19

Handling redundant and additional states in protocol testing

A. Petrenko[1], T. Higashino[2], and T. Kaji[2]

1 - Université de Montréal,
C.P. 6128, succ. Centre-Ville, Montréal, H3C 3J7, CANADA,
Phone: (514) 343-7535, Fax: (514) 343-5834,
petrenko@iro.umontreal.ca

2 - Osaka University,
Toyonaka, Osaka 560, JAPAN
Phone: +81-6-850-6607, Fax: +81-6-850-6609
higashino@ics.es.osaka-u.ac.jp; t-kaji@ics.es.osaka-u.ac.jp

Abstract

This paper addresses the problem of conformance testing of protocols modeled by FSMs with redundant states. Redundant states appear in an FSM which may be nonminimal or nonconnected. The existing test derivation methods usually are not directly applicable to these machines. In this paper, we show that they can be adjusted to cover this class of FSMs and that the traditional assumption on the minimality of machines is not necessary. Another problem with redundant states is that they can cause the appearance of additional states in protocol implementations whose guaranteed detection requires tests of an exponential length. This paper proposes techniques for deriving tests for FSMs with redundant or additional states such that a high fault coverage is achieved while maintaining an acceptable test suite length. The effectiveness of the proposed methods has been evaluated in an experimental way using a benchmark protocol.

Keywords

Conformance testing, FSMs, redundant and additional states, test derivation, fault coverage

1 INTRODUCTION

Conformance testing of a protocol is typically a black-box testing, i.e. it is based on its specification. Formal methods for deriving conformance tests are widely recognized as being capable of producing tests with a high fault coverage [BPY94]. To apply such a method, one usually abstracts a relevant formal model from the available specification of the protocol. For our purposes, we will consider here the FSM-based test derivation methods. They require a single FSM that models the behavior of a given protocol and satisfies certain conditions for

A. Cavalli et al. (eds.), *Protocol Test Systems VIII*

their applicability. In particular, the classical model of completely specified, strongly connected, deterministic and minimal FSMs has been regarded in the testing literature (see, e.g. [SiLe89], [LeYa94], [Ural92]) as the most tractable model for test derivation. A complex protocol is rarely given as a single pure FSM with the properties requested by these methods. The protocol may be given, for example, in the form of a single extended FSM or several communicating (pure or extended) machines. FDT's, such as ESTELLE, LOTOS and SDL, produce modular specifications from which an FSM can be abstracted for each module. To test a system of such modules, it is necessary to specify the behavior of the system as one FSM. Thus, a global FSM is constructed from the given system of component FSMs or EFSMs, using a method such as reachability analysis or unfolding. The resulting FSM may not yet be tractable for test derivation, as it may contain equivalent states and states unreachable from the designated initial state. Even if a single FSM is directly derived from a semi-formal description of the protocol, it may still have redundant states. Quite often, equivalent states are intentionally introduced in order to increase the readability of the specification. If we construct an FSM to model a certain functionality of a complex protocol, then states which were distinguishable in the system as a whole may become equivalent in the obtained FSM. Moreover, states reachable in the system might not be reachable from the chosen initial state. However, connected minimal machines with no redundant states are typically accepted for test derivation. Therefore the original FSM with redundant states is customarily replaced by its connected minimal form.

We observe that the traditional approach explained above leads to the loss of structural information about the behavior of redundant states which are an apparent source of additional states in implementations. This information can be used to elaborate alternative techniques for deriving tests directly from an FSM containing redundant states. To the best of our knowledge, there is no systematic procedure for directly treating this class of machines, except by using an exhaustive procedure.

In Section 2, we introduce some basic definitions and concepts. In Section 3, we consider the class of non-reduced connected FSMs and show that the commonly used assumption on the minimality of the specification machine is no longer necessary and the existing test derivation methods can be generalized to cover this class of FSMs. Based on the structural information, we also develop an alternative technique for FSMs with redundant reachable states which may become additional states in the implementations. The case where redundant states are not reachable in the given machine is analyzed in Section 4, and a proper technique for test derivation is proposed. The techniques elaborated in Sections 3 and 4 are applied in Section 5 to a simple protocol, called INRES. Experimental measurements of the fault coverage of various test suites for this protocol are reported in this section as well. Section 6 discusses how the ideas presented in this paper can be incorporated into existing test derivation methods to ameliorate the fault coverage of conformance tests when the possibility of additional states is allowed for. We conclude by presenting some open research issues.

2 PRELIMINARIES

Let $A = (S, X, Y, \delta, \lambda, s_0)$ be an initialized FSM with m states from which we are required to derive a test suite. Here S, X, and Y are finite and nonempty sets of states, inputs and outputs, respectively; δ: $S \times X \rightarrow S$ is the transition function; λ: $S \times X \rightarrow Y$ is the output function; s_0 is the initial state. We thus consider here only completely specified, i.e. complete, deterministic machines. The usual extensions of the two functions from input symbols to strings (sequences) will be used to specify the behavior of these machines. The two properties of a machine, namely, its minimality and connectedness are important for test derivation.

A finite state machine is *reduced* or *minimal* if no two of its states are equivalent. Every complete deterministic FSM possesses a unique (up to the isomorphism) minimal form. Minimality of the machine provides the possibility of identifying states during testing. In particular, a so-called *characterization set* W is a set of input sequences of the minimal machine which tells every two states apart [Koha78].

A machine is *initially connected* if all of its states are reachable from the initial state, i.e. there exists a transfer sequence α such that $\delta(s_0,\alpha) = s$ for every $s \in S$; $\delta(s_0,\varepsilon) = s_0$, where ε is the empty sequence. It is *strongly connected* if every state is reachable from any other state. Initially connected FSMs are usually considered for test derivation in the case where the reliable reset is available in the implementations, whereas the strongly connected FSMs are used in other cases. Initially or strongly connected machines will simply be referred to as *connected* machines. A connected machine possesses a so-called *state cover V* which is a set of transfer sequences, usually one shortest sequence per state. A state cover is used to check all the transitions from every state of the machine.

An implementation FSM I is assumed to be a complete initialized machine with the input alphabet X of the specification FSM A. One must check the equivalence of the two machines I and A by testing I as a black-box. The equivalence of machines is defined as the equivalence of their initial states. I is viewed as a black box, and any test suite TS which is a set of input sequences, can neither distinguish equivalent states in the FSM I, nor bring I into a state unreachable from its initial state. Therefore, it makes sense to assume that any implementation is a complete connected minimal machine $I = (T, X, Y, \Delta, \Lambda, t_0)$.

Let $\mathfrak{I}$ be a certain set of FSMs with the input alphabet X of the FSM A. A test suite TS is said to be *complete for A in the class* $\mathfrak{I}$ if for every machine I from this set which is not equivalent to A, there is an input sequence α in TS such that the corresponding output sequences of A and I are different, i.e. $\lambda(s_0,\alpha) \neq \Lambda(t_0,\alpha)$ for any $I \neq A$, $I \in \mathfrak{I}$. In this case, we also say that the test suite guarantees the complete fault coverage in the class $\mathfrak{I}$. There are several ways in which this class can be specified [PBD93]. As an example, consider the set of FSMs which differ from the FSM A in their output functions only. Any test suite covering all of the transitions of A (a transition tour) is complete in this class. Another example constitutes the universal set $\mathfrak{I}_m$ of all FSMs with at most m states. The test suite complete in the class $\mathfrak{I}_m$ is simply called *m-complete* [BPY94]. M-complete test suites were first introduced as checking experiments [Moor56], [Henn64]. Since then much of the research in this field has focused on deriving m-complete test suites from connected minimal FSMs (for a recent survey, see [BoPe94]). Note that other classes of completely specified deterministic FSMs have not been studied systematically in the testing theory. As a result, there is no systematic procedure for test derivation which is directly applicable to a machine with unreachable and/or equivalent states.

If a specification machine happens to be nonconnected or nonminimal, then the machine should be transformed into its equivalent form, connected and minimal, before one can apply the currently existing methods of test derivation. Unreachable states are hence deleted and the machine is minimized. We say that the specification FSM has *redundant* states if it does not coincide (up to the isomorphism) with its connected minimal form. A redundant state can be either reachable or unreachable from the initial state.

The problem with the redundant states in the specification is that they could be a source of additional faults. In particular, a redundant state, reachable in the specification FSM and equivalent to another state, can become a new distinct state in an implementation machine. A state unreachable in the specification FSM can become reachable in the implementation machine. Due to these faults, additional states emerge in implementations. An implementation FSM has *additional* states if its connected minimal form has more states than the connected minimal form of the corresponding specification. Keeping in mind that two connected minimal FSMs, if equivalent, have the same number of states, we can easily prove that an implementation FSM with additional states is not equivalent to the specification FSM. Faulty implementations with additional states may escape from detection by tests when the original specification FSM is replaced by its connected minimal form. The replacement results in the loss of the structural information in the specification about the behavior of redundant states which are an apparent source of additional states in implementations. This structural information could offer new ways for deriving test suites with a high fault coverage at reasonable cost. In next sections, we first examine a commonly used assumption on the

minimality of the specification machine and then propose several techniques for deriving tests from FSMs with redundant states based on this structural information. The experimental results reported later in this paper confirm the viability of our approach.

3 FSMS WITH REDUNDANT REACHABLE STATES

3.1 *M*-complete test suites

Assume that an FSM A is connected, but not minimal (non-reduced). As mentioned above, all currently existing formal methods for test derivation require that the complete FSM is minimal. This limitation implies that before tests are derived, the given non-reduced machine should undergo the state minimization procedure. The latter is a classical problem, and there is a suitable algorithm [Hopc71] with complexity $O(pm \log m)$, where p is the number of inputs and m is the number of states. While the procedure is relatively simple, the question still arises as to the necessity of this step and thus on the necessity of the assumption of minimality itself. We first recall the basic techniques for deriving a q-complete test suite, where q is not less than the number of states in the minimal form.

Suppose that a minimal form of the FSM A, i.e. a reduced FSM B with n states ($n<m$), is obtained. It now becomes possible to systematically derive a q-complete test suite by applying one of the existing methods to the FSM B. We may use, for example, a method such as the W-method [Vasi73], [Chow78]. Like all the others, this method requires the specification machine to be minimal. It also implies that the user previously estimates an upper bound q on the number of states in the minimal form of the implementation machines.

The most widely used assumption suggests that this bound coincides with the number of states in the minimal machine from which tests are produced, i.e. $q = n$. To derive from B an n-complete test suite based on this assumption, we find its state cover V_B and characterization set W. A state cover contains exactly n transfer sequences, one per state. Let X^k be a set of all input sequences which have length up to k, in particular, $X^1 = X \cup \{\varepsilon\}$. By concatenating sequences of the three sets, V_B, X^1 and W, a test suite $TS_B = V_B X^1 W$ is produced. TS_B is proven to be n-complete for the FSM B, i.e. it is complete in the class $\mathfrak{I}_n$ [Vasi73]. We shall demonstrate that a similar result can be obtained directly from the original specification FSM A with a slightly modified version of the W-method. In other words, we show that the state minimization is not an obligatory step of the test derivation.

To this end, we extend the classical notion of a characterization set to cover non-reduced FSMs. A set W of input sequences is said to be a *characterization set* of the FSM A if it distinguishes any two non-equivalent states. The W set induces a partition $\Pi(W)$ of the set S of states into the equivalence classes. $|\Pi(W)| = n$. $\Pi(W)$ is an equivalence partition of A [Gill62]. If A is reduced, then we have the traditional notion of the characterization set, as used for example, in [Koha78], [Vasi73], [Chow78], [FBK91].

Next we define a *class cover* V_C of A as a set of transfer sequences leading A to every class of $\Pi(W)$ from its initial state. $|V_C| \geq |\Pi(W)|$. If A is reduced, then the V_C set is its state cover in the usual sense. It is always possible to choose exactly $|\Pi(W)|$ transfer sequences as a class cover.

Similarly to the original W-method, we concatenate the three sets: V_C, X^1, W, to obtain a test suite $V_C X^1 W$ for the FSM A. There exists a one-to-one mapping between the equivalence classes of $\Pi(W)$ and the states of the minimal form B. Then the class cover V_C is a state cover of B. The characterization sets of A and B coincide, since the equivalent states cannot be distinguished by any input sequence. Thus, we have the following proposition.

Proposition 3.1. The test suite $V_C X^1 W$ is n-complete for the FSM A with m states, where n

is the number of states in the minimal form of A.

We illustrate the idea of deriving tests directly from a non-reduced machine using an FSM A shown in Figure 1 as an example. State 1 is the initial state of the FSM A.

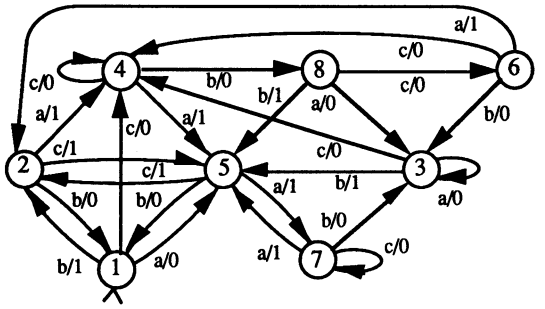

Figure 1 The FSM A.

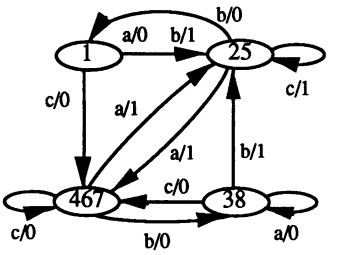

Figure 2 The minimal form B of A.

First, we find a characterization set W of A by constructing a distinguishing tree, as explained in textbooks [Koha78], [Gill62]. It is sufficient to consider only input sequences of length up to m-1 = 7. This is because two states not distinguished by all input sequences of this length are equivalent [Gill62]. We obtain just a single sequence ac, i.e. W={ac}. It induces the equivalence partition $\Pi(W)$ = {1; 2,5; 3,8; 4,6,7}. This partition implies the minimal form of A with four states shown in Figure 2. The sequence ac is a distinguishing sequence for the FSM B. However, to derive a test suite complete in the class $\mathfrak{I}_4$, we make no use of the FSM B, in contrast to the W-method in its original form.

Next, we construct a class cover of the FSM A. The minimal set V_C = {ε, a, c, cb} can be chosen. It is easy to see that this set is also a state cover of B.

The final test suite is:

$V_C X^1 W$ = {ε, a, c, cb}{ε, a, b, c}{ac} =
{$aaac$, aac, $abac$, $acac$, bac, $caac$, cac, $cbaac$, $cbac$, $cbbac$, $cbcac$, $ccac$}.

This very test suite might also be produced from the FSM B by applying the W-method to this machine. If there is only single minimal state cover of B then the two methods always yield the same result.

Next, we note that a test suite complete w.r.t. the number of states in the minimal form of the given specification A might not be the best possible solution from the viewpoint of fault coverage. The problem is that an implementation under test may have been derived from the original (non-reduced) specification and not from its minimal form used for test derivation. Since the original specification has more states than its reduced form, the implementation may also have up to m distinct states, assuming that no state was split into several states during the implementation process. In this case, a test suite $V_C X^1 W$ no longer guarantees the detection of faulty implementations with additional states. In fact, this test suite does not even cover all of the transitions in the original machine. Consider the above example. By executing the test suite $V_C X^1 W$ against the FSM A we find that five transitions are not covered. It means that even a trivial output fault in any uncovered transition goes undetected. The conclusion is that if the implementation is known to be derived from the original specification, then a test suite should foresee the possible appearance of additional states in it. The problem of deriving complete tests in the case where the number of extra states in implementations is unknown is usually considered undecidable [Gill62]. The original FSM provides at least the upper bound m for the

number of states in implementations under the assumption that no state of the specification was split into several states during the implementation process. However, this information is usually lost once a non-reduced FSM is substituted by its minimal form for further test derivation.

Under the above assumption, an m-complete test suite for the minimal FSM with n states has to be derived. It takes the following form, according to the W-method:

$$TS_m = V_B X^{m-n+1} W,$$

where V_B is a state cover, and W is a characterization set of the FSM B. As already discussed, the method in its original form, explicitly calls for the state minimization process.

However, an m-complete test suite can be directly obtained from the given non-reduced FSM. In fact, a statement similar to the Proposition 3.1 can be proven for the class $\mathfrak{I}_q$ with an arbitrary q, $q \geq n$, as well. The structure of such test suite is again based on a characterization set and a class cover of A, but not on that of its minimal form. The missing parameter n is easily deduced from the partition $\Pi(W)$, as this partition is constructed whenever a W set is being derived from the given machine.

In our example, $m = 8$, $W = \{ac\}$, $n = |\Pi(W)| = 4$, $m\text{-}n+1 = 5$, $V_c = \{\varepsilon, a, c, cb\}$ give the test suite

$$TS_8 = \{\varepsilon, a, c, cb\}X^5\{ac\}.$$

Thus, there is no need to transform the original specification into its minimal form if tests are to guarantee complete fault coverage with respect to either the number of states in the minimal form of the specification or any higher limit, including the number of states in the original specification. In fact, the construction of both the minimal machine and the characterization set rely on a common technique for deciding state distinguishability. It is sufficient to apply the technique only once, thus avoiding an unnecessary preprocessing of the specification FSM.

The presented approach yields test suites with guaranteed coverage for the same classes of implementations as the traditional approach based on the W-method. The proposed adjustment of the W-method extends its applicability to the class of complete non-reduced FSMs. Both minimal and non-reduced machines are then treated in a unified way.

3.2 Test suites complete in a special class of FSMs

If the implementations have up to m-n additional states, then a resulting test suite of the structure $VX^{m-n+1}W$ suffers from exponential growth with respect to m-n. The sequences in the set X^{m-n} provide an exhaustive search for potential additional states in an implementation at the penalty of a sharp increase of test size. Regardless of a method used for test derivation, the universal traversal sequences X^{m-n} have to be incorporated into m-complete test suites [Vasi73], [YaLe91].

In our example, $m = 8$, $n = 4$, and $X^{m-n+1} = X^5$. The number of sequences in X^5 is equal to 364. As a result, the number of test cases in the test suite TS_m for the FSM A (Figure 1) is about eight hundred, and the total length of this test suite is well into the thousands. Being theoretically correct, this solution would hardly be accepted in practice. The exponential growth in lengths of the resulting tests precludes the use of such methods on most real protocols. Typically protocols have dozens of various protocol data units and abstract service primitives. Thus the number of inputs $|X|$ in the corresponding FSM model would immediately trigger the explosion of any test incorporating the universal traversal sequences X^{m-n}. This observation motivates us to look for an alternative solution.

Test suites derived to cover additional states rely on a worst-case assumption typical of automata theory. This assumption uses the minimal form, but not the given specification machine itself. In fact, just a single parameter, the maximal number of states, is used by this

assumption. The structural information contained in the original machine is completely ignored. This information indicates reachability of the equivalent states in the machine which are more likely to be implemented as additional states in a faulty implementation. It is natural to expect that an alternative assumption based on this structural information could yield shorter tests with a fault coverage which is acceptable from a practical standpoint. In this section, we try to elaborate such an assumption on reachability of additional states in implementations, and to find a test derivation method for the non-reduced FSMs based on this assumption.

Let V be an arbitrary state cover of the given possibly non-reduced connected FSM A with m states, $|V| \geq m$. By $\mathfrak{I}_V$ we denote a set of all FSMs for which the set V is a state cover. In other words, we assume that every state of a machine in this class is reachable from its initial state with a proper sequence in the set V. It is clear that the number of states in any machine of this class is not more than $|V|$. $|V| \geq n$, where n is the number of states in the minimal form of A, thus $\mathfrak{I}_V \supseteq \mathfrak{I}_n$.

Proposition 3.2. A test suite $TS_A = VX^1W$ is complete for the FSM A with the state cover V and the characterization set W in the class $\mathfrak{I}_V$.

Proof. Consider an FSM I of the class $\mathfrak{I}_V$. If it fails the test suite TS_A, then it is not equivalent to A. Assume therefore, that it passes this test. We must show that I is equivalent to A.

The machine I belongs to the class $\mathfrak{I}_V$. Then by the assumption, all of its states are reachable from the initial state with certain sequences of V. Let the set of its states be T. We have $|T| \leq |V|$. The sequences of the W set are applied to each of the states of I. Then the W set induces a partition of the set T of states in I into the subsets of states which produce the same output reaction in response to W. We denote this partition by $\Pi_I(W)$. It is clear that $|\Pi_I(W)| = |\Pi_A(W)|$, where $\Pi_A(W)$ is the equivalence partition of A. We have $VX^1W = VW \cup VXW$. I passes the test suite, then the FSMs A and I produce the same output sequences in response to the set of input sequences VW. To prove the equivalence of A and I, it remains to demonstrate that $\Pi_I(W)$ is an equivalence partition of I. In this case, the minimal forms of A and I coincide up to the isomorphism [Gill62]. We claim the following lemma.

Lemma 3.3. Let $I = (T, X, Y, \Delta, \Lambda, t_0)$ be a completely specified initially connected deterministic FSM with q states, $q \leq |V|$. If $\Pi_I(W \cup XW) = \Pi_I(W)$ then any two states of the same class of $\Pi_I(W)$ are equivalent.

Proof. We claim that the partition $\Pi_I(W \cup XW)$ is a refinement of the partition $\Pi_I(W)$ of the state set T. Proven by contradiction, if this were not the case, there would be states that are not distinguishable by the set $(W \cup XW)$, but distinguishable by the set W. Suppose that a class of $\Pi_I(W)$ contains two distinguishable states. We choose a class C of $\Pi_I(W)$ and its states t_1 and t_2, such that the length of a sequence β which distinguishes these states is minimal. Thus $\Lambda(t_1,\beta) \neq \Lambda(t_2,\beta)$. β can not be a single input symbol, because in this case, $\Pi_I(W \cup XW) \neq \Pi_I(W)$, a contradiction. Now let the length of β be more than one, i.e. $\beta = x\alpha$. Consider the states $\Delta(t_1,x)$ and $\Delta(t_2,x)$. $\Lambda(\Delta(t_1,x),\alpha) \neq \Lambda(\Delta(t_2,x),\alpha)$. Since the sequence β has the minimal length, the states $\Delta(t_1,x)$ and $\Delta(t_2,x)$ belong to different classes C_1 and C_2 of $\Pi_I(W)$. This means there exists a sequence γ in the W set which distinguishes them. In this case, the sequence $x\gamma$ also distinguishes the states t_1 and t_2, and it belongs to the set XW. This

contradicts our assumption that $\Pi_I(W \cup XW) = \Pi_I(W)$. Therefore, any two states of the same class of $\Pi_I(W)$ are equivalent. This completes the proof of the Lemma, and thus completes the proof of Proposition 3.2.

The fault coverage of the test suite VX^1W is proven to be complete in the class $\mathfrak{I}_V$. The size of this class depends on the particular state cover V chosen for test derivation. At one extreme, a state cover V_B of the minimal form B (or equivalently, a class cover) may serve as the set V for the test suite. The corresponding test suite V_BX^1W is n-complete, if the V_B set has no less than n sequences. At the other extreme, we have the set V_BX^{m-n}, which ensures that all states are reached in any implementation with up to m reachable states. The corresponding test suite $V_BX^{m-n+1}W$ is m-complete; moreover, it is complete in the class $\mathfrak{I}_{V_BX^{m-n}} \supset \mathfrak{I}_m$. Consider an arbitrary state cover V, $V_B \subset V \subset V_BX^{m-n}$. It yields a test suite VX^1W complete in the class $\mathfrak{I}_V \supset \mathfrak{I}_n$. The relationships between the classes of implementations covered by the considered test suites are illustrated in Figure 3.

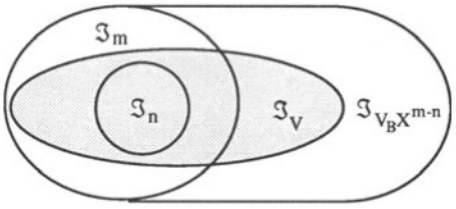

Figure 3 Classes of implementations.

The test suite VX^1W is shorter than the test suite $V_BX^{m-n+1}W$ if $V \subset V_BX^{m-n}$, however, its fault coverage is worse, as it may not cover all the FSMs in the universal set $\mathfrak{I}_m$. Uncovered machines have states which are not reachable when transfer sequences of the chosen set V are applied to their initial states. The test suite based on the selected state cover V reflects a tradeoff between the fault coverage and the size of a test suite.

At this point, different heuristics can be applied to choose a state cover for the given non-reduced machine such that a reasonable tradeoff is reached. As an example, based on the specifics of the given protocol, it might be possible to select a limited number of transfer sequences which could most probably lead to additional potentially non-equivalent states.

Here we present one particular assumption on reachability of these states, namely, **if an additional state exists in a faulty implementation, then it is reachable from the initial state with the shortest transfer sequence of the specification.** Under this assumption, the maximal number of possible states in the implementation detected by the test suite becomes a derivative of the number of the transfer sequences given in the specification. A natural intention to avoid infinite or exhaustive tests is behind this particular assumption.

This assumption suggests that one should construct a test suite based on a state cover of the given machine that contains all possible shortest transfer sequences for every state. In particular, a *canonical* state cover of A is a state cover V such that if α is a transfer sequence of the minimal length taking A from the initial state into a certain state, then $\alpha \in V$. Since the initial state is a very special state in the initialized FSMs, the set V includes only the empty transfer sequence for this state. It is clear that $|V| \geq m$, the number of states in A, and in the case that some state has several minimal transfer sequences, $|V| > m$. Every FSM possesses a unique canonical state cover. Therefore such a state cover characterizes the structure of the given non-reduced FSM. Note that substituting the original non-reduced machine by its minimal form, as

required by the existing test derivation methods, results in the loss of these characteristics. As a consequence, m-complete test suites tend to explode in size.

Example. We illustrate the idea using the FSM A (Figure 1). The canonical state cover is

$V = \{\varepsilon, b, aab, cba, c, a, cbc, aa, cb\}$.

There is only one state, namely, state 3, which has two transfer sequences of the same minimal length, aab and cba. All the remaining states have single minimal transfer sequences. The above assumption suggests that in an implementation, up to 9 distinct states can be reached with this state cover. Following the approach suggested by Proposition 2.2, we can derive a test suite

$VX^1W = \{\varepsilon, b, aab, cba, c, a, cbc, aa, cb\}\{\varepsilon, a, b, c\}\{ac\}$.

It is much shorter that the test suite V_BX^5W.

As shown in Figure 3, the test suite VX^1W does not guarantee the detection of all faults within the bound m. However, it is an n-complete test suite, so its fault detection power is no less than any other n-complete test derived from the minimal form. The domain in the class $\mathfrak{I}_m$ uncovered by the test suite based on the canonical state cover represents a class of FSMs with up to m states whose detection is not guaranteed. Intuitively, this class contains those faulty implementations of A where a certain state of A is split during the implementation process into several non-equivalent states. Faults uncovered by VX^1W seem least realistic, in the sense that implementing an FSM through state splitting can be viewed as a rather unusual way of deriving an FSM implementation.

Thus, if we are required to derive a test suite from an FSM, and extra states in its implementations are not excluded, we may now choose between the two following assumptions:

- The number of the additional states in the implementation does not exceed a certain limit. It is a typical worst-case assumption.

- If these states exist then they are reachable from the initial state with shortest sequences defined by the given machine. It is a more realistic assumption, as it relies on the structural information available in the specification.

In general, any particular bound m is not easy to justify. We can only hope that the actual number of states does not exceed m. Selecting a suitable value for the bound m without knowledge of the class of implementations under test and their interior structure is very difficult, perhaps even requiring guesswork, although any faulty machine within this limit will definitely be detected by an m-complete test suite. The penalty comes from the test explosion effect. Every extra guessed state increases the size of the test suite at least by the factor $O(|X|)$.

In the second case, tests do not suffer the explosion. The maximal number of covered distinct states naturally follows from the properties of the given specification machine with redundant states. The penalty is a possibly incomplete coverage of the universe of machines with the number of states defined by the cardinality of the canonical state cover.

The two alternative assumptions usually yield different test suites. The effectiveness of the implied test derivation strategies can be evaluated in an experimental way. In fact, we have conducted such an experiment for a simple protocol (see Section 5). Before we report it, we first consider the case of specification FSMs with unreachable redundant states (until now the specifications were assumed to be at least initially connected).

4 FSMS WITH REDUNDANT UNREACHABLE STATES

A specification machine with redundant states may not necessarily be connected. Here, we focus our attention on the problems arising from the presence of redundant unreachable states in the given specification machine.

Consider an initialized minimal FSM A with some states unreachable from the initial state. The existing test derivation methods cannot be directly applied to such a machine. To apply an existing method for test derivation, we should substitute the original specification by its connected submachine A^*. Suppose that A has m states, whereas A^* has n states, $n<m$. A number of methods can now be applied to derive a test suite under the assumption that the implementation machines have up to n states.

Consider the following example. The FSM A in Figure 4 has state 4 unreachable from the initial state 1.

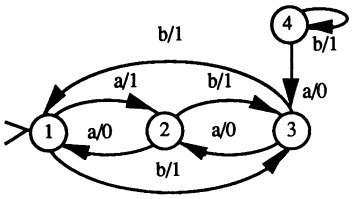

Figure 4 FSM A with an unreachable state.

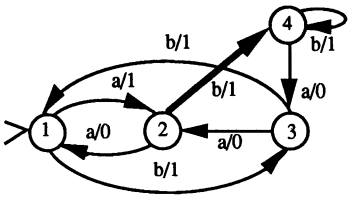

Figure 5 Faulty implementation.

If state 4 with its outgoing transitions is deleted from the original machine, then, for example, the W-method can be employed. A state cover of the connected submachine A^* is $\{\varepsilon, a, b\}$. There is a distinguishing sequence aa, that is $W = \{aa\}$. The test suite $\{\varepsilon, a, b\}\{\varepsilon, a, b\}\{aa\}$ is complete in the class of FSMs with up to three states (3-complete test suite). However, if the transition 2-b/1->3 is implemented as the transition 2-b/1->4, as shown in Figure 5, then this test suite can not reveal the fault. The only known remedy for such a situation is to derive a test suite which is complete w.r.t. four possible states:
$\{\varepsilon, a, b\}X^2\{aa\}$.

As already discussed in Section 3.2, this approach results in a sharp increase in the length of tests. At the same time, it neglects to explore the structural information provided by the original specification, as in the case of the equivalent states. In the presence of unreachable states, this information can be used to increase the distinguishing power of a W set required for state identification, thus increasing the fault coverage of a resulting test suite. An adjustment to the W-method implied by the above suggestion is presented below.

Given an FSM A, a set of input sequences is said to be a *characterization set* W_A of A if it distinguishes every reachable state of A from any other non-equivalent state including unreachable ones.

The new key feature of our notion of the W set is that unreachable states are not just ignored, as they are in the traditional notion of characterization sets. At the same time, it is a further generalization of the notion introduced in Section 3.1. The traditional techniques for constructing characterization sets are applicable to the full extent in this case, since state reachability is not required for deciding state distinguishability. Note that the diagnostic power of a W set could be increased further if we add as a requirement, the pairwise distinguishability of unreachable states. This option might be useful for deriving tests for fault localization. In most cases, however, they become more lengthy than tests for fault detection only.

In the above example, the sequence aa is not a characterization set of the FSM A (states 3 and 4 produce the same output sequence 00). The extended sequence aaa may serve as a characterization set W_A, as it yields different output reactions in the given four states.

A test suite preserves the traditional structure: VX^1W_A, where V is a state cover for all reachable states, i.e. the one of the connected submachine of A.

In our example, the test suite is

$$VX^1W_A = \{\varepsilon, a, b\}\{\varepsilon, a, b\}\{aaa\} = \{aaaaa, abaaa, baaaa, bbaaa\}.$$

The faulty implementation shown in Figure 5 fails this test. Compared to the first test suite, it has four additional test events; however, it is half the size of the test suite

$$VX^{4-3+1}W = \{\varepsilon, a, b\}X^2\{aa\}$$

produced by the original W-method.

The proposed adjustment to the W-method results in an increase in the length of tests linear with respect to m-n, the number of redundant unreachable states. The exhaustive solution offered by the original W-method is exponential with respect to m-n.

The test suites derived by the proposed technique are n-complete but usually not m-complete. They are complete in a certain class $\mathfrak{I}_W$ of FSMs. The set $\mathfrak{I}_W$ is a superset of $\mathfrak{I}_n$; machines in the set $\mathfrak{I}_W \backslash \mathfrak{I}_n$ can be characterized as follows. An FSM $I \in \mathfrak{I}_W \backslash \mathfrak{I}_n$ if it has an additional state whose reaction to the W set is different from the reaction of any reachable state of A. Note that the W set can now identify not only all of the states of the connected minimal submachine as in the original method, but also redundant states once they become reachable in a faulty implementation.

In the case where the given FSM has both reachable and unreachable redundant states, the presented technique has to be used in conjunction with the technique of Section 3.2.

5 EXPERIMENTAL RESULTS

To illustrate the proposed techniques for test derivation from specifications with redundant states, we consider the INRES protocol [Hogr92]. This simple protocol has been widely used in a number of publications, so we omit here its detailed description. The behavior of the responder part of this protocol can be specified by an EFSM given in Figure 6.

This EFSM has three control states and an internal Boolean variable v. First, we unfold it into a pure FSM. The input alphabet is: 1- CR, 2 - IDISr, 3 - ICONrsp, 4 - DT0, 5 - DT1. The output alphabet is 1 - ICONi; 2 - DR; 3 - CC; 4 - ACK0; 5 - ACK0, IDATi; 6 - ACK1; 7 - ACK1, IDATi; 8 - null. The three control states combined with the two possible values of v give six states as shown in Figure 7.

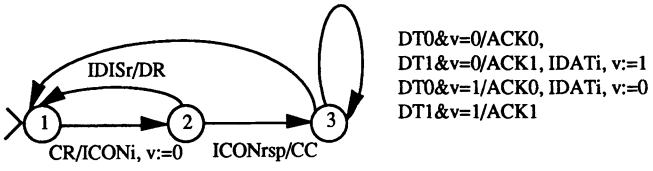

Figure 6 The EFSM of the INRES responder.

The FSM A is completely specified under the completeness assumption [PBD93], the transitions implied by this assumption (looping transitions with the output 8) are not depicted. State 21 is not reachable from the initial state. There are also two equivalent states, namely, 10 and 11. Thus FSM A has both types of redundant states. Its minimal connected form B is given in Figure 8. We use this example to compare several test suites derived by different strategies

discussed in the paper.

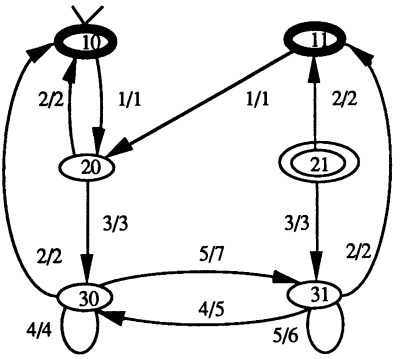

Figure 7 The FSM A unfolded from the EFSM.

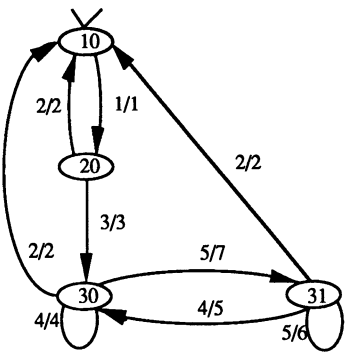

Figure 8 The minimal form B of A.

Solution 1

Assuming $m = n = 4$, i.e. that the number of states in implementations never exceeds that of the minimal form B, we apply the method from Section 3.1 to FSM A. Deriving a characterization set $W = \{3, 4\}$ based on reachable states, we find that state 11 is equivalent to 10, i.e. $\Pi(W) = \{10,11; 20; 30; 31\}$. Finally, we determine a class cover $V_C = \{\varepsilon, 1, 13, 135\}$ and obtain a test suite complete in the class $\mathfrak{I}_4$:

$TS1 = \{\varepsilon, 1, 13, 135\}X^1\{3, 4\} = \{\varepsilon, 1, 13, 135\}\{\varepsilon, 1, 2, 3, 4, 5\}\{3, 4\} =$ { 113, 114, 123, 124, 1313, 1314, 1323, 1324, 1333, 1334, 1343, 1344, 13513, 13514, 13523, 13524, 13533, 13534, 13543, 13544, 13553, 13554, 143, 144, 153, 154, 23, 24, 33, 34, 43, 44, 53, 54 }.

There are 34 test cases of total length 122.

It is easy to see that if the assignment v:=0 of the transition from the first control state of the EFSM is not implemented by mistake, then state 11 is no longer equivalent to state 10 and state 21 becomes reachable. It is intuitively clear that the above test suite hardly has the power to detect such an error.

Solution 2

Now we assume $m = 6$ and derive a test suite complete in the class $\mathfrak{I}_6$ following the same method in the Section 3.1. The class state cover and characterization set are as in Solution 1. The test suite is

$TS2 = \{\varepsilon, 1, 13, 135\}X^3\{3, 4\}$.

This test suite has 850 test cases of total length 4750.

It detects any fault within six states at a very high cost; its length is about 40 times more than that of the $TS1$.

Solution 3

Next we devise a test suite according to the techniques presented in Sections 3.2 and 4. The canonical state cover of the non-reduced FSM A is $\{\varepsilon, 1, 13, 135, 1352\}$. Note that compared

to the above solutions, there is an additional transfer sequence 1352 for a reachable redundant state 11. For state identification we use the following characterization set W = {34, 4}. Note that with this W, it now becomes possible to distinguish all reachable states from the unreachable state 21.

$TS3$ = {ε, 1, 13, 135, 1352}{ε, 1, 2, 3, 4, 5}{34, 4} = {1134, 114, 1234, 124, 13134, 1314, 13234, 1324, 13334, 1334, 13434, 1344, 135134, 13514, 1352134, 135214, 1352234, 135224, 1352334, 135234, 1352434, 135244, 1352534, 135254, 135334, 13534, 135434, 13544, 135534, 13554, 1434, 144, 1534, 154, 234, 24, 334, 34, 434, 44, 534, 54}.

Here are 42 test cases of total length 193. This test suite is only about 60% longer than the $TS1$.

Solution 4

One may view the test suites $TS1$ and $TS2$ as two extreme alternatives. We make a compromising assumption that the number of states is not more than 5 and derive a test suite complete in the class $\mathfrak{I}_5$.

$TS4$ = {ε, 1, 13, 135}X^2{3, 4}.

This test suite has 170 test cases of total length 780.

All the above obtained tests provide complete fault coverage in the class $\mathfrak{I}_4$. The test suite $TS2$ also guarantees complete fault coverage in the class $\mathfrak{I}_6$, but $TS1$, $TS3$ and $TS4$ do not. Ideally, we should compare their fault coverage in the class $\mathfrak{I}_6 \backslash \mathfrak{I}_4$. To this end, we have to enumerate all the machines of this set and compare the numbers of FSMs which fail these test suites. The total amount of $(6x8)^{(6x5)} - (4x8)^{(4x5)}$ machines should be checked against the three test suites. This task is beyond our current computing power. Instead, we have decided to perform a restricted, controlled mutation of transitions to estimate the fault coverage. To keep the number of mutants reasonable, output faults are excluded, and only certain transfer faults are simulated for the following transitions of the FSM A (Figure 7):

10-1/1->20 {21}; 20-2/2->10 {11}; 30-2/2->10 {11}; 11-1/1->20 {10, 11, 21, 30, 31}; 11-3/8->11 {20, 21, 30, 31}; 11-4/8->11 {20, 21, 30, 31}; 21-1/8->21 {10, 11, 20, 30, 31}; 21-2/2->11 {20, 21, 30, 31}; 21-3/3->31 {10, 11, 20, 21, 30}; 21-4/8->21 {10, 11, 20, 30, 31}.

This fault model describes $2^3 \times 5^3 \times 6^4$ = 1,296,000 FSMs. The idea behind the chosen mutations is to generate mainly those mutants which have more than four distinct states.

We have designed a tool which constructs a mutant machine according to the user defined fault model and checks whether or not it passes a given test suite. If the mutant passes the test suite then the tool verifies its equivalence to a specification FSM. Among 1,296,000 FSMs, there are 4368 mutants equivalent to the original FSM A (Figure 7). The fault coverage of a test suite is a percentage of non-equivalent (faulty) mutants that fail it [BPY94]. The results of the experiment are reported in Table 1. Times are measured in minutes for a UNIX machine SONY NWS 3470 (17 MIPS, 16 MB).

Note that the test suite $TS2$ was also executed against the generated mutants mainly to check the performance of the tool in the case of lengthy tests. As expected, CPU time of calculating the fault coverage is mainly determined by the total length of a test suite. On the other hand, a test suite with a low fault coverage, such as $TS1$, also consumes much time since each test case must be tried against about 50% of mutants, as opposed to the test suite $TS3$ which can recognize a faulty mutant much faster.

Table 1 Experimental results

Test suites	*# of test cases*	*Total length*	*Fault coverage (%)*	*CPU time (min)*
*TS*1	34	122	49.983277	382.67
*TS*2	850	4750	100.000000	806.27
***TS*3**	**42**	**193**	**99.945495**	**41.48**
*TS*4	170	780	99.666778	517.99

An additional experiment was conducted to estimate more precisely the fault coverage of the test suite *TS*3. The above given list of mutated transitions was extended by including the following:

11-2/8->11 {20, 21, 30, 31}; 11-5/8->11 {20, 21, 30, 31}; 21-5/8->21 {10, 11, 20, 30, 31}.

The extended fault model describes 194,400,000 FSMs, among them 26,016 machines are found to be equivalent to the specification FSM *A*. It took 2571.67 min. of CPU time to determine that *TS*3 has even a higher fault coverage of 99.997794% in this class of faults.

The results of the experiments indicate that the test suite *TS*3 for the INRES protocol derived by the proposed techniques has a high fault coverage with a reasonable length compared to other considered solutions.

6 CONCLUSION

In this paper, we have addressed the problem of test derivation from an FSM specification with redundant states which may create additional states in implementations. We have first demonstrated that the state minimization required by the existing methods for completely specified FSMs is not an obligatory step of the test derivation process. The methods can be easily extended to deal with even non-reduced machines. In other words, our first result is that the traditional assumption on the minimality is not a necessary one. Next, based on an observation that the existing approaches, which allow for the possibility of additional states in implementations, yield tests of exponential length, we have proposed two techniques used in combination. The first one is based on the idea of extending a state cover with an intention to reach additional states in implementations obtained from equivalent states reachable in the specification. The second technique suggests an extension of a characterization set with an intention to identify those additional states obtained from unreachable states. We have shown how these ideas are incorporated into the classical W-method. Our experimental results demonstrate that the proposed approach offers a reasonable compromise between the fault coverage and length of tests.

The two basic ideas of extending state covers and characterization sets can be similarly incorporated into other methods for deterministic machines, such as the UIOv-method [VCI89], the Wp-method [FBK91] and the methods based on harmonized state identifiers [Petr91], [LPB94b], which rely on the reset in the implementations. These ideas can also be used to improve the fault coverage of tests produced by a number of UIO-based methods [SiLe89], [YPB93] which yield a single test sequence. The adjustments concern covering all of the transitions (even from redundant reachable states) and extending the UIO-sequences (or any other state identifiers) in such a way that reachable states are distinguished from unreachable states.

The ideas involved in the proposed techniques seem useful not only for the deterministic case, but also for nondeterministic FSMs. Regardless of the relation used for testing, be it equivalence or reduction, the existing methods, such as the GWp-method [LBP94a], the HSI-method [LPB94b], and the SC-method [PYL93], [PYB94] can be applied only to observable and connected NFSMs. If a given machine is not observable then it is always possible to

transform it into an observable form. A classical algorithm [HoUl79] for automata determinization may well produce an observable machine with both equivalent and unreachable states. However, these states deserve a special treatment, as they are, in fact, only certain subsets of states of the original non-observable machine from which an implementation is usually derived. Another open issue of the test derivation from nondeterministic machines arises due to peculiarities of the structures of complete tests for the equivalence and the reduction relations [PYB94] used in conformance testing.

Acknowledgments
This work was partly supported by the HP-NSERC-CITI Industrial Research Chair on Communication Protocols at the Université de Montréal and the Telecommunication Advancement Foundation of Japan. The authors would like to thank S. A. Ezust for comments to a previous version of this paper.

7 REFERENCES

[BoPe94] G. v. Bochmann and A. Petrenko, "Protocol Testing: Review of Methods and Relevance for Software Testing", ISSTA'94, ACM International Symposium on Software Testing and Analysis, Seattle, U.S.A., 1994, pp. 109-124.

[BPY94] G. v. Bochmann, A. Petrenko, and M. Yao, "Fault Coverage of Tests Based on Finite State Models", the Proceedings of IFIP TC6 Seventh International Workshop on Protocol Test Systems, 1994, Japan.

[Chow78] T. S. Chow, "Test Design Modeled by Finite-State Machines", IEEE Trans., SE-4, No. 3, 1978, pp. 178-187.

[FBK91] S. Fujiwara, G. v. Bochmann, F. Khendek, M. Amalou, and A. Ghedamsi, "Test Selection Based on Finite State Models", IEEE Trans., SE-17, No. 6, 1991, pp. 591-603.

[Gill62] A. Gill, Introduction to the Theory of Finite-State Machines, NY, McGraw-Hill, 1962, 207p.

[Henn64] F. C. Hennie, "Fault Detecting Experiments for Sequential Circuits", IEEE 5th Ann. Symp. on Switching Circuits Theory and Logical Design, 1964, pp. 95-110.

[Hogr91] D. Hogrefe, "OSI Formal Specification Case Study: The Inres Protocol and Service", University of Berne, Technical Report, 1991.

[Hopc71] J. E. Hopcroft, "An n log n Algorithm for Minimizing States in a Finite Automaton", Theory of Machines and Computations, NY, Academic Press, 1971, pp. 189-196.

[HoUl79] J. E. Hopcroft, J. D. Ullman, Introduction to Automata Theory, Languages, and Computation, Addison-Wesley, 1979, 418p.

[Koha78] Z. Kohavi, Switching and Finite Automata Theory, NY, McGraw-Hill, 1978.

[LeYa94] D. Lee and M. Yannakakis, "Testing Finite-State Machines: State Identification and Verification", IEEE Trans. on Computers, Vol. 43, No. 3, 1994, pp. 306-320.

[LPB94a] G. Luo, A. Petrenko, and G. v. Bochmann, "Test Selection based on Communicating Nondeterministic Finite State Machines using a Generalized Wp-Method", IEEE Trans., Vol. SE-20, No. 2, 1994, pp. 149-162.

[LPB94b] G. Luo, A. Petrenko, and G. v. Bochmann, "Selecting Test Sequences for Partially-Specified Nondeterministic Finite State Machines", the Proceedings of IFIP TC6 Seventh International Workshop on Protocol Test Systems, 1994, Japan.

[Moor56] E. F. Moore, "Gedanken-Experiments on Sequential Machines", Automata Studies, Princeton University Press, Princeton, NJ, 1956.

[PBD93] A. Petrenko, G. v. Bochmann, and R. Dssouli, "Conformance Relations and Test Derivation", IFIP Transactions, Protocol Test Systems, VI, (the Proceedings of IFIP TC6 Fifth International Workshop on Protocol Test Systems, 1993), Ed. by O. Rafiq, 1994, North-Holland, pp. 157-178.

[Petr91] A. Petrenko, "Checking Experiments with Protocol Machines", IFIP Transactions,

Protocol Test Systems, IV (the Proceedings of IFIP TC6 Fourth International Workshop on Protocol Test Systems, 1991), Ed. by Jan Kroon, Rudolf J. Heijink and Ed Brinksma, 1992, North-Holland, pp. 83-94.

[PYB94] A. Petrenko, N. Yevtushenko, and G. v. Bochmann, "Experiments on Nondeterministic Systems for the Reduction Relation", Universite de Montreal, DIRO, Technical Report #932, 1994, 23p (submitted for publication).

[PYL93] A. Petrenko, N. Yevtushenko, A. Lebedev, and A. Das, "Nondeterministic State Machines in Protocol Conformance Testing", IFIP Transactions, Protocol Test Systems, VI, (the Proceedings of IFIP TC6 Fifth International Workshop on Protocol Test Systems, 1993), Ed. by O. Rafiq, 1994, North-Holland, pp. 363-378.

[SiLe89] D. P. Sidhu and T. K. Leung, "Formal Methods for Protocol Testing: A Detailed Study", IEEE Trans. Vol. SE-15, No. 4, 1989, pp. 413-425.

[Ural92] H. Ural, "Formal Methods for Test Sequence Generation", Computer Comm., Vol. 15, No. 5, 1992, pp. 311-325.

[Vasi73] M. P. Vasilevski, "Failure Diagnosis of Automata", Cybernetics, Plenum Publishing Corporation, NY, No. 4, 1973, pp. 653-665.

[VCI89] S. T. Vuong, W. W. L. Chan, and M. R. Ito, "The UIOv-method for Protocol Test Sequence Generation", Proceedings of IFIP TC6 Second International Workshop on Protocol Test Systems, 1989, Ed. by J. de Meer, L. Machert and W. Effelsberg, North-Holland, pp. 161-175.

[YaLe91] M. Yannakakis, D. Lee, "Testing Finite State Machines", Proceedings of the 23d Annual ACM Symposium on Theory of Computing, Louisiana, 1991, pp. 476-485.

[YPB93] M. Yao, A. Petrenko and G. v. Bochmann, "Conformance Testing of Protocol Machines without Reset", IFIP Transactions, Proceedings of the IFIP 13th Symposium on Protocol Specification, Testing and Verification, Ed. by A. Danthine, G. Leduc and P. Wolper, 1993, North-Holland, pp. 241 - 253.

8 BIOGRAPHY

Alexandre Petrenko received the Dipl. degree in electrical and computer engineering from Riga Polytechnic Institute in 1970 and the Ph.D. in computer science from the Institute of Electronics and Computer Science, Riga, USSR, in 1974. Since 1992, he has been with the Université de Montréal, Canada. His current research interests include communication software engineering, protocol engineering, conformance testing, and testability.

Teruo Higashino received the B.E., M.E., and Ph.D. degrees in Information and Computer Sciences from Osaka University, Osaka, Japan, in 1979, 1981 and 1984, respectively. Currently, he is an Associate Professor in the Department of Information and Computer Sciences at Osaka University. In 1990 and 1994, he was a Visiting Researcher of Dept. I.R.O. at the Université de Montréal, Canada. His current research interests include design and analysis of distributed systems and communication protocols.

Tadashi Kaji is a Master course student in the Department of Information and Computer Sciences at the graduate school of Osaka University. His current research interests include testing and verification of communication protocols.