is some input $\sigma \in \Sigma$ such that for all $\gamma \in h_2(q, \sigma) \cap h_2(q', \sigma)$, the states $h_\gamma(q, \sigma)$ and $h_\gamma(q', \sigma)$ are $r(j)$ -distinguishable for some j , $1 \leq j < k$. States q and q' are said to be r -distinguishable if there exists some $k \geq 1$ such that q and q' are $r(k)$ -distinguishable. Clearly, any two separable states are r -distinguishable, but not vice versa. Naturally, the two notions coincide when the FSM is deterministic.

The definition of r -distinguishable ($r(k)$ -distinguishable) states naturally leads to the concept of r -distinguishing set ($r(k)$ -distinguishing set) of two states q and q' ; this can also be defined inductively [12]. A set of input sequences that contains an r -distinguishing ($r(k)$ -distinguishing) set of q and q' is said to r -distinguish ($r(k)$ -distinguish) q and q' .

A set $W \subseteq \Sigma^*$ of input sequences is called a *characterization set* of M if it r-distinguishes each pair of r-distinguishable states of M .

Consider again M in Fig. 1. The pairs of states $(0, 2)$, $(0, 3)$, $(1, 2)$, $(1, 3)$ and $(0, 1)$ are separable; the first four are r-distinguished by $\{b\}$, the last is r-distinguished by $\{ab\}$. On the other hand, states 2 and 3 are not separable, but they are r-distinguished by $\{ab, aab\}$. Thus $W = \{b, ab, aab\}$ is a characterization set of M .

3.4 Test suite generation

This section describes the generation of a test suite from a FSM using state counting. The method is from [5] and is essentially based on the results given in [12] for the case in which the IUT is known to be deterministic.

Suppose a state cover S and a characterization set W have been constructed. Q_S is used to denote the set of all d-reachable states of M . Let Q^1, \dots, Q^j denote the maximal sets of pairwise r-distinguishable states of M . Let also $Q_S^i = Q^i \cap Q_S$, $1 \leq i \leq j$.

Recall that the scope of testing is to check language inclusion between the (unknown) implementation and the specification. Thus, the task is to find a state q' in the implementation such that the input/output exhibited from q' is not allowed from the corresponding state q of the specification. A test suite will be then constructed using a breadth-first search through input/output sequences from each d-reachable state of M , in which the termination criterion is based on the observation that if a pair of states $(q, q') \in Q \times Q'$, from which a failure may be exhibited, is reachable then it is reachable by some minimal input/output sequence. Such a minimal sequence will not have visited the same pair of states twice and, furthermore cannot contain pairs of states that have already been reached by the sequences in S . More specifically, the following two ideas are used:

- If an input/output sequence (s, g) visits states of some Q^i , a tester can use W after each prefix of (s, g) to distinguish between the corresponding states visited along (s, g) in the implementation. If states from Q^i are visited n_i times along a minimal sequence (s, g) in the specification, then n_i distinct states will be visited in the implementation. Thus, n_i cannot exceed m' , the upper bound on the number of states of the implementation, by more than 1.
- There could be some d-reachable states among those in Q^i and the corresponding states in the implementation will also be reached by sequences from S ; this leaves $|Q_S^i|$ less pairs of states to explore.

By combining these two ideas, the breadth-first search can be ended once it has been established that states from some Q^i have been visited $m' - |Q_S^i| + 1$ times.

More formally, given a state $q \in Q_S$, the set $Tr(q)$, called a *traversal set* in [11], is constructed in the following way:

- A set $TrIO(q)$ is defined to consist of all input/output sequences (s, g) for which there exists i , $1 \leq i \leq j$, such that (s, g) visits states from Q^i exactly $m' - |Q_S^i| + 1$ times when followed from q (the initial state of the path is not included in the counting) and this condition does not hold for any proper prefix of (s, g) .
- $Tr(q)$ is the set of input sequences such that there is some corresponding input/output sequence in $TrIO(q)$, i.e. $Tr(q) = \{s \in \Sigma^* \mid \exists g \in \Gamma^* \cdot (s, g) \in TrIO(q)\}$.

Then the test suite produced is [5]:

$$Y = \bigcup_{s \in S} \{s\}pref(Tr(q_s))W$$

where for $s \in S$, q_s denotes the state reached by s .

When all the states of M are d-reachable and pairwise r-distinguishable, the test suite reduces to the set $S\Sigma[m' - m + 1]W$. This is equivalent to the test suite produced by the W -method when testing from a deterministic FSM. Where the specification does not satisfy these conditions, a larger test suite is required. Clearly, every sequence in $Tr(q)$ has length at most $m' + 1$. Based on the above definition, an algorithm for constructing $Tr(q)$ is provided in [5].

Consider the specification M as represented in Fig. 1 and the upper bound on the number of states of the implementation $m' = 4$. There is a single maximal set of pairwise r-distinguishable states, $Q^1 = \{0, 1, 2, 3\}$. Since $Q_S = \{0, 1, 2\}$, $Q_S^1 = \{0, 1, 2\}$. Thus the termination criterion for $TrIO(q)$ gives $m' - |Q_S^1| + 1 = 4 - 3 + 1 = 2$. Hence $Y = S\Sigma[2]W$.

4 Bounded sequence testing from non-deterministic FSMs

This section shows how the above test generation method can be extended to the case of bounded sequences. In this case, the test suite will contain only sequences of length less than or equal to an upper bound $l \geq 1$ and will have to establish if the IUT behaves as specified for all sequences in $\Sigma[l]$. More formally, $Y \subseteq \Sigma[l]$ is a test suite if and only if for every M' in the fault model, $M' \leq_{\Sigma[l]} M$ if and only if $M' \leq_Y M$.

The extension is not straightforward, as it is not sufficient to extract the prefixes of length at most l from the test suite produced in the unbounded case. Consider, for example, M_n , $n \geq 2$, as represented in Fig. 2 (a), $m' = n + 2$ and $l = n + 1$. All states of M_n are d-reachable and pairwise r-distinguishable, $S = \{\epsilon, a, \dots, a^n, b\}$ is a state cover of M_n and $W = \{a^n b\}$ is a characterization set of M_n . Thus $Y = S\Sigma[1]W = \{\epsilon, a, \dots, a^n, b\}\{\epsilon, a, b\}\{a^n b\}$ and $pref(Y) \cap \Sigma[n + 1] = pref(a^{n+1}) \cup pref(\{a^i b a^{n-i} \mid 0 \leq i \leq n\}) \cup pref(b a^{n-1})$. Consider M'_n as represented in Fig. 2 (b). Let $D = \{axbybz \mid x, y, z \in \Sigma^*, \|x\| + \|y\| + \|z\| \leq n - 2\} \subseteq \Sigma[n + 1]$. It can be observed that $M'_n \leq_{\Sigma[n+1] \setminus D} M_n$, but

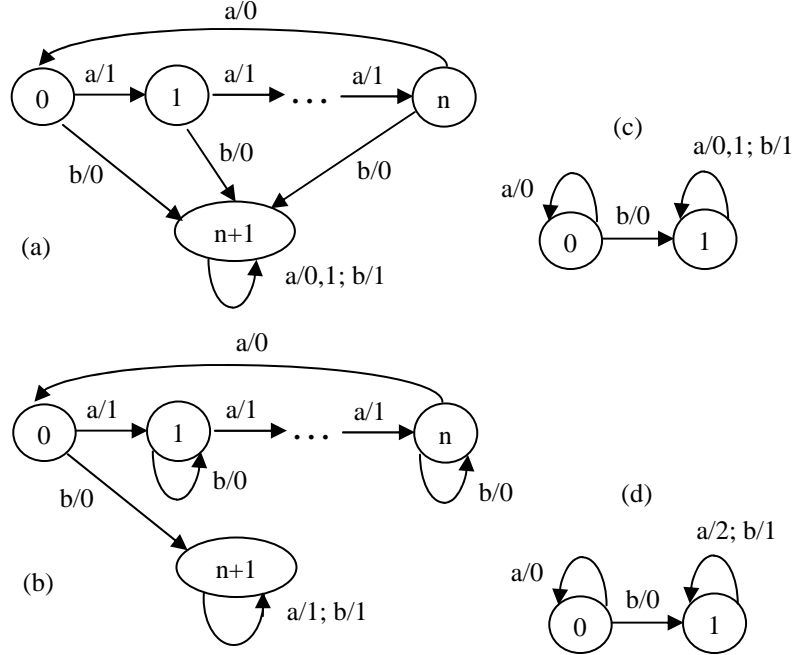


Fig. 2. The state-transition diagrams of M_n (a) and M'_n (b), M_0 (c) and M'_0 (d)

$M'_n \leq_{\{s\}} M_n$ does not hold for any sequence $s \in D$. Since $\text{pref}(Y) \cap D = \emptyset$, $M'_n \leq_{\text{pref}(Y) \cap \Sigma^{[n+1]}} M_n$.

In what follows, it will be shown that state counting can be used to generate tests for bounded sequences, provided that the sets S and W will contain sequences of *minimum length* that reach or distinguish states of M ; these sets will be called a proper state cover and a strong l -characterization set, respectively.

A few preliminary concepts are defined first. Without loss of generality, all FSMs considered are assumed to be initially connected, completely specified and observable and, furthermore, it will be assumed that every state can be reached by some sequence of length less than or equal to l .

For each state $q \in Q$, we define $\text{level}_M(q)$ as the length of the shortest path(s) from q_0 to q , i.e. $\text{level}_M(q) = \min\{\|s\| \mid s \in \Sigma^*, q \in h_1(q_0, s)\}$. For M as represented in Fig. 1, $\text{level}_M(i) = i$, $0 \leq i \leq 3$.

States p and q of M are said to be l -dissimilar if p and q are $r(k)$ -distinguishable for some $k \leq l - \max\{\text{level}_M(p), \text{level}_M(q)\}$. The notion of l -dissimilar (l -similar) states is originally introduced in [1] and is used in [1] and [7] for constructing a minimal deterministic automaton and a minimal deterministic stream X-machine for a finite language. For M as represented in Fig. 1 and $l = 4$, states 2 and 3 are not l -dissimilar since they are not $r(1)$ -distinguishable. On the other hand, every two other states of M are l -dissimilar.

Definition 1. A set $S \subseteq \Sigma^*$ of input sequences is called a proper state cover of M if S is a minimal set such that every state q of M that is d -reachable by some sequence of length $level_M(q)$ is d -reached by some sequence s_q from S and $\|s_q\| = level_M(q)$.

For M as represented in Fig. 1, $S = \{\epsilon, a, aa\}$ is a proper state cover of M .

The definition of a strong l -characterization set and the construction of the test suite are first given for a particular class of FSM specifications (quasi-deterministic FSMs) and then extended to the general type of FSM.

4.1 Quasi-deterministic FSMs

A quasi-deterministic FSM is a FSM in which for every $k > 0$, every pair of states that are not $\Sigma[k]$ -equivalent are $r(k)$ -distinguishable. In particular, this condition is satisfied by any deterministic FSM.

Definition 2. Suppose M is a quasi-deterministic FSM. A set $W \subseteq \Sigma^*$ of input sequences is called a strong l -characterization set of M , $l \geq 1$, if for every states p and q of M and every k , $0 < k \leq l - \max\{level_M(p), level_M(q)\}$, for which p and q are $r(k)$ -distinguishable, W $r(k)$ -distinguishes p and q .

Obviously, it is sufficient to check that W $r(k)$ -distinguishes p and q for the minimum integer $k \leq l - \max\{level_M(p), level_M(q)\}$ for which p and q are $r(k)$ -distinguishable. That is, the shortest possible sequences are included in W . Naturally, W will r -distinguish any two l -dissimilar states of M .

Consider again M_n as represented in Fig. 2 (a), $n \geq 2$. For every pair (i, j) , $0 \leq i < j \leq n$, i and j are $\Sigma[n - j]$ -equivalent and $r(n - j + 1)$ -distinguishable. Furthermore, $n + 1$ is $r(1)$ -distinguishable from any other state. Thus M_n is quasi-deterministic, but not deterministic. $W = \{a, \dots, a^n, b\}$ is a strong l -characterization set of M_n . On the other hand, M in Fig. 1 is not quasi-deterministic since states 2 and 3 are neither $\Sigma[2]$ -equivalent nor $r(2)$ -distinguishable.

4.2 Test suite generation

Suppose that the specification M is a quasi-deterministic FSM, S is a proper state cover of M and W is a strong l -characterization set of M . Q_S is used to denote the set of all states of M reached by sequences in S .

Let Q^1, \dots, Q^j denote the maximal sets of pairwise l -dissimilar states of M and let $Q_S^i = Q^i \cap Q_S$, $1 \leq i \leq j$. Under these conditions, the set $Tr(q_s)$ is defined analogously to section 3.4.

Then the test suite for bounded sequences is:

$$Z = \left(\bigcup_{s \in S} \{s\}pref(Tr(q_s))W_\epsilon \right) \cap \Sigma[l] \setminus \{\epsilon\}$$

where $W_\epsilon = W \cup \{\epsilon\}$.

When $Q_S = Q$ and all states of M are pairwise l -dissimilar, the test suite reduces to the set $S\Sigma[m' - m + 1]W_\epsilon \cap \Sigma[l] \setminus \{\epsilon\}$. This is equivalent to the test suite produced in [8] for deterministic FSMs.

Consider again M_n , $n \geq 2$, as represented in Fig. 2 (a), $m' = n + 2$, $l = n + 1$ and the IUT M'_n as represented in Fig. 2 (b). $S = \{\epsilon, a, \dots, a^n, b\}$ is a proper state cover of M_n and $W = \{a, \dots, a^n, b\}$ is a strong l -characterization set of M_n . There is a single maximal set of pairwise l -dissimilar states, $Q^1 = \{0, \dots, n + 1\}$. Since $Q_S = \{0, \dots, n + 1\}$, $Q^1_S = \{0, \dots, n + 1\}$. Thus $Z = S\Sigma[1]W_\epsilon \cap \Sigma[n + 1] \setminus \{\epsilon\} = \{\epsilon, a, \dots, a^n, b\} \setminus \{\epsilon\} \cup \{\epsilon, a, b\} \setminus \{\epsilon\} \cup \{\epsilon, a, \dots, a^n, b\} \cap \Sigma[n + 1] \setminus \{\epsilon\} = \{a^i \mid 1 \leq i \leq n + 1\} \cup \{a^i b a^j \mid 0 \leq i \leq n, 0 \leq j \leq n - i\} \cup \{a^i b b \mid 0 \leq i \leq n - 1\} \cup \{b b a^i \mid 1 \leq i \leq n - 1\} \cup \{b a b, b b b\}$. As $abb \in Z$, $M'_n \leq_Z M_n$ does not hold.

Note that W_ϵ , rather than only W , is needed in the definition of Z . Consider the specification M_0 as represented in Fig. 2 (c), $l = 2$, $m' = 2$ and the faulty implementation M'_0 as represented in Fig. 2 (d). The only sequence that detects the fault in the IUT is ba . $S = \{\epsilon, b\}$ is a proper state cover of M_0 and $W = \{b\}$ is a strong l -characterization set of M_0 . Thus $Z = S\Sigma[1]W_\epsilon \cap \Sigma[2] \setminus \{\epsilon\} = \{a, b, ab, ba, bb\}$. As $ba \in Z$, $M'_0 \leq_Z M_0$ does not hold. On the other hand, if W was used instead of W_ϵ in the definition of the test suite, then ba would not be contained in Z , so no fault would be detected.

4.3 General type of FSMs

We now consider the general type of FSM specifications. First, note that the test suite given above may not be valid when the specification is not quasi-deterministic. Consider, for example, M_1 as represented in Fig. 3 (a), $m' = 3$ and $l = 4$. M_1 is not quasi-deterministic since states 0 and 1 are neither $r(1)$ -distinguishable nor Σ -equivalent. All states of M_1 are d -reachable and pairwise l -dissimilar. Then, according to the above definitions, $S = \{\epsilon, a, aa\}$, $W = \{a, aa\}$ and $Z = S\Sigma[1]W_\epsilon \cap \Sigma[4] \setminus \{\epsilon\}$. Consider M'_1 as defined in Fig. 3 (b). It can be observed that $M'_1 \leq_{\Sigma[4] \setminus \Sigma^2\{bb\}} M_1$, but $M'_1 \leq_{\{s\}} M_1$ does not hold for any sequence $s \in \Sigma^2\{bb\}$. Since $Z \cap \Sigma^2\{bb\} = \emptyset$, Z will detect no fault in M'_1 .

Intuitively, this happens because, when M is not quasi-deterministic, there may be states p' and q' in the implementation M' that are r -distinguished by shorter sequences than those that r -distinguish the corresponding states p and q of the specification M . In our example, states 0 and 1 of M'_1 are r -distinguished by $\{b\}$, whereas states 0 and 1 of M_1 are not $r(1)$ -distinguishable and a longer sequence, aa , is used to r -distinguish between them. Consequently, the incorrect transition $h'(2, b) = (0, 1)$ cannot be detected by the above Z , since b was not included in W . On the other hand, the sequence $aabaa \in S\Sigma[1]W$, which results from the inclusion in W of the distinguishing sequence aa , has length 5 and, consequently, will not be contained in the test suite.

Now, observe that a sequence s can r -distinguish states p' and q' of the implementation only if the corresponding states p and q of the specification are not $\{s\}$ -equivalent, i.e. there exists $g \in \Gamma^*$ such that $(s, g) \in (L_M(p) \setminus L_M(q)) \cup (L_M(q) \setminus L_M(p))$. Thus, the problem can be addressed by extending W to include any sequence s for which there exist states of M that are neither $\{s\}$ -equivalent,

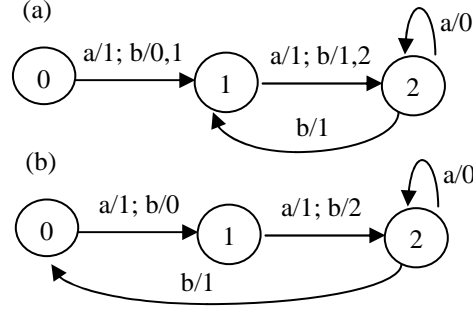


Fig. 3. The state-transition diagrams of M_1 (a) and M'_1 (b)

nor $r(\|s\|)$ -distinguishable. Then the definition of a strong l -characterization set can be extended to the general type of FSM as follows:

Definition 3. A set $W \subseteq \Sigma^*$ of input sequences is called a strong l -characterization set of M , $l \geq 1$, if the following two conditions hold:

- For every states p and q of M and every k , $0 < k \leq l - \max\{\text{level}_M(p), \text{level}_M(q)\}$, for which p and q are $r(k)$ -distinguishable, W $r(k)$ -distinguishes p and q .
- $s \in W$ for every $s \in \Sigma^*$ for which there exist states p and q of M with $\|s\| \leq l - \max\{\text{level}_M(p), \text{level}_M(q)\}$ such that p and q are neither $\{s\}$ -equivalent nor $r(\|s\|)$ -distinguishable.

With this revised definition of W , the construction of the suite remains the same as for quasi-deterministic FSM specifications. The following two sections of the paper provide the formal proofs to validate this construction.

For M_1 in the above example, states 0 and 1 are neither $\{b\}$ -equivalent, nor $r(1)$ -distinguishable. Thus $W = \{a, b, aa\}$. Then $aabb \in Z = S\Sigma[1]W_\epsilon \cap \Sigma[4] \setminus \{\epsilon\}$, so $M' \leq_Z M$ does not hold.

Consider again M as represented in Fig. 1, $m' = 4$ and $l = 4$. $S = \{\epsilon, a, aa\}$ is a proper state cover of M and $Q_S = \{0, 1, 2\}$. The pairs of states $(0, 2)$, $(0, 3)$, $(1, 2)$ and $(1, 3)$ are r -distinguished by $\{b\}$; 0 and 1 are r -distinguished by $\{ab\}$ and are Σ -equivalent. Since states 2 and 3 are Σ -equivalent, no other sequence needs to be included in W . Thus $W = \{b, ab\}$ is a strong l -characterization set of M . The maximal sets of pairwise l -dissimilar states of M are $Q^1 = \{0, 1, 2\}$ and $Q^2 = \{0, 1, 3\}$. Thus $Q_S^1 = \{0, 1, 2\}$ and $Q_S^2 = \{0, 1\}$ and the two termination criteria for $TrIO(q)$ give $m' - |Q_S^1| + 1 = 4 - 3 + 1 = 2$ and $m' - |Q_S^2| + 1 = 4 - 2 + 1 = 3$, respectively. The tree generated in the construction of $TrIO(1)$ is represented in Fig. 4. A node is a leaf if the path from the root to it has visited (after the root) $n_1 = 2$ states from Q^1 or $n_2 = 3$ states from Q^2 . On the other hand, only paths of length at most $l - \text{level}_M(1) = 4 - 1 = 3$ need to be constructed; in Fig. 4, the remaining branches are drawn with dashed line.

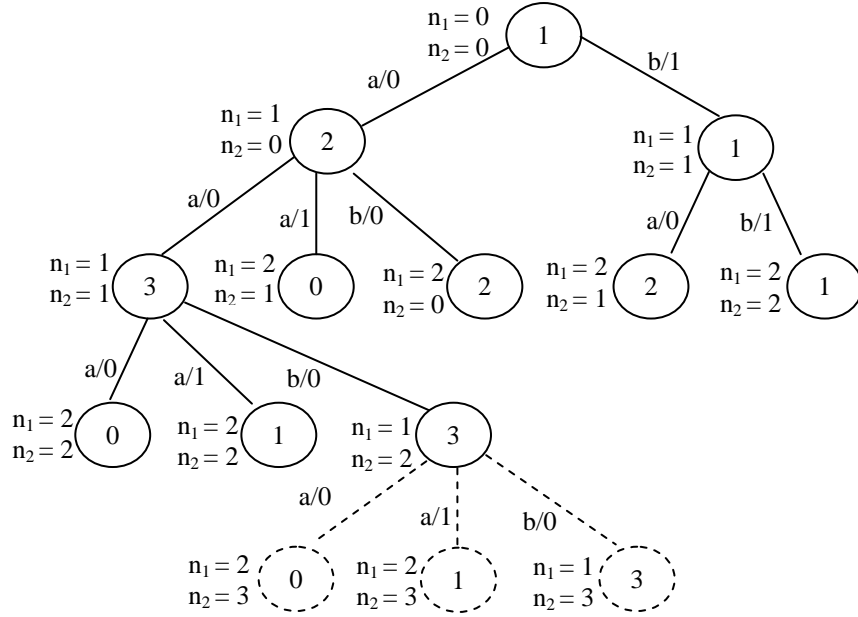


Fig. 4. The tree associated with $TrIO(2)$

5 The l -bounded product FSM

In order to compare the languages associated with two observable FSMs M and M' , one can build a cross-product of their states, such that states (q, q') of the cross-product FSM correspond to pairs of states q, q' in the two FSMs. A transition on input σ and output γ between states (q, q') and (p, p') exists in the cross-product FSM if and only if the transitions $(p, \gamma) \in h(q, \sigma)$ and $(p', \gamma) \in h'(q', \sigma)$ exist in M and M' , respectively. The result of such a construction corresponds to the intersection of the languages $L(M)$ and $L(M')$. When checking that M' is a reduction of M , a transition in M' that is not allowed by M will lead in the cross-product FSM to a *Fail* state. When only the results produced by the two FSMs in response to input sequences of length at most l are compared, an integer i , $1 \leq i \leq l$, can be added to the state space and incremented by each transition. No transition needs to be defined for $i = l$. The resulting construction will be called an l -bounded product FSM of M and M' .

Definition 4. Given $l \geq 1$, the l -bounded product FSM formed from $M = (\Sigma, \Gamma, Q, h, q_0)$ and $M' = (\Sigma, \Gamma, Q', h', q'_0)$ is the FSM $P_l(M, M') = (\Sigma, \Gamma, Q_P, H, (q_0, q'_0, 0))$ in which $Q_P = Q \times Q' \times \{0, \dots, l\} \cup \{Fail\}$ with $Fail \notin Q \times Q' \times \{0, \dots, l\}$ and H is defined by the following rules for all $(q, q'), (p, p') \in Q \times Q', i \in \{0, \dots, l-1\}, \sigma \in \Sigma$ and $\gamma \in \Gamma$:

- if $(p, \gamma) \in h(q, \sigma)$ and $(p', \gamma) \in h'(q', \sigma)$ then $((p, p', i+1), \gamma) \in H((q, q', i), \sigma)$.

– if $(p', \gamma) \in h'(q', \sigma)$ and $\gamma \notin h_2(q, \sigma)$ then $(Fail, \gamma) \in H((q, q', i), \sigma)$

and is undefined elsewhere.

As M and M' are observable, $P_l(M, M')$ is also observable (when M and M' are both deterministic, $P_l(M, M')$ is also deterministic [8]). On the other hand, $P_l(M, M')$ is not completely specified even though M and M' are completely specified. More importantly, checking $M' \leq_{\Sigma[l]} M$ corresponds to establishing if the *Fail* state of $P_l(M, M')$ is reachable.

Lemma 1. *The Fail state of $P_l(M, M')$ is not reachable if and only if $M' \leq_{\Sigma[l]} M$.*

Proof: From Definition 4, it follows that, for every $s \in \Sigma^*$ and $g \in \Gamma^*$, $H_g((q_0, q'_0, 0), s) = Fail$ if and only if $s = s_1\sigma$ with $s_1 \in \Sigma[l-1]$ and $\sigma \in \Sigma$ and $g = g_1\gamma$ with $g_1 \in \Gamma[l-1]$ and $\gamma \in \Gamma$ such that $g_1 \in h_2(q_0, s_1) \cap h'_2(q'_0, s_1)$ and $g_1\gamma \in h'_2(q'_0, s_1\sigma) \setminus h_2(q_0, s_1\sigma)$.

6 State counting for bounded sequences

State counting can now be used to prove that, whenever the *Fail* state is reachable, it will be reached by some sequence in the test suite. As in the unbounded case, it will be shown that the test suite contains all “minimal” input sequences that could reach *Fail*. Among the shortest sequences, the minimal sequences are those for which also the “distance” (defined in what follows) to the set S is the shortest. The basic idea is similar to that used in bounded sequence testing from deterministic FSM specifications [8]; however, when considering non-deterministic FSMs, we have to take into account that non-equivalent states may not necessarily be r-distinguishable (see Lemma 3).

Given $x \in \Sigma^*$ and $A \subseteq \Sigma^*$ with $\epsilon \in A$, the length of the shortest sequences $(s) t \in \Sigma^*$ for which there exists a sequence $s \in A$ such that $st = x$ is denoted by $d(x, A)$, i.e. $d(x, A) = \min(\{\|t\| \mid t \in \Sigma^*, \exists s \in A \cdot st = x\})$. Since $\epsilon \in A$, the set $\{t \in \Sigma^* \mid \exists s \in A \cdot st = x\}$ is not empty, so $d(x, A)$ is well defined.

Lemma 2. *Let $p, q \in Q$, $p', q' \in Q'$, $U \subseteq \Sigma^*$ and $k > 0$. If $p' \leq_{U \cap \Sigma^{[k]}} p$, $q' \leq_{U \cap \Sigma^{[k]}} q$ and U $r(k)$ -distinguishes p and q then U $r(k)$ -distinguishes p' and q' .*

Proof: Follows by induction on k .

Lemma 3. *Let $s \in S$ and $t \in Tr(q_s)$ such that $\|st\| \leq l$ and s is the longest prefix of st that is in S . If $M' \leq_{(S \cup \{s\}pref(t)W_\epsilon) \cap \Sigma[l]} M$ then there exist $y_1 \in \{s\}pref(t) \setminus \{s\}$, $y_2 \in S \cup pref(y_1) \setminus \{y_1\}$ and $w_1, w_2 \in \Gamma^*$ such that the following two conditions hold:*

- $\|y_2\| < \|y_1\|$ or $\|y_2\| \leq \|y_1\|$ and $d(y_2, S) < d(y_1, S)$
- $H_{w_1}((q_0, q'_0, 0), y_1) = (q_1, q', \|y_1\|)$ and $H_{w_2}((q_0, q'_0, 0), y_2) = (q_2, q', \|y_2\|)$ for some states $q_1, q_2 \in Q$ and $q' \in Q'$ such that $L_M(q_1) \cap L_{M'}(q') \cap (\Sigma \times \Gamma)[l - \|y_1\|] = L_M(q_2) \cap L_{M'}(q') \cap (\Sigma \times \Gamma)[l - \|y_1\|]$.

Proof: Let $i, 1 \leq i \leq j$. Suppose s_1 and s_2 are two distinct elements of S such that $q_{s_1}, q_{s_2} \in Q^i$ and let $q'_{s_1} \in h_1(q_0, s_1)$ and $q'_{s_2} \in h_1(q_0, s_2)$. Since q_{s_1} and q_{s_2} are l -dissimilar there exists $k, 0 < k \leq l - \max\{\text{level}_M(q_{s_1}), \text{level}_M(q_{s_2})\}$, such that q_{s_1} and q_{s_2} are $r(k)$ -distinguishable. As W is a strong l -characterization set of M , W $r(k)$ -distinguishes q_{s_1} and q_{s_2} . Since $M' \leq_{SW \cap \Sigma[l]} M$, $q'_{s_1} \leq_{W \cap \Sigma[k]} q_{s_1}$ and $q'_{s_2} \leq_{W \cap \Sigma[k]} q_{s_2}$. Thus, by Lemma 2, q'_{s_1} and q'_{s_2} are $r(k)$ -distinguishable. Consequently, the sequences in S will reach at least $|Q_S^i|$ distinct states of M' .

On the other hand, since $t \in Tr(q_s)$, there is some $g \in \Gamma^*$ and $i, 1 \leq i \leq j$, such that (t, g) visits states from Q^i exactly $m' - \text{card}(Q_S^i) + 1$ times when followed from q_s . Since S has already reached at least $|Q_S^i|$ states, there will be a state q' of M' that either has been visited twice by (t, g) or has been reached by some sequence in S . Thus, there exist $y_1 \in \{s\}pref(t) \setminus \{s\}$, $y_2 \in S \cup \{s\}pref(y_1) \setminus \{y_1\}$ and $w_1, w_2 \in \Gamma^*$ such that $h_{w_1}(q_0, y_1) = q_1$, $h_{w_2}(q_0, y_2) = q_2$, $h'_{w_1}(q'_0, y_1) = q'$ and $h'_{w_2}(q'_0, y_2) = q'$ for some states $q_1, q_2 \in Q^i$ and $q' \in Q'$. Then $H_{w_1}((q_0, q'_0, 0), y_1) = (q_1, q', \|y_1\|)$ and $H_{w_2}((q_0, q'_0, 0), y_2) = (q_2, q', \|y_2\|)$.

Let $\mu = \max\{\|y_1\|, \|y_2\|\}$. We prove by contradiction that q_1 and q_2 are not $r(l - \mu)$ -distinguishable. Assume q_1 and q_2 are $r(l - \mu)$ -distinguishable, $\mu < l$. Since W is a strong l -characterization set of M , W $r(l - \mu)$ -distinguishes q_1 and q_2 . On the other hand, since $M' \leq_{\{y_1, y_2\}W \cap \Sigma[l]} M$, $q' \leq_{W \cap \Sigma[l - \mu]} q_1$ and $q' \leq_{W \cap \Sigma[l - \mu]} q_2$. Thus, by Lemma 2, W would $r(l - \mu)$ -distinguish q' from itself. This is obviously a contradiction.

We now show that $\|y_2\| < \|y_1\|$ or $\|y_2\| \leq \|y_1\|$ and $d(y_2, S) < d(y_1, S)$. If $y_2 \in pref(y_1) \setminus \{y_1\}$ then $\|y_2\| < \|y_1\|$. Otherwise $y_2 \in S \setminus \{s\}$, so $\|y_2\| = \text{level}_M(q_2)$. Then there are two cases:

- $q_1 = q_2$. Then $\text{level}_M(q_2) \leq \|y_1\|$ so $\|y_2\| \leq \|y_1\|$. Since $y_1 \notin S$ and $y_2 \in S$, $d(y_2, S) < d(y_1, S)$.
- $q_1 \neq q_2$. We prove by contradiction that $\|y_2\| < \|y_1\|$. Assume $\|y_1\| \leq \|y_2\|$. Then $\text{level}_M(q_1) \leq \|y_1\| \leq \|y_2\| = \text{level}_M(q_2)$. Hence $\text{level}_M(q_1) \leq \text{level}_M(q_2) = \|y_2\|$. As $q_1, q_2 \in Q^i$, q_1 and q_2 are $r(l - \|y_2\|)$ -distinguishable. On the other hand, we have shown that q_1 and q_2 are not $r(l - \mu)$ -distinguishable. Since $\mu = \|y_2\|$, this is a contradiction.

Thus $\|y_2\| < \|y_1\|$ or $\|y_2\| \leq \|y_1\|$ and $d(y_2, S) < d(y_1, S)$. Since $\|y_2\| \leq \|y_1\|$, $\mu = \|y_1\|$, so q_1 and q_2 are not $r(l - \|y_1\|)$ -distinguishable.

Finally, we prove by contradiction that $L_M(q_1) \cap L_{M'}(q') \cap (\Sigma \times \Gamma)[l - \|y_1\|] = L_M(q_2) \cap L_{M'}(q') \cap (\Sigma \times \Gamma)[l - \|y_1\|]$. Assume otherwise and let $(s_0, g_0) \in L_{M'}(q') \cap (\Sigma \times \Gamma)[l - \|y_1\|] \cap ((L_M(q_2) \setminus L_M(q_1)) \cup (L_M(q_1) \setminus L_M(q_2)))$. Since W is a strong l -characterization set of M and q_1 and q_2 are not $r(l - \|y_1\|)$ -distinguishable, $s_0 \in W$. As $M' \leq_{\{y_1, y_2\}W \cap \Sigma[l]} M$ and $\|y_2 s_0\| \leq \|y_1 s_0\| \leq l$, it follows that $M' \leq_{\{y_1 s_0, y_2 s_0\}} M$. Thus, since $(s_0, g_0) \in L_{M'}(q')$, $(s_0, g_0) \in L_M(q_1)$ and $(s_0, g_0) \in L_M(q_2)$. This provides a contradiction, as required.

Lemma 4. *Let $(q_1, q', j_1), (q_2, q', j_2) \in Q \times Q' \times \{0, \dots, l\}$, $0 \leq j_2 \leq j_1 \leq l - 1$, and $(x, w) \in (\Sigma \times \Gamma)[l - j_1]$. Suppose $L_M(q_1) \cap L_{M'}(q') \cap (\Sigma \times \Gamma)[l - j_1] = L_M(q_2) \cap L_{M'}(q') \cap (\Sigma \times \Gamma)[l - j_1]$. If $H_w((q_1, q', j_1), x) = \text{Fail}$ then $H_w((q_2, q', j_2), x) = \text{Fail}$.*

Proof: If $H_w((q_1, q', j_1), x) = Fail$ then $x = s\sigma$ with $s \in \Sigma[l - j_1 - 1]$, $\sigma \in \Sigma$ and $w = g\gamma$ with $g \in \Gamma[l - j_1 - 1]$, $\gamma \in \Gamma$ such that $g \in h_2(q_1, s) \cap h'_2(q', s)$ and $g\gamma \in h'_2(q', s\sigma) \setminus h_2(q_1, s\sigma)$. Since $L_M(q_1) \cap L_{M'}(q') \cap (\Sigma \times \Gamma)[l - j_1] = L_M(q_2) \cap L_{M'}(q') \cap (\Sigma \times \Gamma)[l - j_1]$, $g \in h_2(q_2, s) \cap h'_2(q', s)$ and $g\gamma \in h'_2(q', s\sigma) \setminus h_2(q_2, s\sigma)$. As $j_2 \leq j_1$, it follows that $H_w((q_2, q', j_2), x) = Fail$.

Lemma 5. *If $M' \leq_Z M$ then the Fail state of $P_l(M, M')$ is not reachable.*

Proof: We provide a proof by contradiction. Assume *Fail* is reachable and let X be the set of all sequences of minimum length that reach *Fail*. Let $\mu = \min\{d(x, S) \mid x \in X\}$ and $X_\mu = \{x \in X \mid d(x, S) = \mu\}$.

We prove by contradiction that $X_\mu \cap (\bigcup_{s \in S} \{s\}pref(Tr(q_s))) \neq \emptyset$. Assume $X_\mu \cap (\bigcup_{s \in S} \{s\}pref(Tr(q_s))) = \emptyset$ and let $x \in X_\mu$. Then $x \notin \bigcup_{s \in S} \{s\}pref(Tr(q_s))$. Since $\epsilon \in S$, $x \in S\Sigma^*$. Let $s \in S$ be the longest prefix of x that is in S . Then $x = stu$, for some $t \in Tr(q_s)$ and $u \in \Sigma^* \setminus \{\epsilon\}$ with $\|stu\| \leq l$ and there exist $g, v \in \Gamma^*$ such that $g \in h_2(q_0, st) \cap h'_2(q'_0, st)$ and $gv \in h'_2(q'_0, stu) \setminus h_2(q_0, stu)$. Since $M' \leq_Z M$ and $(S \cup \{s\}pref(t)W_\epsilon) \cap \Sigma[l] \subseteq Z$, by Lemma 3, there exist $y_1 \in \{s\}pref(t) \setminus \{s\}$, $y_2 \in S \cup pref(y_1) \setminus \{y_1\}$ and $w_1, w_2 \in \Gamma^*$ such that the following two conditions hold:

- $\|y_2\| < \|y_1\|$ or $\|y_2\| \leq \|y_1\|$ and $d(y_2, S) < d(y_1, S)$
- $H_{w_1}((q_0, q'_0, 0), y_1) = (q_1, q', \|y_1\|)$ and $H_{w_2}((q_0, q'_0, 0), y_2) = (q_2, q', \|y_2\|)$ for some states $q_1, q_2 \in Q$ and $q' \in Q'$ such that $L_M(q_1) \cap L_{M'}(q') \cap (\Sigma \times \Gamma)[l - \|y_1\|] = L_M(q_2) \cap L_{M'}(q') \cap (\Sigma \times \Gamma)[l - \|y_1\|]$.

Let $z \in \Sigma^*$ such that $st = y_1z$ and $w_z \in \Gamma^*$ such that $g = w_1w_z$. As $H_{g v}((q_0, q'_0, 0), x) = Fail$, $H_{w_z v}((q_1, q', \|y_1\|), zu) = Fail$. Then, by Lemma 4, $H_{w_z v}((q_2, q', \|y_2\|), zu) = Fail$. Thus $H_{w_2 w_z v}((q_0, q'_0, 0), y_2 zu) = Fail$. If $\|y_2\| < \|y_1\|$ then $y_2 zu$ is a sequence shorter than x that reaches *Fail*. Thus $x \notin X$, which is a contradiction. Otherwise, $\|y_2\| = \|y_1\|$ and $d(y_2, S) < d(y_1, S)$. Since no sequence in $\{y_1\}pref(zu)$ is contained in S , $d(y_2 zu, S) < d(y_1 zu, S)$. Consequently $\|y_2 zu\| = \|x\|$ and $d(y_2 zu, S) < d(x, S)$. Thus $x \notin X_\mu$, which provides a contradiction, as required. Hence $X_\mu \cap (\bigcup_{s \in S} \{s\}pref(Tr(q_s))) \neq \emptyset$.

On the other hand, since $M' \leq_Z M$, no sequence in Z will reach *Fail*. Thus $X_\mu \cap Z = \emptyset$. As $\bigcup_{s \in S} \{s\}pref(Tr(q_s)) \subseteq Z$, this provides a contradiction, as required. Hence *Fail* is not reachable.

Theorem 1. *$M' \leq_{\Sigma[l]} M$ if and only if $M' \leq_Z M$.*

Proof: “ \Rightarrow ”: Obvious, since $Z \subseteq \Sigma[l]$. “ \Leftarrow ”: Follows from Lemmas 5 and 1.

7 Conclusions

This paper extends the state counting based method of deriving tests from a non-deterministic FSM to the case of bounded sequences. The method for bounded sequences has practical value, as many applications of finite state machines actually use only input sequences of limited length. In such applications, the test

suite produced may contain only a small fraction of all sequences of length less than or equal to the upper bound. The test suite for M_n in our example (Fig. 2 (a)), $m' = n + 2$ and $l = n + 1$ will contain only $(n^2 + 9n + 6)/2$ sequences out of a total of $2^{n+2} - 2$ sequences.

Improvements in the size of the test suite may be obtained by using only subsets of W to identify the states reached by the sequences in $Tr(q_s)$, in a way similar to the Wp method for unbounded [3] and bounded [8] sequences. This will be the subject of a future paper. Possible future work also involves the generalization of these bounded sequence testing methods to classes of extended finite state machines, such as stream X-machines [6].

References

1. Campeanu, C., Santean, N., Yu, S. Minimal cover automata for finite languages. *Theoretical Computer Science*, 267, 3-16 (1999)
2. Chow, T. S. Testing software design modeled by finite state machines, *IEEE Transactions on Software Engineering*, 4(3), 178-187 (1978)
3. Fujiwara, S., von Bochmann, G., Khendek, F., Amalou, M. and Ghedamsi A. Test Selection Based on Finite State Models. *IEEE Transactions on Software Engineering*, 17(6), 591-603 (1991)
4. Hierons, R. M. Adaptive testing of a deterministic implementation against a non-deterministic finite state machine. *The Computer Journal*, 41(5), 349-355 (1998)
5. Hierons, R. M. Testing from a Non-Deterministic Finite State Machine Using Adaptive State Counting. *IEEE Transactions on Computers* 53(10), 1330-1342 (2004)
6. Holcombe, M., Ipate, F. *Correct Systems: Building a Business Process Solution*. Springer Verlag, Berlin (1998)
7. Ipate, F. On the Minimality of Finite Automata and Stream X-machines for Finite Languages, *The Computer Journal*, 48(2), 157-167 (2005)
8. Ipate, F. Bounded Sequence Test Selection from Finite State Machines, submitted.
9. Lee, D. and Yannakakis, M. Principles and Methods of Testing Finite State Machines - A Survey. *Proceedings of the IEEE*, 84(8), 1090-1123 (1996)
10. Luo, G. L., Bochmann, G. v. and Petrenko, A. Test selection based on communicating nondeterministic finite-state machines using a generalized Wp-method. *IEEE Transactions on Software Engineering*, 20(2), 149-161 (1994)
11. Petrenko, A., Yevtushenko, N., Lebedev, A. and Das, A. Nondeterministic state machines in protocol conformance testing. In *Proc. of Protocol Test Systems, VI (C-19)*, Pau, France, 28-30 September, Elsevier Science, 363-378 (1994)
12. Petrenko, A., Yevtushenko, N., Bochmann G.v. Testing deterministic implementations from nondeterministic FSM specifications. In *Proc. of 9th International Workshop on Testing of Communicating Systems (IWTC'S'96)*, Darmstadt, Germany, 9-11 September 1996, Chapman and Hall, 125-140 (1996)
13. Sidhu, D. and Leung, T. Formal methods for protocol testing: A detailed study. *IEEE Transactions on Software Engineering*, 15(4), 413-426, 1989.
14. Yevtushenko, N. V., Lebedev, A. V. and Petrenko, A. F. On checking experiments with nondeterministic automata. *Automatic Control and Computer Sciences*, 6, 81-85 (1991)