Experimental Evaluation of Coverage Criteria for FSM-based Testing[*]

Adenilso Simão [1], Alexandre Petrenko [2], Jose Carlos Maldonado [1]

[1] Instituto de Ciências Matemáticas e de Computação
Universidade de São Paulo
São Carlos, São Paulo, Brazil

[2] Centre de recherche informatique de Montreal (CRIM)
Montreal, Quebec, Canada

{adenilso,jcmaldon}@icmc.usp.br, petrenko@crim.ca

Resumo. *Para planejar a atividade de teste, os testadores devem determinar uma estratégia, incluindo um critério de cobertura, que possua uma boa relação de custo/benefício referente aos recursos disponíveis e os objetivos de teste. Propriedades teóricas conhecidas dos critérios de cobertura nem sempre são suficientes e, dessa forma, dados empíricos são necessários. Neste artigo, são apresentados resultados de uma avaliação experimental de vários critérios de cobertura usados comumente para Máquinas de Estados Finitos.*

Abstract. *In order to plan testing activities, testers face the challenge of determining a strategy, including a test coverage criterion, that offers an acceptable compromise between the available resources and test goals. Known theoretical properties of coverage criteria do not always help and thus empirical data are needed. In this paper, we present the results of an experimental evaluation of several commonly used coverage criteria for Finite State Machines.*

1. Introduction

Model-based testing refers to the derivation of test suites from a model representing software behavior. Behavior models can be constructed early in the development cycle, allowing testing activities to start before the coding phase, as tests can be based on what the software should do, and not on what the software does. Finite State Machines (FSMs) are state-based models which have been intensively used in conformance testing of protocols [Bochmann and Petrenko 1994] and object-oriented software testing [Binder 2000]. The existence of several methods for test generation from state-based models provides flexibility for testers to devise effective testing strategies.

Test generation methods are based on coverage criteria. A coverage criterion defines a set of testing requirements that must be covered by an adequate test suite. It is usually derived from elements of the model that the tester considers important to be tested. There exist several coverage criteria that can be used to guide test generation, as

---

well as to assess the quality of a given test suite. Usually, the cost of a coverage criterion can be estimated by the length of a test suite that is required to satisfy it. When one has to choose among several coverage criteria, it is desirable to use the strongest, most effective applicable criterion, i.e., the criterion that has the highest probability to reveal the faults in the implementation under test with a minimum cost. A high fault detection capability usually comes with the price: the tests may simply explode and then a weaker criterion might be used instead. Budget and schedule constraints must also be taken into account. For instance, if the tests are manually executed, their total length should be much shorter than those executed autoamtically. Therefore, it is important to be able to estimate the length of tests adequate to various.

The comparison of test coverage criteria can be based on their theoretical properties, e.g., upper bounds for test lengths and subsumption relationships [Frankl and Weyuker 1993]. As an example, Binder [2000] discusses the tradeoffs of various state-based test strategies, highlighting the importance of comparing the expected length of test suites generated by different approaches when a test strategy must be chosen. The discussion is based on the worst-case minimum and maximum lengths. However, the maximum lengths are reached for FSMs with a special structure, e.g., Moore lock FSMs which require the longest sequence to reach and identify a certain state [Moore 1956]. Thus, the available upper bounds have to be used with care. It is important to have at least some indications on the average lengths of the test suites adequate for various coverage criteria. Concerning the subsumption relation, it can be established for some criteria, however, not all of them are comparable w.r.t. this relation. Then, it is important to have other means of comparing such criteria. In this context, experimental data are useful for choosing coverage criteria and defining effective testing strategies. Experimental data characterizing the average lengths of test suites adequate to various criteria help in assessing the applicability of a particular criterion. Furthermore, assuming the tester has chosen a given criterion, an important question is how the test suites adequate to this criterion relate to others, to know how the cost would change if the tester decides to generate a test suite that is adequate according to another stronger criterion.

Despite the importance of experimental data, there is a lack of work in the literature that provides those concerning FSM tests. The book [Binder 2000] refers just the worst-case test lengths. We are aware of only the work of Dorofeeva et al. [2005b] which reports the results of an experiment comparing various test generation methods. However, no experimental comparison among coverage criteria for FSMs is available. In this paper, we present an experimental comparison of test coverage criteria for FSMs. We consider four representative criteria: state coverage, transition coverage, initialization fault coverage, and transition fault coverage criteria and provide experimental data on the length of tests generated from an FSM specification to satisfy these coverage criteria. We investigate the impacts of FSM parameters on the cost associated with the usage of those criteria. We are interested in comparing the average length of the test suites for those criteria both to each other and to the theoretical upper limits. The experiments involve random generation of FSM specifications and tests in order to provide experimental characterization of how the test length depends on FSM parameters and coverage criteria.

The paper is organized as follows. Section 2 contains basic definitions related to FSMs and test suites. In Section 3, we present the main concepts related to test coverage criteria and define the criteria we investigate in this paper. The discussion on how to compare the cost of different criteria based on the length of the adequate test suites is presented in Section 4. Section 5 explains the settings of the experiments we conducted

to compare the criteria. The results of the experiments and their analyses are presented in Section 6. In Section 7, we discuss the threats to the validity of the results. Finally, in Section 8, we draw concluding remarks and point to future work.

2. FSM and Tests

A Finite State Machine is a deterministic Mealy machine, which can be defined as follows.

Definition 1. *A* Finite State Machine (FSM) *M is a 7-tuple* (S, s_0, I, O, D, δ, λ)*, where*
- S *is a finite set of states with the initial state* s_0*,*
- I *is a finite set of inputs,*
- O *is a finite set of outputs,*
- $D \subseteq S \times I$ *is a specification domain,*
- $\delta : D \rightarrow S$ *is a transition function, and*
- $\lambda : D \rightarrow S$ *is an output function.*

A FSM M is said to be *completely* specified (a complete FSM, CFSM), if $D = S \times I$. Otherwise, M is called a *partially* specified machine (a partial FSM, PFSM). A tuple (s, x) $\in D$ is a *transition* of M. Figure **1**a presents an example of a partial FSM. The initial state is highlighted in bold.
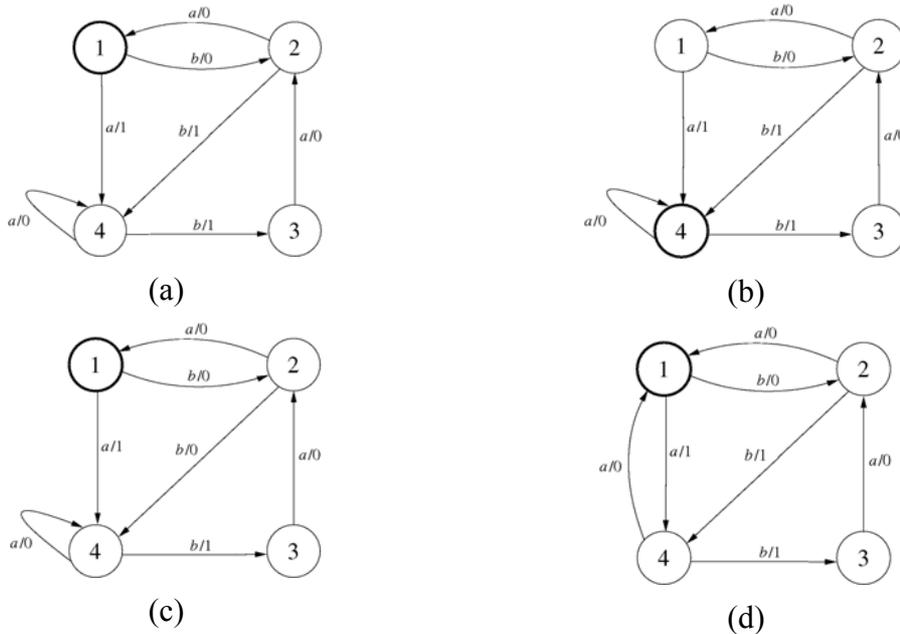

Figure 1. (a) A partial FSM. Mutants of the FSM with: (b) initialization fault; (c) output fault; and (d) transfer fault.

A string $x_1...x_k \in I^*$ is said to be a *defined* input sequence at state $s \in S$ if there exist $s_1, ..., s_k, s_{k+1}$, where $s_1 = s$, such that (s_i, x_i) $\in D$ and $\delta(s_i, x_i) = s_{i+1}$ for all $i = 1, ..., k$. We use $\Omega_M(s)$ to denote the set of all defined input sequences for state s and Ω_M as a shorthand for $\Omega_M(s_0)$, i.e., for the input sequences defined for the initial state of M and, hence, for M itself. Given sequences α, $\beta \in I^*$, we write $\alpha \leq \beta$, if α is a *prefix* of β, and $\alpha < \beta$, if α is a proper prefix of β. For a sequence $\beta \in I^*$, *pref*(β) is the set of prefixes of β, i.e., *pref*(β) = {$\alpha \mid \alpha \leq \beta$}. For a set of sequences $T \subseteq I^*$, *pref*(T) is the union of *pref*(β), for all $\beta \in T$; T is *prefix-closed* if $T = pref(T)$.

We extend the transition and output functions from input symbols to defined input sequences, including the empty sequence ε, assuming that $\delta(s, \varepsilon) = s$ and $\lambda(s, \varepsilon) = \varepsilon$ for any $s \in S$. A sequence $\alpha \in \Omega_M$ is a *transfer* sequence to a state s, if $\delta(s_0, \alpha) = s$. A FSM M is said to be *initial connected* if for each state $s \in S$ there exists a transfer sequence to s. In this paper, we assume that FSMs for which tests are generated are initially connected. A natural r is called an *accessibility degree* of the FSM M if for each state there exists a transfer sequence to the state with at most r input symbols.

Given an FSM $M = (S, s_0, I, O, D, \delta, \lambda)$, states s and t are *distinguishable*, denoted by $s \nsim t$, if there exists an input sequence $\gamma \in \Omega_M(s) \cap \Omega_M(t)$, such that $\lambda(s, \gamma) \neq \lambda(t, \gamma)$; γ is called a *separating* sequence for s and t. A natural d is called a *distinguishability degree* of the FSM M if for any two distinguishable states there exists a separating sequence with at most d input symbols. An FSM is *reduced*, if all state pairs are distinguishable.

Definition 2. *A defined input sequence of FSM M is called a* test case *(or simply a* test*) of M. A* test suite *of M is a finite set of tests of M, such that no test is a proper prefix of another test.*

To model implementation faults we use the notion of a mutant of a given specification FSM.

Definition 3. *Given a specification FSM $M = (S, s_0, I, O, D, \delta, \lambda)$, a* mutant *of M is any FSM over the state set S and input set I.*

A mutant N is *distinguishable* from M, denoted $N \nsim M$, if there exists $\gamma \in \Omega_M \cap \Omega_N$ such that $\lambda(s_0, \gamma) \neq \Lambda(s'_0, \gamma)$. We say that γ *kills N. N* has a *transfer* fault in the transition $(s, x) \in D$ with respect to M, if $\delta(s, x) \neq \Delta(s, x)$. N has an *output* fault in the transition $(s, x) \in D$ with respect to M, if $\lambda(s, x) \neq \Lambda(s, x)$. N has an *initialization* fault with respect to M, if $s_0 \neq s'_0$. N has a *transition* fault in $(s, x) \in D$ with respect to M, if it has an output, transfer, or both faults. Figure 1 shows examples of mutants with each of these faults. The mutant in Figure 1b has an initialization fault, since the initial state is changed to state 4. The mutant in Figure 1c has an output fault in the transition $(2, b)$. The mutant in Figure 1d has a transfer fault in the transition $(4, a)$.

3. Test Coverage Criteria

A test coverage criterion can be thought of as a systematic way of defining testing requirements, which an adequate test suite must fulfill. Therefore, we can compare two test suites with respect to a given criterion by analyzing the set of testing requirements they satisfy. Let K be a test coverage criterion. We use $TR_K(M)$ to denote the set of testing requirements the criterion K defines for a given FSM M. Let T be a test suite. We define $TS_K(M, T) \subseteq TR_K(M)$ as the set of testing requirements that are satisfied by T. The *test coverage* of T, denoted $C_K(M, T)$, is the ratio between the number of testing requirements it fulfills and the total number of testing requirements, i.e., $C_K(M, T) = |TS_K(M, T)| / |TR_K(M)|$. If $TS_K(M, T) = TR_K(M)$, it is said that T is *K-adequate* for M.

Test coverage criteria are usually defined with specification or fault coverage in mind. When an FSM is the specification for testing, tests covering an FSM specification target one or several elements such as inputs, outputs, states, and fragments of its transition graph. Covering inputs and outputs is usually considered as extremely weak requirements for FSM testing, so we will not consider them in this paper. As to covering fragments of the transition graph, path coverage is a common way of defining coverage.

However, such coverage has to be selective, as the number of paths is infinite in presence of cycles. One of the most cited criteria is the transition coverage, which we consider in this paper. It is a special case of an "x-switch" coverage criterion, proposed in [Chow 1978], which defines a testing requirement as a tuple of transitions to cover by a test; for simplicity we concentrate only on the traditional transition coverage criterion.

Testing with fault coverage in mind relies on implementation fault models. Among simple FSM fault models we should mention initialization faults and transition faults considered in this paper. The former states that the only possible implementation faults are related to a wrong initial state of a specification FSM, while the later assumes that implementations fault occur in transitions.

We thus choose the following four representative FSM test coverage criteria: i) state coverage; ii) transition coverage; iii) initialization fault coverage; and iv) transition fault coverage. These criteria are defined in the next sections.

3.1. State Coverage Criterion

For the state coverage (S) criterion, we assume that reaching a state of the FSM M by some test is a testing requirement. To simplify the presentation, we define $TR_S(M) = S$, though a more general way of defining it is to use a subset of states (to reach and thus to cover by tests) instead of the whole set S. $TS_S(M, T)$ is the set of states that are covered by T, and thus, $C_S(M, T) = |TS_S(M, T)| / |S|$. As an example, for the FSM in Figure 1a a test suite $\{ab, b\}$ is S-adequate; note that the initial state is reachable with the empty transfer sequence, while the prefix a of the test ab is a transfer sequence to state 4.

3.2. Transition Coverage Criterion

For the transition coverage (T) criterion, we assume that covering a transition of the FSM M is a testing requirement. Again, for simplicity, we define $TR_T(M) = D$. $TS_T(M, T)$ is the set of transitions covered by tests in T, i.e., $TS_T(M, T) = \{(s, x) \in D \mid \exists \pi \in T, \alpha x \leq \pi, \delta(s_0, \alpha) = s\}$. Note that, since only initially connected FSMs are considered, each state can be reached and, therefore, each transition can be covered. Thus, $C_T(M, T) = |TS_T(M, T)| / |D|$. If T is T-adequate, then it is easy to verify that T is also S-adequate. Therefore, the transition coverage criterion subsumes the state coverage criterion. The usefulness of this criterion is that a T-adequate test suite detects all output faults in implementations, provided that there are no transfer faults. For our example FSM in Figure 1a a test suite $\{aa, aba, ba, bb\}$ is T-adequate.

3.3. Initialization Fault Coverage Criterion

For the initialization fault (IF) coverage criterion, we define coverage with respect to initialization faults, i.e., the testing requirements address the states that could wrongly be used as the initial state of an FSM implementation. To satisfy such a requirement, a test suite should include a sequence, which is separating for the initial state and a state in question, a suspected initial state, applied to both states. Then, we define $TR_{IF}(M) = \{s \in S \mid s \nsim s_0\}$. Note that $TR_{IF}(M)$ ranges from the empty set for M with no distinguishable states to $S \setminus \{s_0\}$ for a reduced M. The criterion is thus applicable to FSM with at least one pair of distinguishable states. $TS_{IF}(M, T)$ is defined as follows.

$TS_{IF}(M, T) = \{s \in S \mid s \nsim s_0, \exists \pi, \chi \in T, \gamma \leq \pi, \beta\gamma \leq \chi, \delta(s_0, \beta) = s, \lambda(s_0, \gamma) \neq \lambda(s, \gamma)\}$, and thus, $TC_{IF}(M, T) = |TS_{IF}(M, T)| / |TR_{IF}(M)|$.

In this formula, γ is a sequence that distinguishes s_0 from a state s, so the test suite T should contain a test, which starts with γ, as well as a test, which takes the FSM M into the state s and then continues with γ. An IF-adequate test suite should have such tests for each state distinguishable from the initial state. Therefore, for reduced FSMs, a test suite that is IF-adequate is also S-adequate, i.e., the criterion IF subsumes the criterion S. For the example FSM in Figure 1a a test suite {*aa*, *aba*, *bb*} is IF-adequate.

3.4. Transition Fault Coverage Criterion

For a pair $(s, x) \in D$, we define a coverage with respect to transition faults (TF), by considering that the transition from state s under input x in some mutant may have an unexpected output or/and wrongly end in another state distinguishable from $\delta(s, x)$. Thus, the set of testing requirements is defined as $TR_{TF}(M) = \{(s, x, s') \in D \times S \mid \delta(s, x) \not\sim s'\}$. Since $TR_{IF}(M)$ is empty for M with no distinguishable states, the criterion is applicable for FSM with at least one pair of distinguishable states. A more general case would be to consider in $TR_{TF}(M)$ a subset of transitions, each of which may end in some state from a subset of states, similar to the so-called fault function, used in [Petrenko and Yevtushenko 1992]. Thus, a testing requirement is a pair of a transition, represented by the pair (s, x), and a state from which the tail state of the transition should be distinguished. To satisfy such a requirement, a test suite should not only cover a transition as in the case of the transition coverage criterion, but also have a corresponding separating sequences applied in both concerned states. $TS_{TF}(M, T)$ is defined by the requirements that are satisfied:

$$TS_{TF}(M, T) = \{(s, x, s') \in D \times S \mid \delta(s, x) \not\sim s', \exists \pi, \chi \in T, \alpha x \gamma \leq \pi, \beta \gamma \leq \chi, \delta(s_0, \alpha) = s, \delta(s_0, \beta) = s', \lambda(\delta(s_0, \alpha x), \gamma) \neq \lambda(s', \gamma)\}.$$

Thus, $TC_{TF}(M, T) = |TS_{TF}(M, T)| / |TR_{TF}(M)|$. For the example FSM (Figure 1a) a test suite {*aaaaa*, *abaaa*, *baa*, *bbaaa*} is TF-adequate. For reduced FSMs, if a test suite is TF-adequate, then we can prove that it is also IF- and T-adequate. Therefore, TF criterion subsumes criteria IF, T, and, consequently, S, for reduced FSMs, as shown in Figure 2.
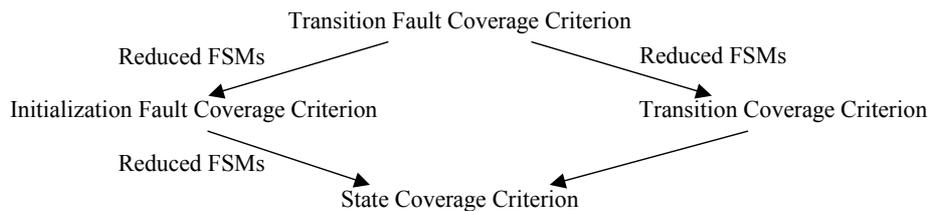

Figure 2. The subsumption relation of FSM coverage criteria.

We note that the idea of this criterion is similar to the one proposed in [Petrenko et. al. 1996], where the fault coverage of a given test suite is defined as the percentage of states that are distinguished from the tail state of each transition by the test suite.

4. Comparing Adequate Tests

The definition of testing strategies requires a careful analysis of the cost and benefits of all applicable coverage criteria. This analysis can be based on the known theoretical properties of the criteria. For instance, one may prefer to choose a criterion most powerful in revealing faults. If, however, the chosen criterion requires an adequate test suite that is unpractical (due to test explosion) or too costly to execute, it will hardly be

chosen. Thus, in many practical situations, the cost of applying a criterion becomes a major factor in choosing a proper test coverage criterion. For simplicity we assume here that the total length of an adequate test suite w.r.t. a given criterion is the cost of the criterion for a given specification FSM.

The upper bounds of test length for the criteria we are considering grow rapidly with the FSM parameters (see discussions below), these bounds characterize the so-called test explosion effect. While at least some of these bounds are shown to be tight, we want to know if the notorious test explosion may occur for a given FSM and for each coverage criterion and, if it does, how big it might be on average compared to what the formulae indicate. Ideally, if an FSM specification is available in a machine-processable form and appropriate FSM test generation tool is easily accessible one would just generate a test suite for each of the candidate coverage criterion and choose the one which corresponds to a desired compromise between test effectiveness and cost. In reality, however, a number of factors can prevent testers from following this simple-minded method. For example, a test strategy may have to be chosen even before a detailed specification is obtained or tools might not always be readily available. Last but not least, one may not need to generate an adequate test suite for a given criterion; he may well restrict himself to, e.g., "90%" of coverage for a certain criterion. In such situations, experimental data, if available, may provide indications on the expected length of test suites 90%-adequate for the chosen coverage criterion.

The upper bounds of the length of tests adequate for various coverage criteria can be derived by considering an FSM with "worst" values of parameters for a given criterion. For the state coverage criterion, such parameter is the accessibility degree r, which is the maximum length of a minimal transfer sequence to a given state; clearly, $0 \le r \le n - 1$. The length of S-adequate test suites does not exceed rn, thus $n(n - 1)$ (note that the formula can further be refined by excluding prefixes of transfer sequences). Similarly, the length of T-adequate test suite is bounded by $kn(r + 1) = kn^2$. For the initialization and transition fault coverage criteria, the distinguishability degree d has also to be taken into account. It may reach the value of $n - 1$ for complete FSMs and $n(n - 1)/2$ for partial FSMs [Petrenko and Yevtushenko 2005]. An IF-adequate test suite may contain $n - 1$ separating sequences applied in the initial state as well as $n - 1$ transfer sequences each of which is followed by a separating sequence. The total length does not exceed $d(n - 1) + (n - 1)(r + d) = (n - 1)(n - 1 + 2d)$. Then, complete FSMs may require up to $(n - 1)(n - 1 + 2n - 2) = 3(n - 1)^2$; while partial FSMs $(n - 1)(n - 1 + 2n(n - 1)/2) = (n + 1)(n - 1)^2$. For the transition fault coverage, a single transition may require at most two tests, each of which does not exceed the value $r + 1 + d$; thus kn transitions need $2kn(r + 1 + d) = 2kn(n + d)$ inputs. Thus, the length of TF-adequate test suite does not exceed $2kn(2n - 1)$ for complete FSMs and $kn^2(n + 1)$ for partial ones.

In addition to the above characterization of worst cases, one may also consider asymptotic characterization of FSM parameters for "almost all FSMs". Indeed, the book [Trakhtenbrot and Barzdin 1973] indicates that the accessibility degree r is asymptotically equal to $\log_k n$ and the distinguishability degree d is asymptotically equal to $\log_k \log_l n$ for complete FSM with n states, k inputs, and l outputs. These formulae give the values expected to be valid for almost all FSMs. We use them to derive the expected length of the test suites for the four criteria. For S-criterion the expected length is $rn = n \log_k n$. For T-criterion, the length is $kn(r + 1) = kn(\log_k n + 1)$. IF-criterion yields (for complete FSMs) $d(n - 1) + (n - 1)(r + d) = (n - 1)(\log_k n + 2\log_k \log_l n)$.

Finally, for TF-criterion, the expected length is $2kn(r + 1 + d) = 2kn(1 + \log_k n + \log_k \log_l n)$.

Table 1. Formulae for the test length for state, transition, initialization fault and transition fault coverage criteria

Coverage Criterion	Maximum Length for all FSMs	Expected Length for almost all (complete) FSMs	Fitted Formulae	Ratios
S	$n(n{-}1)$	$n\log_k n$	$1.31n^{1.07}{-}0.23$	The base
T	kn^2	$kn\log_k n$	$1.17kn^{1.15}{+}6.31$	$T/S{=}0.893kn^{0.08}$
IF	CFSMs: $3(n{-}1)^2$ PFSMs: $(n{+}1)(n{-}1)^2$	$(n{-}1)(\log_k n{+}2\log_k\log_l n)$	$2.65n^{0.96}{-}2.25$	$IF/S{=}2.023n^{-0.09}$
TF	CFSMs: $2kn(2n{-}1)$ PFSMs: $kn^2(n{+}1)$	$2kn(1{+}\log_k n{+}\log_k\log_l n)$	$2.17kn^{1.33}{+}7.34$	$TF/T{=}1.855n^{0.18}$ $TF/IF{=}0.819kn^{0.37}$

At the same time, given a specification FSM and a coverage criterion, it is not clear how close to these bounds the test length might be. Since currently it does not seem plausible to gather sufficient data about actual specifications and tests adequate for various criteria, experiments involving random generation of specifications and tests may provide experimental characterization of how the test length depends on FSM parameters and coverage criteria. The remaining part of this paper is devoted to the experiments addressing the following questions:

- How does the average length of an adequate test suite compare to the upper bound?

- How test suites adequate for various criteria relate in terms of the length?

- Given a test suite adequate for one criterion, how adequate is it for another criterion?

- Which FSM parameters contribute more to test explosion and for which of the four criteria?

5. Experiments Setup

The experiments are based on the following main operations on FSMs and tests: (i) FSM generation; (ii) Generation of a test suite adequate for a given criterion; and (iii) Minimization of a test suite with respect to a given criterion. In the following sections, we explain these operations.

5.1. FSM Generation

We implemented a tool to randomly generate initially connected FSMs with given numbers of states, inputs, outputs, and transitions. The tool first generates sets of states, inputs and outputs with the required number of elements. The generation proceeds then in two phases. In the first phase, a state is selected as the initial state and marked as "reached". Then, for each state s not marked as "reached", the generator randomly selects a reached state s', an input x and an output y and adds to the machine being generated a transition from s' to s with input x and output y, and mark s as "reached". When this phase is completed, an initially connected FSM is obtained. In the second phase, the generator adds, if needed, more transitions (by randomly selecting two states, an input, and an output) to the machine until the required number of transitions is obtained.

There are at least two alternatives to the random generation approach. First, one may involve human testers in experiments by asking them to generate FSMs using their experience and domain knowledge. This setting would allow considering the human factor in the experiments and hopefully obtaining more "realistic" FSM specifications. However, manual generation of a sufficient number of FSMs could be excessively expensive. Another alternative would be to use only FSMs found in the literature, forming a benchmark of FSMs. This setting is attractive, but, again, not many such FSMs are publicly available.

5.2. Test Generation

In order to compare the length of test suites implied by various test coverage criteria, one needs first to generate these tests in a uniform way, as the test length may significantly vary depending on algorithms used for test generation. As an example, to derive a test suite adequate for the state coverage criterion, one may use a graph traversal algorithm or just unfold the transition graph into a spanning tree, obtaining test suites of different lengths. Similarly, there are various algorithms for generating test sequences for the other criteria. One possibility of reducing any impact of using different search algorithms and enforcing the uniformity of test generation with various criteria is to use only one test generation algorithm that yields a test suite adequate for all the test coverage criteria considered. Once such a "super" test suite is obtained, one may then determine a (minimal) subset of this test suite adequate to a given criterion and to compare the lengths of the resulting adequate test suites. This approach is implemented as a two-step procedure: 1) generate a (quasi-minimal) test suite adequate for all the four criteria and 2) minimise it for each criterion.

For a given specification FSM $M = (S, s_0, I, O, D, \delta, \lambda)$, a test suite S-, T-, IF-, and TF-adequate for state, transition, initialization fault and transition fault coverage criteria, respectively, is generated in the following manner. For each pair of states s and s', we determine a shortest distinguishing input sequence, $\gamma_{s,s'}$. Note that, as non-reduced FSMs can also be generated, there may be some state pairs for which no such sequence exists. Then, we determine a minimal transition cover T by building a spanning tree of M. We augment T with the empty sequence. The test suite is initialized with T. Finally, for each $\alpha \in T$, $\delta(s_0, \alpha) = s$, and each $s' \in S$, such that $\delta(s_0, \alpha)$ is distinguishable from s', we include $\alpha\gamma_{s,s'}$ in T. The resulting test suite is n-complete for any reduced FSM M, since the adopted test generation algorithm is in fact the HSI-method [Yevtushenko and Petrenko 1990] developed for reduced FSMs, and the test suite is what we need for our experiments: it is S-T-IF-TF-adequate.

5.3. Test Minimization

Given a specification FSM M and an S-T-IF-TF-adequate test suite T, we need to determine its subsets adequate to state, transition, initialization and transition fault coverage criteria, i.e., S-, T-, IF-, and TF-adequate test suites, respectively.

Thus, the problem of test minimization arises. Given a test suite T and a particular criterion K, we want to find $T' \subseteq T$ such that $TS_K(M, T') = TS_K(M, T)$ and a cost function $w(T')$ is minimized. As a special case, if T is K-adequate, T' is also K-adequate. The cost function can be defined to reflect the cost of applying a given test suite. We define the cost $w(\alpha)$ of a sequence $\alpha \in I^*$ as $|\alpha| + 1$, i.e., the length of α plus the implicit reset symbol used to bring the FSM back to the initial state before applying α. We define $w(R)$

as the sum of $w(\alpha)$ of all sequences $\alpha \in R$, such that α is not a proper prefix of another sequence in R. It is thus assumed that all inputs are of the same cost; though, if needed one can easily diversify cost of inputs.

For the state coverage criterion, we need to find a minimal subset $T' \subseteq T$ that reaches every state of a given FSM. This can be posed as a weighted set-cover problem, where the ground set is the set of states and covering elements are tests (as well as all their prefixes). This problem is known to be NP-complete [Karp 1972]. A natural greedy algorithm can be used to find a near optimal covering set. We start with an empty covering set $T' = \varnothing$. At each step, we pick up a sequence $t \in T \setminus T'$ that is the most cost-effective and include it in T'. The *cost-effectiveness* of a sequence t with respect to T' is defined as the ratio between the cost and coverage increments induced by the inclusion t in T', i.e., $(w(T' \cup \{t\}) - w(T')) / |TS_{SC}(M, T' \cup \{t\}) \setminus TS_{SC}(M, T')|$. For the transition coverage criterion, a similar approach can be followed by replacing the set states S by the set of defined transitions D.

For the initialization fault and transition fault coverage criteria, the test minimization problem cannot directly be cast as a set-cover problem, since to cover some testing requirements two sequences may be needed at same time. In this case, the test minimization problem is defined as a set-cover with pairs (SCP). The SCP problem can be viewed as a generalization of the classical set-cover problem (see [Hassin and Segev 2005] for discussion on its complexity). The authors of [Hassin and Segev 2005] propose a generalization of the greedy algorithm to work with pairs of elements. At each iteration, the cost-effectiveness of single sequences as well as pair of sequences is evaluated and the most cost-effective one is selected (either a single sequence or a pair of sequences). We built a tool solving the set-cover problems which is used to minimize a test suite for each of the four coverage criteria.

6. Experimental Results

In the following sections presents the settings and results from of the experiments we carried out to answer the questions stated in Section 4.

6.1. Average Length versus Upper Bounds

We address here the question: For each criterion, how does the average length of the adequate test suites compare to the upper bounds? The formulae of the upper bounds of the test length (Table 1) contain the major FSM parameter n, the number of states, which is varied in our experiments from three to 20. For each value of n, we generate 1000 initially connected deterministic FSMs with four inputs, four outputs for each of the following degrees of completeness: 0.4, 0.6, 0.8, 0.9, and 1.0. The degree of completeness is the ratio between the defined transitions and the number of possible transitions in a deterministic FSM, i.e., kn. Notice that the degree 1.0 corresponds to complete FSMs. Thus, for each value of n, we generate 5000 FSMs, totaling 90000 FSMs. Figure 3 shows the maximal test length defined by the corresponding formulae for the upper bounds and the average length for state, transition, initialization fault and transition fault coverage criteria. The average length of adequate tests in our experiments is far below the worst-case length. Moreover, we notice that it growths not as fast as the upper bounds suggest. It is thus interesting to determine how the average length grows for the various criteria. For state coverage and initialization fault coverage criteria, we model this growth as a function of the form $f(n) = a\,n^b + c$, where n is the number of states, for some parameters a, b and c. For transition coverage and transition fault

coverage criteria, we model this growth as a function of the form $f(n) = a\,kn^b + c$, where n is the number of states and k is the number of inputs for some constants a, b and c, where k is the number of inputs. Notice that $k = 4$ for all the FSMs we generated in this experiment. The forms of these formulae are chosen to resemble the theoretical upper bound formulae. We use the implementation of the nonlinear least-squares (NLLS) Marquardt-Levenberg algorithm [Bates and Watts 1988] available in the *gnuplot* tool to the values of a, b and c that make $f(n)$ fit best to the collected data. The resulting functions are given in Table 1.
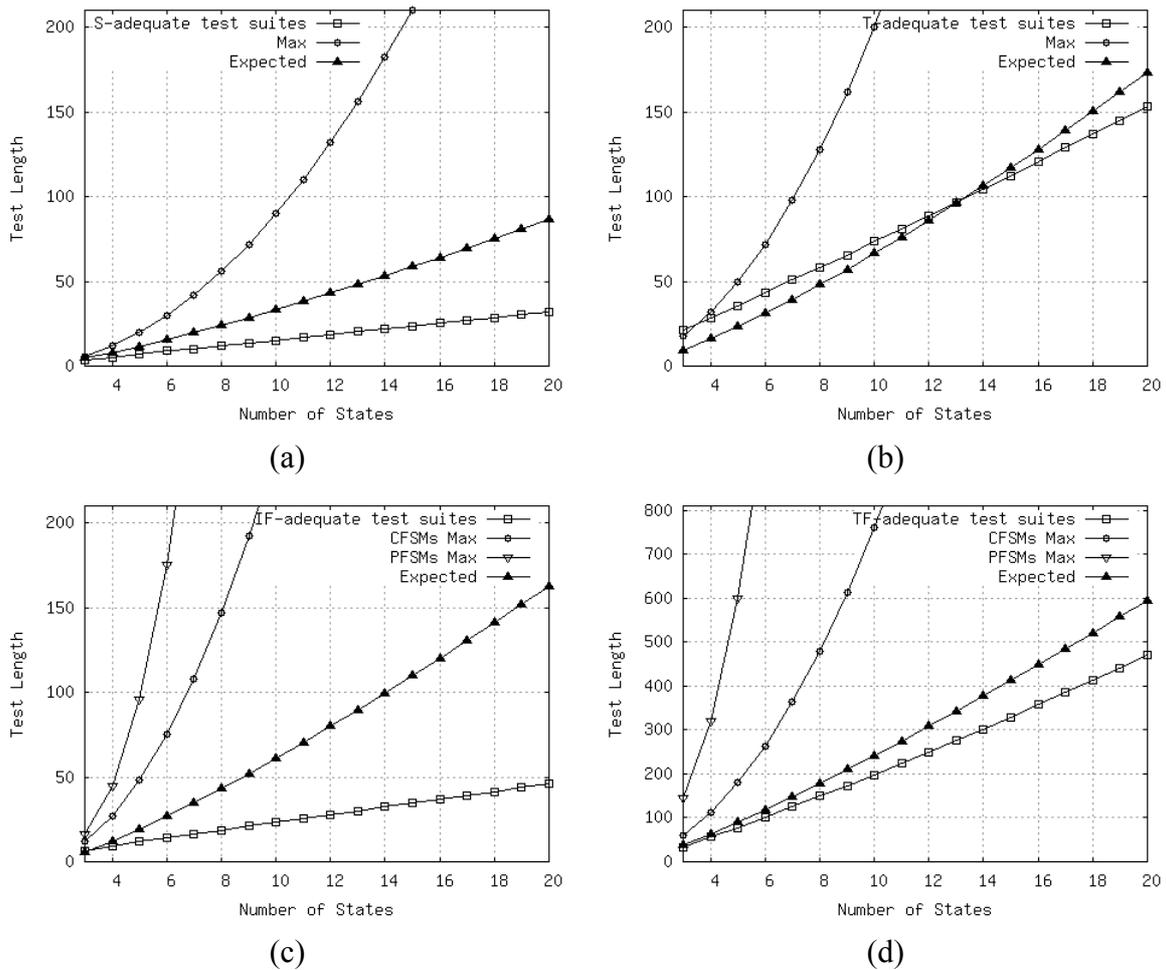

(a) (b)

(c) (d)

Figure 3. Maximum, expected and average lengths of adequate test suites w.r.t. the number of states for: (a) state coverage, (b) transition coverage, (c) initialization fault coverage and (d) transition fault coverage.

The table also contains the computed ratios of the test length, which allow one to estimate the price of changing a coverage criterion in terms of the expected test length increase. As an example, the test length increase by switching from state coverage criterion to transition coverage criterion is approximated by the function $0.893\ kn^{0.08}$. Thus, all things being equal, a T-adequate test suite is roughly 3.5 times larger then an S-adequate test suite (recall that $k = 4$ for the FSMs we considered) while the number of states has a marginal impact. Notice that in the ratio between IF-adequate and S-adequate tests $2.023\ n^{-0.09}$, the impact of the number of states is negative, which implies that the difference in the length of test suites for both criteria tends to decrease as the number of states increases.

6.2. Criteria Relative Strength

Addressing our third question "Given a test suite adequate for one criterion, how adequate is it for another stronger criterion?", we determine the coverage of a test suite adequate for one criterion w.r.t. other criteria. We randomly generate 5000 FSMs with two inputs, two outputs, number of states ranging from three to 20, and the degrees of completeness 0.4, 0.6, 0.8, 0.9 and 1.0. An adequate test suite is obtained for each FSM and criterion and then its coverage for the other criteria is determined. For instance, given an S-adequate test suite, we calculate the percentage of covered transitions. Table 2 shows the relative strength of the four criteria. We present both the average and the standard deviation. For instance, we can observe that the average coverage of a T-adequate test suite on average 0.928 of the testing requirements of the initialization fault coverage criterion, with the standard deviation of 0.122. Notice that, as we generated both reduced and unreduced FSMs, some test suites that are adequate for transition fault coverage criterion are not adequate even for state coverage, since there may exist some states that are not distinguishable from any other states. In this case, transition fault coverage criterion does not require to cover all the states. However, as expected, test suites adequate to this criterion are almost always adequate to any of the other criteria.

Table 2. Relative strength of FSM coverage criteria.

	S	T	IF	TF
S	-	1.000/0.000	0.970/0.064	0.994/0.034
T	0.679/0.132	-	0.772/0.107	0.989/0.047
IF	0.645/0.248	0.928/0.122	-	0.993/0.053
TF	0.299/0.182	0.691/0.134	0.478/0.171	-

6.3. FSM Parameters

Addressing the question: "Which FSM parameters contribute more to test explosion and for which of the four criteria?", we investigate the effect of various FSM parameters on the length of the test suites for the four criteria. We observe that the impact of the number of states is essential, as discussed in Section 5.2. Here we are interested in other parameters that characterize an FSM, namely, the number of inputs, outputs, and transitions.

Figure 4a shows how the test suite length varies with the number of inputs. We generate FSMs with ten states, two outputs, the number of inputs ranging from two to seven and the degrees of completeness 0.4, 0.6, 0.8, 0.9 and 1.0 (100 FSMs for each setting, totaling 3000 FSMs). The obtained data indicate that, w.r.t. the number of inputs, the test length grows almost linearly for transition and transition fault coverage criteria. At the same time, the number of inputs does not impact the test length for state and initialization fault coverage criteria.
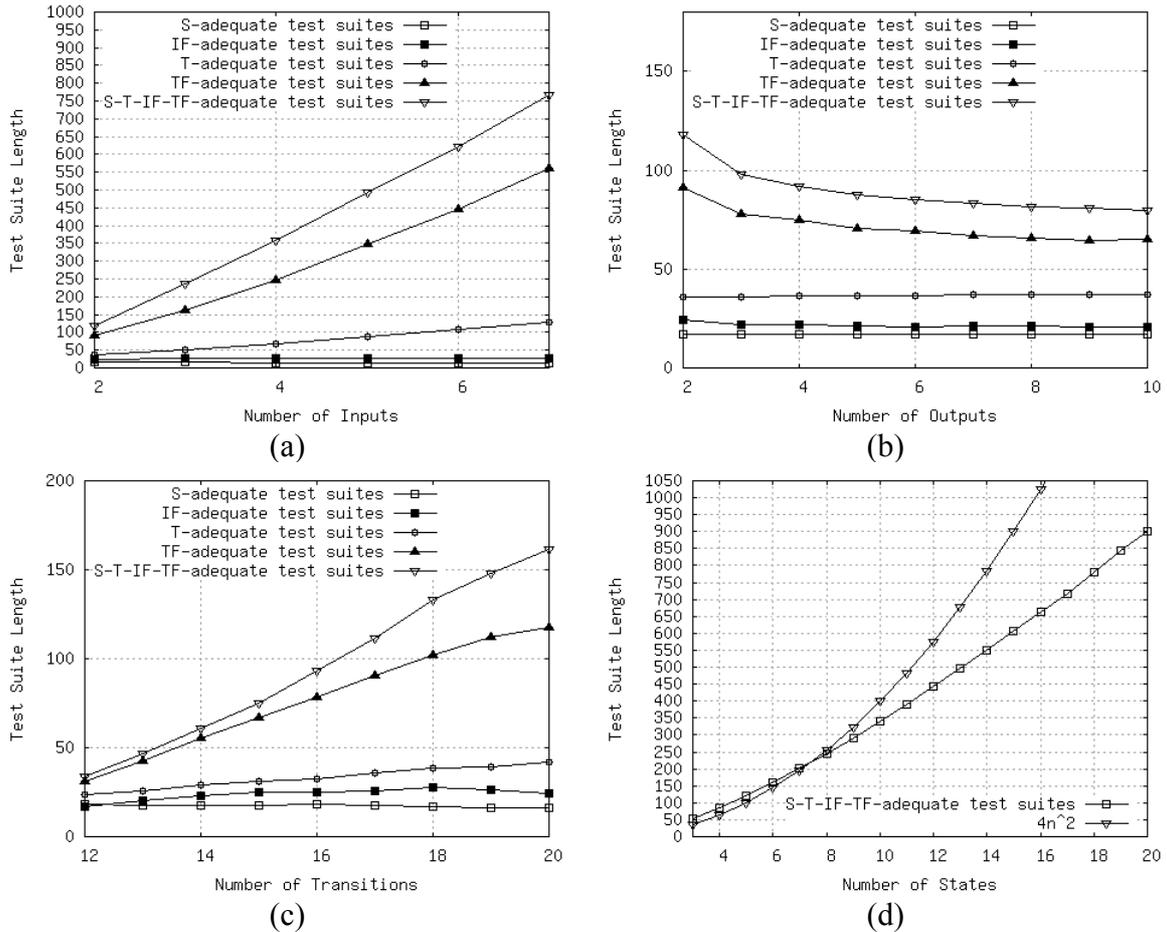
Figure 4. Average length of adequate test suites for each coverage criteria w.r.t. (a) the number of inputs; (b) the number of outputs; (c) the number of transitions. (d) the average length of S-T-IF-TF-adequate test suites versus the curve $4n^2$

Figure 4b shows how the test length for considered criteria depends on the number of outputs. We generate FSMs with ten states, two inputs and the number of outputs ranging from two to ten and the degrees of completeness 0.4, 0.6, 0.8, 0.9 and 1.0 (100 FSMs for each setting, totaling 4500 FSMs). We observe that, as expected, the test length for state and transition coverage criteria does not significantly depend on the number of outputs. On the other hand, the length of tests adequate for transition fault coverage criterion decreases when the number of outputs increases. The reason is that the length of separating sequences tends to decrease if an FSM has more outputs. Accordingly, the length of test suites for criteria that rely on separating sequences tends to decrease as well. Although the length of a test suite adequate for the initialization fault coverage criterion which also uses separating sequences should also depend on the number of outputs, its impact on the length is negligible in the performed experiments.

Figure 4c shows how the test suite length varies with the number of transitions. Recall that, for fixed numbers of states and inputs, the number of transitions determines the completeness degree of the FSMs. We generate FSMs with ten states, two outputs, two inputs, and with the number of transitions ranging from 12 to 20 (100 FSMs for each setting, totaling 900 FSMs). We observe that the test length for state and initialization fault coverage criteria does not vary, while for transition and transition fault coverage criteria grows quasi-linearly.

Dorofeeva *et al.* [2005b] point out that the length of test suites generated by Wp, HSI, UIOv and H methods is of the order $4n^2$. These methods generate *n*-complete test suites. In our experiment, we generated S-T-IF-TF-adequate test suites, which are also *n*-complete for reduced FSMs. We expected that the test suite lengths for the S-T-IF-TF-adequate test suites were also of the same order. In Figure 4d, we present the average length of S-T-IF-TF-adequate test suites and the curve $4n^2$. For each value of *n*, the average is computed over test suites generated for 900 complete reduced FSMs with four inputs and four outputs, totaling 16200 FSMs. In Dorofeeva et al's experiments, 1100 complete reduced FSMs are generated with the FSM parameters different from ours, in particular, the number of states ranges from 30 to 100 and the number of inputs and outputs from six to ten. Although the different settings hinder the comparison of obtained data, we observe that our experimental data do not confirm Dorofeeva et al.'s conclusion. We fitted the data to the $f(n) = a\, n^b + c$ with nonlinear least-squares and obtained 13.01 $n^{1.418} - 3.697$. The data suggest that the length of *n*-complete test suites in our experiments grows slower than $O(n^2)$. However, this observation must be checked with more experiments.

7. Threats to Validity

There are several caveats in interpreting experimental results that must be noted:

1. As explained in Section 5.1, to ensure that only initially connected FSMs are generated, initially a tree FSMs with the required number of states and the minimal number of transitions is first randomly generated and then more transitions are added. This procedure tends to generate FSMs in which the states with a lower accessibility degree may have more defined transitions than the states with a higher accessibility degree, especially for partial FSMs with few transitions. As the number of transitions increases, the transitions tend to be more normally distributed. A possible approach that could be used to bypass this problem would be to randomly generate an FSM, and then check whether it is initially connected. However, this approach does not look practical, since the probability of generating an initially connected FSM by a random FSM generator is not high.

2. As previously stated, in order to not bias a test suite by test generation methods, we use a single method to generate test suites that are adequate to all the considered criteria and then minimize them using the same minimization method, solving a set cover problem. Another approach that could be tried here is to generate tests using several alternative techniques for obtaining tests adequate for a given criterion and to consider an average test length. For instance, we may generate a transition coverage adequate test suite by determining a transition tour or by determining a spanning tree of the graph underlying the FSMs. However, the comparison would still be biased by the methods selected for generation.

3. In the main algorithm for generating tests, for each pair of states, we determine in advance a shortest separating sequence, which is used throughout the algorithm. This approach is similar to traditional test generation methods, such as W, Wp, and HSI. However, Dorofeeva *et al.* [2005a] demonstrate that shorter test suite can be obtained if the separating sequences are determined on-the-fly. If shorter tests may thus be generated, test suites adequate for the transition fault and initialization fault coverage criteria may also be shorter than the ones obtained in our experiments. Even if the charts for the test length may further be refined, the obtained characterization of adequate tests and their ratios may well persist.

4. The test minimization is a computationally hard problem. Therefore, approximation algorithms based on greedy approaches were employed, as a result, the minimized test suites are not guaranteed to be minimal. The replication of these experiments with another minimization algorithm may allow to factor out the impact of a minimization algorithm on the adequate test length.

8. Conclusion

This paper is devoted to experiments with common coverage criteria used for FSM-based testing. We have developed an approach for generating tests adequate for each of the criteria in such a way that the results do not significantly depend on methods used for test generation from randomly generated FSMs. The idea is first to generate a test suite which is adequate for all the considered criteria and them minimize it for each criterion separately using a single test minimization algorithm which solves a combinatorial set cover problem. The prototype tool environment developed for the experiments has a much wider application area, as it can be used to actually generate tests adequate for various test coverage criteria.

The obtained experimental data shed some light on the expected length of test suites adequate to state, transition, initialization, and transition fault coverage criteria. In particular, the experiments show that, as expected, the tests are much shorter than the upper limits suggest. Moreover, the average length of test suites grows much slower than the corresponding formulae suggest. For instance, the length of test suites adequate to transition fault coverage criterion are of the order $O(kn^{1.33})$, which is lower than the theoretical $O(kn^3)$. The formulae for the expected length of the test suites for the four criteria which we derived using some known (though rarely used) results on asymptotic characterization of FSMs give values much closer to experimental data than worst-case estimations. We also compared the relative strength of the criteria. As transition fault coverage criterion subsumes transition and initialization fault coverage criteria only for reduced FSMs, the experimental results suggest that, even for unreduced ones, test suites adequate to transition fault cover, on average, cover about 99% of the requirements of transition and initialization fault coverage criteria.

The experiments confirmed that the number of states has the greatest impact on the length of the test suites adequate to all criteria. The number of inputs influences almost linearly the length for transition and transition fault coverage criteria. At the same time, the number of inputs does not impact the test length for state and initialization coverage criteria. An increase in the number of outputs does not lead to an increase in the test length for state and transition coverage criteria. On the other hand, the test length for initialization fault and transition fault coverage criteria tends to decrease with the growth in the number of outputs, due to the resulting shortening of separating sequences. As expected, the number of transitions has a nearly linear impact on the test length for transition and transition fault coverage criteria, while no sensible influence on the length of tests adequate for state and initialization fault coverage criteria. Our experimental data also suggest that the length of n-complete test suites increases more slowly than $O(n^2)$, as concluded in a previous work. However, as the parameters of the FSMs generated in experiments differ, more experiments are necessary to draw a more definitive conclusion.

We need to conduct more experiments also to refine the formula for estimating the test length we suggested. In our fitted formulae, we only allow the variation of the number of states, using a fixed number of inputs. It would be interesting to find fitted formulae that include both variables. We also intend to assess the variation in the test

suite length w.r.t. other FSM parameters, such the accessibility degree, distinguishability degree, and distinguishability ratio. It would be interesting to try to enrich the experimental data using more realistic data obtained with the help of testers, for example, along with random generation of FSMs and test suites, one could consider FSMs and test suites manually built by the testers.

References

Bates, D. M. and Watts, D. G. (1988). *Nonlinear Regression and Its Applications*. Wiley.

Bochmann, G. v., Petrenko, A., Protocol Testing: Review of Methods and Relevance for Software Testing, In *ACM International Symposium on Software Testing and Analysis (ISSTA'94)*, USA, 1994.

Binder, R. (2000). *Testing Object-Oriented Systems*. Addison-Wesley, Inc.

Chow, T. S. (1978). Testing software design modeled by finite-state machines. In *IEEE Transactions on Software Engineering*, 4(3):178–187.

Dorofeeva, R., El-Fakih, K., and Yevtushenko, N. (2005a). An improved conformance testing method. In *Formal Techniques for Networked and Distributed Systems*, volume 3731 of *Lecture Notes in Computer Science*, pages 204–218. Springer.

Dorofeeva, R., Yevtushenko, N., El-Fakih, K., and Cavalli, A. R. (2005b). Experimental evaluation of fsm-based testing methods. In *Third IEEE International Conference on Software Engineering and Formal Methods (SEFM 2005)*, pages 23–32. IEEE Computer Society.

Frankl, P. R., Weyuker, E. J., (1993). A Formal Analysis of the Fault-Detecting Ability of Testing Methods. In *IEEE Transactions Software Engineering* 19(3):202-213.

Hassin, R. and Segev, D. (2005). The set cover with pairs problem. In *Proceedings of the 25th Annual Conference on Foundations of Software Technology and Theoretical Computer Science*, pages 164–176.

Karp, R. M. (1972). Reducibility among combinatorial problems. In Miller, R. E. and Thatcher, J. W., editors, *Complexity of Computer Computations*, pages 85–103.

Moore, E. F. (1956), Gedanken-experiments on sequential machines, In *Automata Studies, Annals of Mathematics Series*, No. 34, pages 129–153.

Petrenko, A., Bochmann, G. v., and Yao, M. (1996). On fault coverage of tests for finite state specifications. In *Computer Networks and ISDN Systems*, 29(1):81–106.

Petrenko, A. and Yevtushenko, N. (1992). Test suite generation for a fsm with a given type of implementation errors. In *Proceedings of the IFIP 12th International Symposium on Protocol Specification, Testing, and Verification*, pages 229–243.

Petrenko, A. and Yevtushenko, N. (2005). Testing from partial deterministic fsm specifications. *IEEE Transactions on Computers*, 54(9):1154–1165.

Trakhtenbrot, B. A. and Barzdin, Y. M. (1973). *Finite Automata, Behaviour and Synthesis*. North-Holland.

Yevtushenko, N. and Petrenko, A. (1990). Synthesis of test experiments in some classes of automata. In *Automatic Control and Computer Sciences*, 24(4):50–55.