Fault detection and diagnosis capabilities of test sequence selection methods based on the FSM model

T Ramalingam*, Anindya Das† and K Thulasiraman*

Different test sequence selection methods, namely, the D-method, C-method, W-method, T-method, U-method, Uv-method and the Wp-method, are reviewed and analysed for their fault detection capabilities. We show that the C-method and the Uv-method do not provide complete fault coverage. These seven methods are formally analysed for their fault diagnosis capabilities under single fault assumption. Among these methods the W-method and the Wp-method provide the best resolution in diagnosing the fault. The test sequence selection methods are then compared based on the length of the test sequences they select, and their fault detection and diagnosis capabilities.

Keywords: fault detection, test sequence selection, FSM, conformance testing

Conformance testing (CT) of a protocol is intended to ensure that a given implementation of a protocol is equivalent to the standard specification of that protocol[1,2]. The quality of interworking among heterogeneous subsystems in a distributed system can be assured through conformance testing of each subsystem. Conformance testing involves selection of a test suite from the specification, execution of the test suite on the implementation under a specific test environment, and analysis of the test results[2]. A recent trend is to describe the specification of a protocol using one of the specification languages such as Estelle[3], LOTOS[4] or SDL[5]. Such a specification has many advantages, including automatic test suite selection[6]. Both control flow and data flow aspects of a protocol have to be tested to certify an implementation. The control flow part of any protocol can be represented as a deterministic finite state machine (FSM), and it can be derived from the protocol specified in Estelle, LOTOS or SDL[7,8]. In this paper, we focus on the test suite selection methods for testing the control flow aspects of a protocol. We assume that the specification and the implementation are represented as FSM.

The OSI conformance testing methodology and framework[2] defines a test suite as a set of test cases, one for each test purpose. A test case is a set of event sequences. A verdict of *pass*, *fail* or *inconclusive* is also assigned to each sequence in the test case. A verdict for an event sequence depends on the specification of the protocol and the test purpose. Different methods have been proposed in the literature for selecting test suites from an FSM specification. Test suites selected in these methods are simply a (test) sequence of input-event and the expected output-event pairs. Also, the pass verdict is implicitly assigned to the test sequence. During testing an implementation which is viewed as a black box with some Points of Control and Observation (PCO) is stimulated with the input events from the test sequence and the output events are observed. If the output event matches with the expected output event for each input-output pair in the test sequence, then the implementation is said to pass the test sequence, otherwise the implementation is said to fail. Ural[9] has reviewed various methods proposed in the past for selecting test sequences from the FSM representation of a protocol. As it is impractical to test the implementation exhaustively, these methods select only a test sequence of finite length using some selection criteria. For example, the transition tour method[10] selects test sequence in such a way that each transition of the protocol is traversed at least once. Depending on the criteria, these methods select test sequences of varied lengths, fault detection capabilities (ability to detect the presence of a fault) and fault diagnosis capabilities (ability to localize the fault).

*Department of Electrical and Computer Engineering, Concordia University, Montreal, Canada
†DIRO, University of Montreal, Montreal, Canada

Paper received: 20 July 1993; revised paper received: 16 February 1994

In this paper, we review and analyse the existing FSM-based test sequence selection methods for their fault detection and fault diagnosis capabilities. We also summarize the results on the lengths of the test sequences they select. As noted in Ural[9], this comprehensive study will be useful in choosing suitable methods for selecting test sequences of a given protocol. This review and analysis also helps in understanding the complexities involved in developing test sequence selection methods with better fault detection and fault diagnosis capabilities. Detailed descriptions of these test sequence selection methods may be found in the literature[9 23]. Analysis of fault coverage of some of these methods may also be found elsewhere[21 23].

The paper is organized as follows. The basic definitions are given in the next section. We then review various FSM-based test sequence selection methods and analyse their fault detection and diagnosis capabilities. These methods are compared based on their fault detection and diagnosis capabilities, and the length of test sequences the methods select. Finally, we conclude with discussions and some avenues for future research. Proofs of lemmas are omitted to conserve space. However, they may be found elsewhere[24].

PRELIMINARIES

As discussed earlier, we model the control flow aspect of a protocol (henceforth referred to as 'protocol' for simplicity) specification as well as its implementation by an FSM.

An FSM M can be formally defined as a 5-tuple $M = (S, s_1, I, O, T)$, where S is the nonempty set of states of M in which s_1 is a designated state called the *initial state*. I and O are nonempty sets of possible inputs and outputs of the protocol, respectively. The transition function T is a partial function defined as $T : S \times I \rightarrow S \times O$. $T(s_i, a) = (s_j, o)$ means that the FSM M at state s_i makes a transition to state s_j when the input a is applied producing the output o. Graphically, this is also represented as $s_i - a/o \rightarrow s_j$. We call an FSM *M completely specified* if at each state s_i in M and for each input a in I, there is an outgoing transition from s_i with input a.

An FSM $M = (S, s_1, I, O, T)$ can also be represented by a directed labelled graph $G = (V, E)$, where $S = V$ and each transition $s_i - a/o \rightarrow s_j$ corresponds to an edge in E directed from s_i to s_j with label a/o. Thus an edge in E is specified by a triple $(s_i, s_j;a/o)$.

An FSM is said to have *reset capability* if for each state s_i in S there exists a transition $(s_i, s_1;r/-)$, called a *reset transition* which resets the FSM to its initial state where 'r' denotes the 'reset' command and '$-$' denotes that the FSM does not produce any output for the reset command.

An FSM can be modified into a completely specified one by using what is called a *completeness assumption*[25]. Under this assumption, for each $s_i \in S$ and for each $a \in I$, if s_i does not have an outgoing transition with input a then a self-loop transition $(s_i, s_i;a/-)$ is added to s_i.

We denote an ordered pair (a, b) of input and output by a/b. An input-output sequence is a sequence of input-output pairs. We use the operator '$\bullet$' for concatenating input-output pairs or input symbols. '@' is an operator for concatenating input-output or input sequences. These operators are omitted in certain sequences whenever there is no confusion. An input-output sequence is said to be *applicable* at a state of an FSM if the output part of the sequence is observed when the input part of the sequence is applied to the FSM at that state. Formally, a sequence $X = a_1/o_1 \bullet a_2/o_2 \bullet \cdots \bullet a_l/o_l$ is applicable at state s_i iff $\exists\, s_{i_j} \in S$, $j = 1, 2, \ldots l, l \geqslant 1$ such that $s_i - a_1/o_1 \rightarrow s_{i_1}$ and $s_{i_{j-1}} - a_j/o_j \rightarrow s_{i_j}$ for $2 \leqslant j \leqslant l$.

Henceforth, we refer to the FSM representations of the specification of a protocol and its implementation as SPEC and IUT, respectively. In this review paper, for the analysis of fault coverage and fault diagnosis capabilities, we assume that the number of states in the IUT is not greater than the number of states in the SPEC. The faulty IUTs are assumed to have only two types of faults, namely label fault and tail state fault[25]. A transition $(s_i, s_j;a/o)$ of the SPEC is said to have a *label fault* if the corresponding transition in the IUT is $(s_i, s_j;a/o')$, where $o' \neq o$. A transition $(s_i, s_j;a/o)$ of the SPEC is said to have a *tail state fault* if the corresponding transition in the IUT is $(s_i, s_p; a/o)$, where $p \neq j$. The *fault coverage* of a test sequence selection method is the percentage of faulty IUTs the method can detect from the set of all IUTs with any number of label faults and/or tail state faults. Thus, a method is said to have *complete fault coverage* if it can detect any faulty IUT.

To detect and diagnose the fault, the input part of each input-output pair of a test sequence selected is applied to the IUT one-by-one and the output is observed. If the output is different from the expected one, the testing process is stopped and the IUT is declared faulty. The output sequence obtained thus far is analysed for diagnosing the fault. A test sequence selection method has *t-fault resolution capability of level k* if for any IUT with at most t faulty transitions, a test sequence selected by the method can localize at least one faulty transition to within a set of k transitions provided the IUT is faulty. In this paper, we analyse the test sequence selection methods for their 1-fault resolution capabilities.

Let $TEST(s_i, s_j; a/o)$ denote a sequence for testing the transition $(s_i, s_j; a/o)$. It can be divided into three subsequences, as shown below:

$$TEST(s_i, s_j; a/o) = preamble@body@postamble$$

The *preamble* is a subsequence for putting the IUT in the state s_i from its current state. The subsequence for traversing and testing the transition $(s_i, s_j; a/o)$ lies in the *body*. The *postamble* is a subsequence for putting the IUT into a known state after applying the *body* subsequence.

TEST SEQUENCE SELECTION METHODS: REVIEW AND ANALYSIS

In this section we review the existing FSM-based test sequence selection methods and analyse their fault detection and fault diagnosis capabilities. In all these methods, n will denote the number of states in the SPEC.

Distinguishing sequence method

The distinguishing sequence method (in short, the *D-method*)[12, 15, 17] assumes that the SPEC is strongly connected, reduced and completely specified. The IUT is assumed to have at most n states. It is assumed that the SPEC has a special type of input sequence called a distinguishing sequence. Formally, an input sequence X_0 is said to be a *distinguishing sequence* of a SPEC if the output sequence obtained while applying X_0 at each state in the SPEC is distinct. Let $D(s_i)$ denote the path from s_i with the input sequence X_0. The D-method involves two phases: the state verification phase and the transition testing phase. Given a distinguishing sequence X_0, the state verification phase tests whether (i) the IUT has exactly n states, and (ii) X_0 is also a distinguishing sequence of the IUT. This phase is performed using the following test subsequence starting at state s_1:

$$X_0\, T(q_1, s_2)\, X_0\, T(q_2, s_3) \ldots X_0\, T(q_n, s_1)\, X_0$$

Here, q_i denotes the state the SPEC reaches after applying X_0 at $s_i, i = 1, 2, \ldots, n$. $T(s_i, s_j)$ denotes (the sequence along) a shortest path from s_i to s_j in the SPEC, where $1 \leqslant i, j \leqslant n$. If we get the expected output sequence on applying the above sequence to the IUT, then clearly, the IUT also has n distinct states. In the transition testing phase each transition $(s_i, s_j; a/o) \in E$ is tested using the sequence $T(q_p, s_{i-1})\, X_0\, T(q_{i-1}, s_i)\, a\, X_0$, where q_p is the state of the IUT before starting the testing of transition $(s_i, s_j; a/o)$. The prefix $T(q_p, s_{i-1})$ $X_0\, T(q_{i-1}, s_i)$ is to ensure that the IUT is put in state s_i before applying the input of the transition $(s_i, s_j; a/o)$. X_0 at the end of the subsequence is to confirm whether the tail state of the transition under test is in fact s_j. The main advantage of the D-method is that it ensures complete fault coverage[23]. It is also likely that the length of the test sequence will be smaller than those obtained by other test sequence selection methods having state verification phase. Kohavi and Lavallee[13] have described a method to transform FSMs which do not have distinguishing sequences into equivalent ones which have distinguishing sequences. It will be interesting to evaluate this transformational approach with respect to the overhead involved when applying it to real life protocols.

In the following, we present our results on the analysis of the 1-fault resolution capability of this method. Our first claim is that its fault diagnosis capability is very limited if the IUT fails in the state verification phase. For example, consider the SPEC and the IUT given in *Figure 1*. Clearly, $X_0 = axbxx$ is a distinguishing sequence of the SPEC. Observe that the transition $(s_1, s_2; x/0)$ of the SPEC has a tail state fault in the IUT. Consider the prefix:

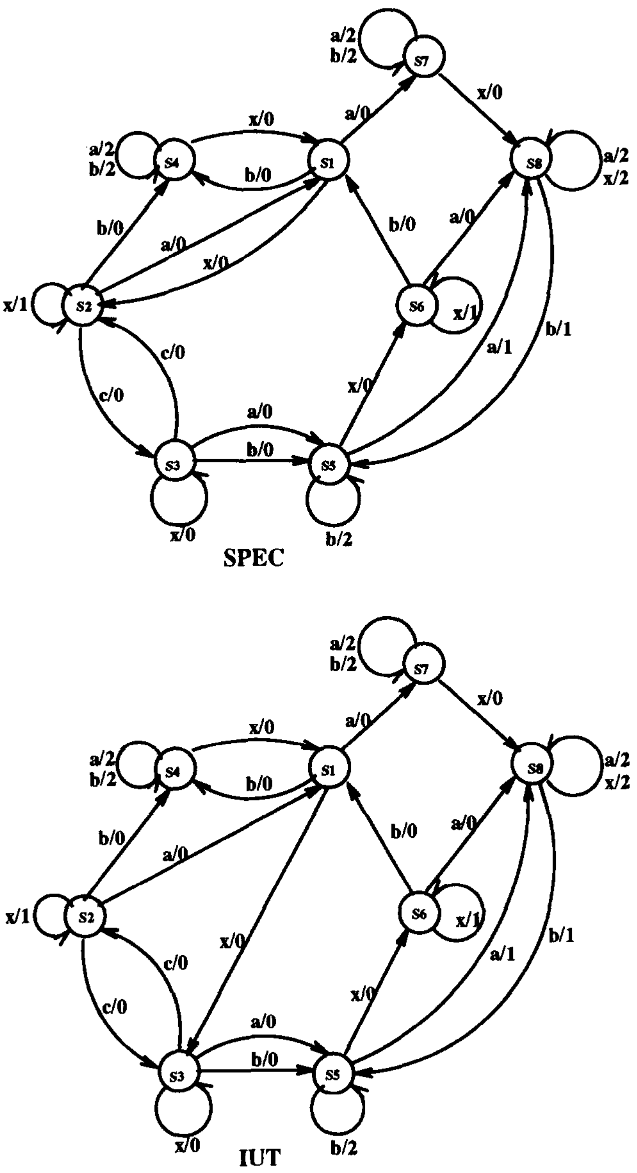


Figure 1 Example D-method

$$X_0\, T(q_1, s_2)\, X_0\, T(q_2, s_3)\, X_0\, T(q_3, s_4)\, X_0$$

of the state verification subsequence, where $q_1 = s_6$ and $q_2 = q_3 = s_2$. $T(q_1, s_2), T(q_2, s_3)$, and $T(q_3, s_4)$ are the sequences bx, c and b, respectively. The actual input sequence, expected output sequence, and the output sequence observed on applying the input sequence to the IUT are given below:

Test subsequence	=	*axbxx*	*bx*	*axbxx*	*c*	*axbxx*	*b*	*axbxx*
Expected output	=	00101	00	00000	0	00001	0	20000
Observed output	=	00101	00	00000	0	00001	0	00101

Thus the IUT fails while verifying the distinguishing sequence at state s_4. However, the fault is at the transition $(s_1, s_2; x/0)$, which is a part of $T(q_1, s_2)$, $D(s_2)$ and $D(s_3)$. In general, if the IUT fails in the state verification phase, then the fault could be in any of the transitions traversed before the detection of the fault. Thus the D-method has the 1-fault resolution capability of level only $|E|$. On the other hand, suppose that the method ensures the following property:

> If an IUT passes the test sequence selected in the state verification phase of the D-method, then for each state s_i of the SPEC, the corresponding state in the IUT responds to X_0 in the same way as s_i.

Assume that the IUT fails in the second phase while testing the transition $(s_i, s_j; a/o)$. Note that $TEST$ $(s_i, s_j; a/o) = T(q_p, s_{i-1}) X_0 T(q_{i-1}, s_i) a X_0$. If an unexpected output is observed from the IUT while applying either $T(q_p, s_{i-1})$ or the first occurrence of X_0, then a transition in $D(s_p)$ or $T(q_p, s_{i-1})$ is faulty. On the other hand, if the first unexpected output $o' \neq o$ corresponds to the input a, then the transition $(s_i, s_j; a/o)$ has a label fault. However, if the first unexpected output corresponds to the last occurrence of X_0 then we can conclude that the transition $(s_i, s_j; a/o)$ has a tail state fault. Since the length of $T(q_p, s_{i-1})$ is at most $n-1$, we conclude that the fault can be diagnosed within $l_d + n - 1$ transitions, where l_d is the length of the distinguishing sequence X_0. We summarize our results on the 1-fault resolution capability of the D-method in the following lemma:

Lemma 1 *Assume that the IUT has a most n states and at most one fault. The D-method, in general, has the (*1*)-fault resolution capability of level only* $|E|$. *However, if the successful completion of the state verification phase on an IUT also implies that the response to the distinguishing sequence,* X_0 *at each state* s_i *in the SPEC is the same as the response to* X_0 *in the corresponding state in the IUT, then the fault can be located within* $l_d + n - 1$ *transitions, where* l_d *is the length of* X_0.

A known theoretical upper bound for l_d is $(n-1)n^n$[17]. Actually, l_d would be much smaller for real life protocols. The method as such does not ensure that the state in the IUT corresponding to each state s_i responds to X_0 in the same way as s_i even if the IUT passes the state verification phase. This is due to the fact that a single transition fault can permute the distinguishing sequence's response of one state into the other[16].

Characterizing sequences method

The characterizing sequence method (henceforth referred to as the *C-method*) proposed by Kohavi *et al.*[16, 17] is a fault detection experiment for testing FSMs which may not have any distinguishing sequences. Only a brief description of the method is presented here. It is assumed that the protocol specification SPEC is strongly connected and reduced. Though the C-method assumes that the SPEC is completely specified, it can be relaxed by imposing the completeness assumption on the IUT. C-method uses a set, called a characterizing set, for identifying the states. A set $C = \{C_1, C_2, \ldots, C_s\}$ of input sequences is called a *characterizing set* of a protocol specification SPEC if no two states in SPEC have the same set of output sequences when all the sequences from C are applied to them. Formally, suppose that $O_i(C_k)$ denotes the output sequence obtained by applying C_k at the state s_i. Then C is a characterizing set if for any two distinct states s_i and s_j in S, $\{O_i(C_k) \mid C_k \in C\} \neq \{O_j(C_k) \mid C_k \in C\}$. Each sequence in C is called a *characterizing sequence*. When C is a singleton, the unique characterizing sequence becomes a distinguishing sequence. A set $V_i \subseteq C$ is called an identifying set[17] of the state s_i if V_i is a minimal subset of a characterizing set C such that $\{O_i(v) \mid v \in V_i\} \neq \{O_j(v) \mid v \in V_i\}$ for any state $s_j \neq s_i$.

We describe the method for $|C| = 2$. To identify the states an identifying sequence I_i for each state s_i is computed using its identifying set V_i. Suppose $V_i = \{C_j\}$ $(j = 1$ or $2)$ for a state s_i, where $1 \leqslant i \leqslant n$. Then $I_i = C_j$ is an identifying sequence for s_i and it is called an *identifying sequence of first order*. On the other hand, if $V_i = \{C_1, C_2\}$ then an *identifying sequence of second order* is computed as follows. Let q_i and r_i denote the states of the SPEC after C_1 and C_2, respectively, are applied at s_i. As defined earlier, $T(s_i, s_j)$ is the transfer sequence which takes the SPEC from s_i to s_j. Suppose m states respond to C_1 in the same way as s_i. Then the C-method uses the sequence $I_i = [C_1 T(q_i, s_i)]^{m+1} C_2$ for uniquely identifying s_i[16]. The reason for applying $m+1$-times the sequence $C_1\, T(q_i, s_i)$ is to ensure that the IUT is in the same state s_i before an application of C_1 and the application of C_2. Similar to the D-method, this method can also be divided into two phases: state verification and transition testing phases. Let p_i be the state the SPEC enters after applying I_i at s_i, for $i = 1, 2, \ldots, n$. The test subsequence for the state verification phase is an alternating sequence of identifying and transfer sequences as given below:

$$I_1\, T(p_1, s_2)\, I_2\, T(p_2, s_3)\, I_3 \ldots I_{n-1}\, T(p_{n-1}, s_n)\, I_n$$

In the transition testing phase each transition of the SPEC is tested. The test subsequence corresponding to a transition, say $(s_i, s_j; a/o)$ consists of (i) a sequence required to put the IUT in state s_i and confirming it, (ii) the input symbol of the transition under test, and (iii) a sequence for identifying the tail state s_j. More details on the C-method may be consulted elsewhere[16, 17].

We next analyse the fault coverage and 1-fault resolution capability of the C-method. We claim that the C-method does not have complete fault coverage. We demonstrate this using the fictitious protocol SPEC and its implementation IUT shown in *Figure 2*. The same example has been used[21] for analysing the fault coverage of the U-method (to be discussed later). Let $C_1 = aa$ and $C_2 = ba$. Clearly, $C = \{C_1, C_2\}$ is a characterizing set. Also, $I_1 = I_2 = C_1$ and $I_3 = C_2$. The test sequence for the state verification phase is *aaaaabba*. The corresponding expected output sequence is 10101111. Subsequences for testing the transitions $(s_1, s_2; a/1)$, $(s_2, s_3; b/1)$, $(s_2, s_1; a/0)$, $(s_1, s_3; b/1)$, $(s_3, s_2; a/0)$ and $(s_3, s_1; b/1)$ in that order are *aaabbaaaaa*,

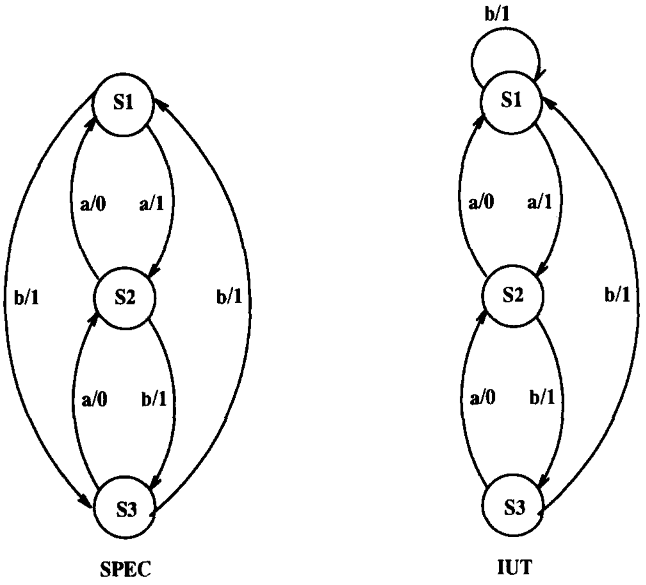


Figure 2 Protocol passed by C-method

aaaabba, *aaaaaaa*, *aaaabba*, *bbabaaa* and *bbaaabbaa*, respectively. Thus, the test sequence selected and the expected output sequence are given below:

Test sequence:
aaaaabbaaaaabbaaaaaaaaaabbaaaaaaaaaaaaabbabbabaaabbaaabbaa
Expected output:
1010111101010111010101010111101010101010101111111100111101110

The above test sequence passes the faulty IUT given in *Figure 2*. The transition $(s_1, s_3; b/1)$ has a tail state fault in the IUT. Observe that though I_3 is an identifying sequence of s_3 in the SPEC, it is not an identifying sequence for s_3 in the IUT (responses for I_3 at both s_1 and s_3 in the IUT are identical).

Let us assume that the IUTs of a SPEC are known to have at most one fault. We have shown[24] that the fact that an IUT fails while testing the transition $(s_i, s_j; a/o)$ using the test subsequence $TEST(s_i, s_j; a/o)$ does not necessarily imply that the transition in $TEST(s_i, s_j; a/o)$ is faulty. In fact the fault could be in any of the transitions traversed by the test sequence up to the point when the IUT failed. Thus we conclude that the C-method only has the 1-fault resolution capability of level $|E|$. In the following lemma we present our result on the 1-fault resolution capability of the C-method under a requirement on the state verification phase. We consider only the interesting case of all the identifying sequences being of order at most 2. Test sequences selected using the C-method for SPEC having characterizing set of cardinality more than two are in general very long and Kohavi suggests to use some alternative methods[13, 16]. Proof of the lemma is omitted to conserve space:

Lemma 2 *The C-method, in general, has the* 1*-fault resolution capability of level only* $|E|$. *However, if the success of the state verification phase ensures that the identifying sequence,* I_i *of each state in the SPEC is also an identifying sequence of the corresponding state in the IUT and if the fault is detected in the transition testing phase. Then the C-method diagnoses the fault to within* $(m+3)(l_c+n-1)+1$ *transitions. Here,* $C = \{C_1, C_2\}$ *is the characterizing set of the SPEC,* m *is the maximum number of states exhibiting the same response while applying* C_1 *at those states and* l_c *is the maximum length of a characterizing sequence in* C.

W-method

Chow[18] proposes a method for testing the control structure of software designs modelled by FSMs. In recent years, this method has been widely applied for selecting test sequences for protocol testing[22, 23]. Henceforth we refer to this method as the *W-method.* It is assumed that the protocol specification (SPEC) and its implementation (IUT) are strongly connected, reduced, completely specified and they accept the same input set, say I. The method also assumes that the SPEC has reset capability which is correctly implemented in the IUT. The IUT is allowed to have any number of states bounded by a finite estimate.

For state verification purposes this method uses a characterizing set of the IUT. It provides a scheme for obtaining a characterizing set of the IUT from a characterizing set of the SPEC. Given two sets of input sequences A and B, its concatenation AB is defined as $AB = \{a \text{ @ } b \mid a \in A \wedge b \in B\}$. By A^i we denote the concatenation of A i times for any non-negative integer i. A^0 is the empty sequence ε. Let $I[j]$ denote the set of all possible input sequences of length at most j, for $j \geqslant 0$. Clearly, $I[j] = \{\varepsilon\} \cup I \cup I^2 \cup \ldots \cup I^j$. Let $C = \{C_1, C_2, \ldots, C_s\}$ be a characterizing set of the SPEC. Let n and m denote the number of states in the SPEC and an upper bound on the number of states on the IUT, respectively. It has been proved[18] that $W = I[m-n]C$ is a characterizing set for any fault free IUT. To confirm a state in the IUT, this method applies all the sequences from W at that state.

A directed spanning tree rooted at the initial state is used for reaching any state from the initial state. We refer to this tree as a *state cover tree*. Let P_i denote the unique path in a state cover tree T from the initial state to a given state s_i, for $i = 1, 2, \ldots, n$. Reset transitions are traversed for putting the SPEC or the IUT into the initial state from any given state. Thus, P_i, for $i = 1, 2, \ldots, n$ and r are the only transfer sequences considered in this method. An input sequence $TEST(s_i, s_j; a/o)$ for testing a given transition $(s_i, s_j; a/o)$ is obtained using reset transitions, P_i, $(s_i, s_j; a/o)$ and W. That is, $TEST(s_i, s_j; a/o)$ is a concatenation of rP_iaZ for each $Z \in W$. The required test sequence is an arbitrary concatenation of the test subsequences for all the transitions in the SPEC.

The W-method assures complete fault coverage[18]. Let us compare this method with the C-method. When $m \leqslant n$, W is nothing but C. In this case the W-method applies every sequence in C at a given state s_i for identifying it in the IUT. However, the C-method only applies an identifying set V_i – a subset of C – for identifying s_i in the IUT. This is the key difference between the C-method and the W-method. This difference and the reliable reset capability together make the W-method detect any fault in the IUT, while the C-method does not have such a capability. Lemma 3 summarizes our result on the 1-fault resolution capability of the W-method on IUTs having at most n states:

Lemma 3 *If an IUT has at most n states and at most one fault, then the W-method always locates the fault to within $n + l_c$ transitions, where l_c is the length of a longest characterizing sequence selected in the W-method.*

The 1-fault resolution capability level can be rewritten as $1 + d + l_c$, where d is the depth of the state cover tree. Therefore, by minimizing the height of the tree one can improve the 1-fault resolution capability. It is worth mentioning that the reliable reset capability assumption on the SPEC and the IUT has a significant role in diagnosing the fault to within $n + l_c$ transitions. It is known that every reduced machine with n states has a characterizing set C of cardinality at most $n - 1$, where the maximal length of a characterizing sequence in C is $n - 1$[17]. So in such a case the level of 1-fault resolution capability reduces to $2n - 1$. Further localization of the fault could be achieved by applying additional test sequences to the IUT. Analysing the fault diagnosis capability of the W-method on IUTs having more states than the SPEC is an interesting problem for further study.

Transition tour method

The transition tour method (*T-method* for short)[10] assumes that the SPEC is completely specified. In this and the other two methods (U-method and Uv-method) that follow, we specify a test sequence as an input-output sequence. The test sequence is selected based on a minimal transition tour which traverses each transition in the SPEC at least once. Here the test subsequence corresponding to a transition is simply its label. This method selects a test sequence of the shortest length among all the methods discussed in this paper. The T-method neither has a state verification phase, nor does it verify the intermediate states in the IUT as it traverses the transitions. Hence, the method does not have the capability of detecting tail state faults[22,26]. For the same reason, this method cannot diagnose faults in the IUT with tail state faults, even if it certifies the IUT as faulty. In the worst case, the T-method only has the 1-fault resolution capability of level $|E|$, where $|E|$ is the number of transitions in the SPEC.

Unique input output sequence method

The unique input output (UIO) sequence method (*U-method*)[19,20,27] requires that the SPEC be strongly connected. This method also assumes for each state the existence of an input-output sequence which uniquely identifies that state. Such sequences are called *UIO-sequences*. Formally, a UIO-sequence for state s_i of an FSM M denoted by UIO_i is an input-output sequence of minimum length such that UIO_i is applicable at state s_i and it is not applicable at any other state in M. The U-method tests the transitions as follows:

> To test a transition, say $(s_i, s_j; a/o)$, the IUT is first put in state s_i. Then the input a is applied and the output is checked to verify that it is o as expected. Finally, the input part of UIO_j is applied to the current state of the IUT, and the resulting output sequence is examined to check whether the current state of the IUT is in fact s_j as expected.

Aho *et al.*[19] presented an efficient algorithm for minimizing the length of the test sequence. The algorithm gives an optimum solution to a subclass of rural postperson problem (RPP)[28,29]. The method requires that the SPEC either has the reset capability or a self loop at each state.

In general, each state of the SPEC has more than one UIO-sequence (*multiple UIO-sequences*). In the MU-method[30] further optimization of the test sequence is achieved by using multiple UIO-sequences at each state. Chen *et al.*[31] have given an efficient technique for additional reduction in the length of the test sequence by overlapping the test subsequences of successive transitions tested. A generalization of the U-method has been presented[32] that assumes the existence of multiple UIO-sequences at each state. While the U-method is applicable only for restricted classes of protocols, this generalized method can be applied on any protocol specified as an FSM. We observe that the fault coverages provided by all these methods are very close to the one assured by the U-method.

The U-method is well known for its minimal length test sequence and practical application[33]. Although the fault coverage of this method is better than the T-method, it does not provide complete fault coverage[21]. We observe that the fault resolution capability of this method is affected due to the following reasons: (i) while applying the UIO-sequences, or transferring the IUT from one state to another state using transfer sequences, it might traverse transitions which have not yet been tested; and (ii) as the method does not have a state verification phase, a UIO-sequence of a state in the SPEC may not be a UIO-sequence of the corresponding state in the IUT. We conclude that the U-method has 1-fault resolution capability of level only $|E|$.

Improved UIO sequence method

As its name suggests, the improved UIO-sequence method (*Uv-method*) is an improvement over the U-method. The improvement is suggested by Chan *et al.*[21]. In addition to the assumption made in the U-method, this method assumes that both the SPEC and the IUT are completely specified. It also assumes that the IUT is strongly connected. The method consists of two phases: UIO-verification phase and transition testing phase. In the UIO verification phase, the method checks whether the selected UIO-sequences of the states of the SPEC are also UIO-sequences of the corresponding states in the IUT. This could be done by verifying at each state s_i of the IUT that UIO_i is applicable and all $UIO_j, j \neq i$ is not applicable at that state. Though Chan *et al.* have suggested the need for verifying the UIO-sequences in the IUT, they do not provide a method for achieving this requirement. In general, finding a method to meet this requirement seems to be difficult. In the transition testing phase, all transitions are tested as in the U-method.

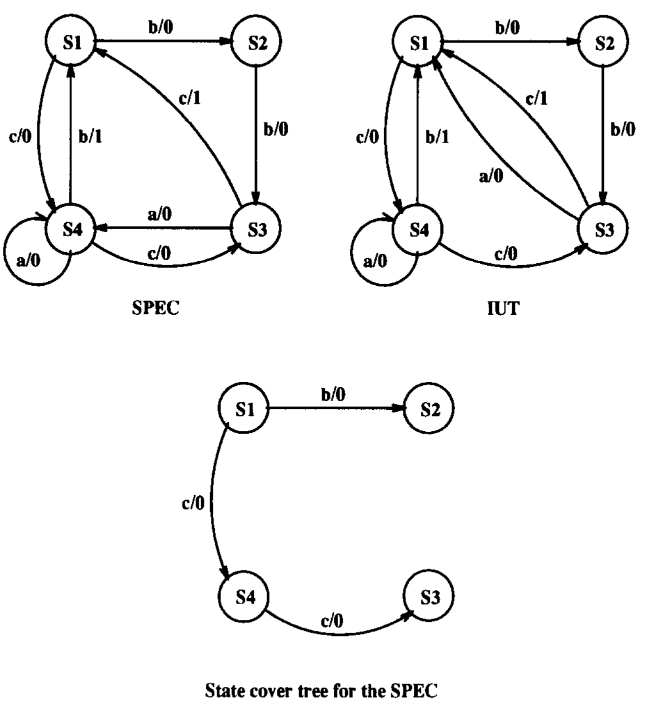
It is known that the fault coverage of the Uv-method is better than the U-method[21]. We claim that the method, however, does not guarantee complete fault coverage. Consider the SPEC and an IUT of an abstract protocol shown in *Figure 3*. The reset transitions are not shown explicitly in the figure. The transition $(s_3, s_4; a/o)$ has a tail state fault in the IUT. Observe that for both the SPEC and its IUT, $UIO_1 = c/0 \bullet b/1$, $UIO_2 = b/0 \bullet a/0$, $UIO_3 = c/1$ and $UIO_4 = b/1$ are the UIO-sequences of the states s_1, s_2, s_3 and s_4, respectively. If the paths along the state cover tree T shown in *Figure 3* and the reset transitions only are used as transfer sequences in the UIO verification phase, then the IUT passes this phase. The test sequence selected in the transition testing phase is the concatenation of the test subsequences for the transitions and the transfer sequences, as given below:

$$TEST(s_1,s_2;b/0)@T(s_4,s_3)@TEST(s_3,s_4;a/0)@ T(s_1,s_3)@TEST(s_3,s_1;c/1)@T(s_1,s_4)@TEST (s_4,s_1;b/1)@T(s_1,s_4)@TEST(s_4,s_4;a/0)@T(s_1,s_4) @TEST(s_4,s_3;c/0)@TEST (s_1,s_4;c/0)@T(s_1,s_2)@TEST(s_2,s_3;b/0)$$

Here, $T(s_1,s_4) = c/0$, $T(s_4,s_3) = c/0$, $T(s_1,s_3) = c/0 \bullet c/0$ and $T(s_1,s_2) = b/0$. The resulting test sequence which passes the fault IUT is given below:

$$b/0 \bullet b/0 \bullet a/0 \bullet c/0 \bullet a/0 \bullet b/1 \bullet c/0 \bullet c/0 \bullet c/1 \bullet c/0 \bullet b/1 \bullet c/0 \bullet b/1 \bullet c/0 \bullet b/1 \bullet c/0 \bullet a/0 \bullet b/1 \bullet c/0 \bullet c/0 \bullet c/1 \bullet c/0 \bullet b/1 \bullet b/0 \bullet b/0 \bullet c/1$$

This method is similar to the C-method. However, while the C-method only checks the applicability of the identifying set of each state at that state, this method checks for each state s_i the applicability of UIO_i and the non-applicability of all UIO_j, $j \neq i$ at that state. Still, the Uv-method does not provide complete fault coverage. As in the U-method, the Uv-method may use some transitions for putting the IUT in the start state of the transition under test. Such transitions constitute a preamble for the transition under test. This preamble may contain faulty transitions which are yet to be tested. Also, some of the transitions which constitute the UIO-sequence of the tail state of the transition under test may be faulty. In other words, even if the IUT passes the first phase, the state of the SPEC after applying a UIO-sequence, say, UIO_i at state s_i need not be equivalent to the state of the IUT after applying UIO_i at the state of the IUT corresponding to s_i. Thus even when the method detects a faulty IUT, it may not be able to identify the faulty transition. In other words, this method, in general, has the 1-fault resolution capability of level only $|E|$. The following lemma summarizes our result on the 1-fault resolution capability of the Uv-method under certain conditions. The proof of the lemma may be found elsewhere[24]:



State cover tree for the SPEC

Figure 3 Protocol passed by Uv-method

Lemma 4 *Suppose that the IUTs have at most one fault and the successful completion of the first phase assures that the UIO-sequence of each state in the SPEC is also a UIO-sequence of the corresponding state in the IUT, then the Uv-method locates the fault to within $n + 2l_u$ transitions, where n is the number of states in the SPEC and l_u is the length of a longest UIO-sequence considered, provided that fault is detected.*

A theoretical upper bound for l_u is $(n - 1)n^n$, which is the same as the upper bound known for a distinguishing sequence[17]. However, l_u is at most 5 for most of the known real life protocols[20].

Wp-method

The Wp-method introduced by Fujiwara *et al.*[23] is based on the W-method. The Wp-method assumes that the SPEC is strongly connected, reduced and completely specified. The SPEC and the IUT have reset capability and the same input set. It further assumes a finite upper bound, say, m on the number of states of the IUT. Let C be a characterizing set of the SPEC. Let $V_i \subseteq C$ be an identifying set of the state s_i in the

SPEC for each state s_i. As discussed in the W-method, $W = I[m-n]\,C$ is a characterizing set of its correct implementations which have at most m states. Clearly, in such correct implementations, $W_i = I[m-n]\ V_i$ is the identifying set of the states corresponding to each state s_i of the SPEC.

The Wp-method consists of two phases: state verification phase and transition testing phase. The first phase is mainly to verify whether W is a characterizing set of the IUT. This is done by applying every sequence in W at each state in the IUT. Paths in a state cover tree, say T, and the reset transitions are used for putting the IUT in each state for applying sequences from W. As a result, all the transitions in the state cover tree T are also tested by the end of this phase.

It is claimed[23] that if an IUT passes the state verification phase, then

Claim 1: all transitions in T are implemented correctly in the IUT.

Claim 2: W_i is an identifying set for the states in the IUT corresponding to the state s_i of the SPEC.

In the transition testing phase, all the transitions of the SPEC which are not in the state cover tree T are tested. Transitions in T are not tested here, as it is done in the first phase itself. Reset transitions and Paths in a state cover tree, say T, are used for putting the IUT in the start state of the transition under test. To confirm the tail state of a given transition, say, $(s_i, s_j; a/o) \in E - T$ in the IUT, each sequence from the identifying set W_j is applied at the tail state. Here, E denotes the set of all transitions in the SPEC.

Clearly, this method is an improvement over the W-method. Note that the aim of the first phase of the Wp-method is to assure that W_j is an identifying set of the state in the IUT corresponding to s_j. As a result, in the second phase the tail state of the transition $(s_i, s_j; a/o)$ in the IUT is confirmed by applying the sequences from W_j in the current state, instead of applying the whole characterizing set W. This is the main difference between the W-method and the Wp-method. Thus the length of the test sequence selected by the Wp-method is always less than or equal to the one selected by the W-method. The Wp-method also assures complete fault coverage[23].

We now analyse the 1-fault resolution capability of this method. We observe that Claims 1 and 2 need not always be true for IUTs passing the state verification phase successfully. We demonstrate this on an abstract protocol whose SPEC and IUT are given in *Figure 1*. Reset transitions are not shown explicitly in this figure. Here, the SPEC and the IUT have the same number of states, i.e. $m = n$. Clearly, $C = \{axbxx\}$ is a characterizing set of the SPEC. Since $m = n$, we have $W = C$ and $W_i = W$, for $1 \leqslant i \leqslant 8$. The state cover tree T used in testing is shown in *Figure 4*. With state cover tree T and the characterizing set W the IUT passes the state verification phase. In the transition testing phase, transition $(s_3, s_5; b/0)$ is tested using the test subsequence $TEST(s_3, s_5; b/0)$. This test subsequence, its

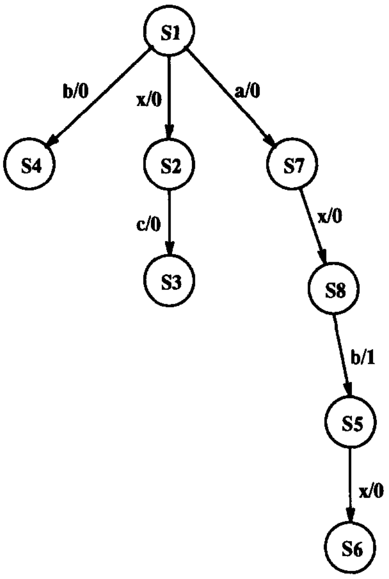


Figure 4 State cover tree of the SPEC given in *Figure 1*

expected output and the output given by the IUT are given below:

$$\begin{aligned} TEST(s_3, s_5; b/0) &= r\ P_3\ b\ W \\ &= r\ x\ c\ b\ a\ x\ b\ x\ x \\ \text{Expected output} &= -\ 0\ 0\ 0\ 1\ 2\ 1\ 0\ 1 \\ \text{Observed output} &= -\ 0\ 0\ 0\ 2\ 0\ 0\ 0\ 0 \end{aligned}$$

Using Claim 1, one could conclude that the transition $(s_3, s_5; b/0)$ has a tail state fault, and its faulty tail state in the IUT is s_4. However, this transition is fault free in the IUT. The actual fault is at the transition $(s_1, s_2; x/0)$. Though the IUT passes the first phase, the state cover tree transition $(s_1, s_2; x/0)$ has a tail state fault in the IUT. Also, though $W_2 = W_3 = W$ is an identifying set for states s_2 and s_3 in the IUT, the output sequences obtained while applying W on these states are, respectively, different from the output sequences obtained while applying W at s_2 and s_3 of the SPEC. That is, Claim 2 is not valid for this IUT. In fact the responses for W at s_2 and s_3 of the IUT are the permutation of the respective responses in the SPEC. Our results on the level of 1-fault resolution capability of the Wp-method are presented in the following lemma:

Lemma 5 *Suppose that the IUTs have at most n states and at most one fault. Then the Wp-method has 1-fault resolution capability of level* $n + l_c$, *where n is the number of states in the SPEC and* l_c *is the length of a longest sequence in* C $(= W)$. *However, if the successful completion of the first phase ensures that the identifying set* V_i $(= W_i)$ *is also an identifying set of the state in the IUT corresponding to* s_i *for each state* s_i *in the SPEC, then the wp-method localizes the fault to within* $1 + l_c$ *transitions.*

Although this method, in general, provides a shorter test sequence than the W-method, its fault resolution capability is the same as the W-method. The 1-fault resolution capability of the Wp-method can be improved by minimizing the depth of the state cover tree. This is also the case with the W-method. As noted in the W-method, if one considers a characterizing set C of cardinality $n - 1$ in which each characterizing sequence is of length at most $n - 1$, it can be deduced that the Wp-method has 1-fault resolution capability of level $2n - 1$. Moreover, the fault can be localized to within n transitions if the success of the first phase implies that the identifying set of a state in the SPEC is also an identifying set of the corresponding state of the IUT. As this is the best fault resolution possible among the existing methods, the Wp-method is improved[34] so that the two properties (Claim 1 and Claim 2) claimed by Fujiwara *et al.*[23] for this method hold on the successful completion of the first phase.

COMPARISON OF TEST SEQUENCE SELECTION METHODS

In this section we compare the test sequence selection methods based on their fault detection and diagnosis capabilities, and the length of the test sequences they select.

Fault coverage and diagnosis

Among all the methods discussed in this paper, only D-, W- and Wp-methods provide complete fault coverage. Since the T-method does not verify the intermediate states as it traverses the transitions, the fault coverage of this method is less than that of all state identification based methods. Among the state identification based methods, the fault coverages of C- and U-methods are less than that of the state verification-based methods: D, W and Wp. Comparing the C- and U-methods, the C-method has better fault coverage than the U-method, since the C-method checks the tail states of transfer sequences while the U-method does not. It is also known that the Uv-method has better fault coverage than the U-method[21]. A partial ordering among the fault coverages of the test sequence selection methods is summarized below. Here, $F(X)$ denotes the fault coverage of the method X:

$$F(T) < F(U) < F(C) < F(D) = F(W) = F(Wp)$$

$$F(U) < F(Uv) < F(W)$$

In general, the W- and Wp-methods have the best 1-fault resolution capabilities among the known methods. Let $FRL(X)$ denote the level of 1-fault resolution capability of the method X, where X is one of D-, C-, Uv-, W- and Wp-methods. Suppose that these methods somehow guarantee that the sequences used for identifying the states are also the identifying sequences of the corresponding states in the IUT on successful completion of the state verification phase. Then it follows from Lemmas 1–5 and from the known upper bounds for l_d, l_c and l_u that $FRL(D) \geqslant FRL(C) \geqslant FRL(Uv) \geqslant FRL(W) \geqslant FRL(Wp)$.

Length of test sequences selected

In general, the lengths of test sequences selected using the T-method are always less than those of the sequences selected using state identification-based methods. The test sequences are even longer if the sequences used for identifying the states are also verified in the IUT. Suppose $L(X)$ is the length of a test sequence selected in method X. The order among the lengths of test sequences selected using the various methods is given below. To compare the lengths, we assume that the same set of UIO-sequences is used in the U- and Uv-methods. The characterizing set for each of the C-, W- and Wp-methods is the union of the UIO-sequences selected for the U-method. It is assumed that the number of states of the IUT is no more than the number of states of the SPEC:

$$L(T) < L(U) \leqslant L(Uv) \quad L(U) \leqslant L(Wp) \leqslant L(W),$$
$$\text{and} \quad L(U) \leqslant L(C)$$

As noted by Ural[9], the upper bound on the length of the test sequences selected in the D-method is the longest among all the methods discussed in this paper. However, in practice it is the least among the sequences selected using the W-, Wp-, Uv- and C-methods.

CONCLUSION

We have surveyed different methods for selecting test sequences for testing communication protocols based on the FSM model. Fault coverage of each method, if available in the literature, is also reviewed. Unlike the earlier claim, we show that the Uv-method does not have complete fault coverage. We have also shown that the C-method does not have complete fault coverage. With a single fault assumption, all the FSM-based methods are formally analysed to see if they can be used for diagnosing the fault in an implementation. The level of 1-fault resolution capabilities derived for these methods demonstrates that the W- and Wp-methods are the best for diagnosing the fault to within a few transitions. A comparison of the test sequence selection methods is made based on the fault coverage, 1-fault diagnosis capability and the length of test sequences they select.

In all the test sequence selection methods discussed in this paper, the test subsequence for testing a transition is, in general, the concatenation of a preamble, label of the transition under test, state identification sequence of the tail state of the transition under test, and a postamble. For the purpose of optimization, in some of the methods certain subsequences may be overlapped or omitted. Thus a test sequence selection method will have better fault resolution capability if it confirms the correctness of all transitions in the preamble, in the sequence for identifying the tail state and the postamble of any transition, before testing this transition itself. As we have pointed out, every test sequence selection method considered in this paper meets this requirement only partially. Using this approach, it is possible to improve the existing methods for better 1-fault resolution capabilities, as we have shown in companion papers[34, 35].

In this paper, we have studied if a given test sequence selection method has complete fault coverage or not. A more general open problem is to find the exact fault coverage of a given method. One way of estimating the fault coverage of a method is to use suitable test coverage metrics[36, 37].

REFERENCES

1 Stallings, W *Networking Standards: a guide to OSI, ISDN, LAN, and MAN,* Addison-Wesley, New York (1993)

2 ISO/IEC 9646 *Information Technology – Open Systems Interconnection – Conformance Testing Methodology and Framework* (1991)

3 Budkowski, S and Dembinski, P 'An introduction to Estelle: A specification language for distributed systems', *Comput. Networks & ISDN Syst.,* Vol 14 (1987) pp 3–23

4 Bolognesi, T and Brinksma, E 'Introduction to the ISO specification language LOTOS', *Comput. Networks & ISDN Syst.,* Vol 14 (1987) pp 25–59

5 CCITT/SGx/WP3-1 *SDL, specification and description language,* CCITT Recommendations Z.100-Z.104 (1988)

6 ISO SC21 WG1 P54 *Information Technology – Open Systems Interconnection – Formal Methods in Conformance Testing,* Working Document (June 1993)

7 Von Bochmann, G and Sunshine, C A 'A survey of formal methods', in P E Green (ed.), *Computer Networks and Protocols,* Plenum Press, New York (1983) pp 561–578

8 Lee, D Y and Lee, J Y 'A well-defined Estelle specification for the automatic test generation', *IEEE Trans. Comput.,* Vol 40 No 4 (April 1991) pp 526–542

9 Ural, H 'Formal methods for test sequence generation', *Comput. Commun.,* Vol 15 No 5 (June 1992) pp 311–325

10 Naito, S and Tsunoyama, M 'Fault detection for sequential machines by transition tours', *Proc. 11th IEEE Fault Tolerent Comput. Conf.,* (1981) 238–243

11 Gill, A 'State identification experiments in finite auotomata', *Infor. & Control,* Vol 4 (1961) pp 132–154

12 Hennie, F C 'Fault detection experiments for sequential circuits', *Proc. 5th Ann. Symp. Switching Circuit Theory and Logic Design,* Princeton, NJ (November 1964) pp 95–110

13 Kohavi, Z and Lavallee, P 'Design of sequential machines with fault detection capabilities', *IEEE Trans. Electr. Comput.,* Vol 16 No 4 (August 1967) pp 473–484

14 Kohavi, I and Kohavi, Z 'Variable length distinguishing sequences and their application to the design of fault detection experiments', *IEEE Trans. Comput.,* Vol 17 (August 1968) pp 792–795

15 Gonec, G 'A method for the design of fault-detection experiment', *IEEE Trans. Comput.,* Vol 19 (June 1970) pp 551–558

16 Kohavi, Z, Rivierre, J A and Kohavi, I 'Checking experiments for sequential machines', *Infor. Sci.,* Vol 7 (1974) pp 11–28

17 Kohavi, Z *Switching and Finite Automata Theory,* McGraw-Hill, New York (1978)

18 Chow, T 'Testing software design modeled by finite state machine', *IEEE Trans. Softw. Eng.,* Vol 4 (March 1978) pp 178–187

19 Aho, A V, Dahbura, A T, Lee, D and Uyar, M U, 'An optimization technique for protocol conformance test generation based on UIO sequences and rural chinese postman tours', *Symposium of Protocol Specification, Testing and Verification* (1988) pp 75–86

20 Sabnani, K and Dahbura, A 'A protocol test generation procedure', *Comput. Networks & ISDN Syst.,* Vol 15 (1988) pp 285–297

21 Chan, W Y L, Vuong, S T and Ito, M R 'An improved protocol test generation procedure based on UIOs', *ACM SIGCOMM* (1989) pp 283–293

22 Sidhu, D P and Leung, T K 'Formal methods for protocol testing: A detailed study', *IEEE Trans. Softw. Eng.,* Vol 15 No 4 (1989) pp 413–426

23 Fujiwara, S, von Bochmann, G, Khendek, F, Amalou, M A and Ghedamsi, A 'Test selection based on finite state model', *IEEE Trans. Softw. Eng.,* Vol 17 (June 1991) pp 591–603

24 Ramalingam, T, Das, A and Thulasiraman, K *Analysis of fault detection and diagnosis capabilities of conformance testing methods for the FSM-based protocols,* Technical report, Department of Electrical Engineering, Concordia University, Montreal, Canada (1993)

25 Dahbura, A T and Sabnani, K K 'An experience in estimating fault coverage of a protocol test', *IEEE INFOCOM,* (March 1988) pp 71–79

26 Dahbura, A T, Sabnani, K K and Uyar, M U 'Formal methods for generating protocol conformance test sequences', *Proc. IEEE,* Vol 78 No 8 (1990) pp 1317–1326

27 Sabnani, K K and Dahbura, A T 'A new technique for generating protocol tests', *Proc. 9th Data Commun. Symp.,* IEEE Press (September 1985) pp 36–43

28 Kuan, M K 'Graphic programming using odd or even points', *Chinese Math,* Vol 1 (1962) pp 273–277

29 Edmonds, J and Johnson, E L 'Matching, Euler tours and the Chinese postman', *Math. Program.,* Vol 5 (1973) pp 88–124

30 Shen, Y N, Lombardi, F and Dahbura, A T 'Protocol conformance testing using multiple UIO sequences', *IEEE Trans. Commun.,* Vol 40 (August 1992) pp 1282–1287

31 Chen, M S, Choi, Y and Kershanbaum, A 'Approaches utilizing segment overlap to minimize test sequences', *Symp. Protocol Specification Testing and Verification,* Ottawa, Canada (June 1990)

32 Ramalingam, T, Thulasiraman, K and Das, A *Conformance testing of communication protocols: A general method based on matroid and graph theoretic approaches,* Technical report, Electrical Engineering, Concordia University (1993)

33 Sherif, M H and Uyar, M U 'Protocol modeling for conformance testing: Case study for the ISDN LAPD protocol', *AT&T Tech. J.* (January/February 1990) pp 60–83

34 Ramalingam, T, Das, A and Thulasiraman, K 'On conformance test and fault resolution of protocols based on FSM model', *Proc. Int. Conf. Comput. Networks, Architecture and Applic.,* Trivandrum, India (October 1992)

35 Ramalingam, T, Das, A and Thulasiraman, K *On testing and diagnosis of communication protocols based on the FSM model,* Technical report (1994)

36 Voung, S T and Alilovic-Curgus, J 'On test coverage metrics for communication protocols' *4th Int. Workshop on Protocol Test Systems,* Leischendam, The Netherlands (October 1991)

37 McAllister, M, Voung, S T and Alilovic-Curgus, J 'Automatic test case selection based on test coverage metrics', *5th Int. Workshop on Protocol Test Syst.,* Montreal, Canada (October 1992)