An Optimized Test Sequence Satisfying the Completeness Criteria

HongSe Son*, DaeHun Nyang†, SangYong Lim†, JinHo Park†, Young-Han Choe*
ByungMun Chin*, and JooSeok Song†

*ETRI(Electronics and Telecommunications Research Institute),
161 Kajong-Dong, Yusong-Gu, Taejon, 305-350, Korea
E-mail:{hson, bmchin, yhchoe}@pec.etri.re.kr
†Computer Science Dept of Yonsei University,
134 Shinchon-Dong, Seodaemun-Gu, Seoul, Korea
E-mail: nyang@emerald.yonsei.ac.kr

Abstract

We present a problem of commonly used characterization sequences and propose a new test sequence to resolve the problem. The proposed test sequence could decide whether the output fault arises in the edge being tested or one of edges in the UIO sequence. Additionally, the fault coverage is much wider than other test sequence generation methods. To achieve the goal, we introduce k-strong FSM, and show that it can be constructed from a characterization sequence. Also, we show that the length of the test sequence satisfying the completeness can be reduced. Finally, we illustrate our technique on a specific example.

1. Introduction

Conformance testing of communication protocol implementations is an area of considerable research since testing can help guarantee correct operation. In that literature, an implementation is regarded as a black box, and its conformity is verified by observing output behaviors for input sequences.

Various automatic test sequence generation methods have been proposed such as UIO-, W-, DS-methods [3]. With the methods, one transition could be tested, so a technique to constitute an overall test sequence for a specification FSM is required. One is to combine the characterization sequence using reset input and the shortest path to the state to be tested [2], [3]. Another is a transition tour, which is regarded to have worse fault capability [8].

As explained, studies in the test sequence generation method have been concentrated on the fault coverage and the length of a generated test sequence [1], [2], [4], [5], [7], [9]. For the wide fault coverage, fault tolerant UIO methods and pair wise distinguishing sequence have been proposed [4], [6], [7], [9]. However, test sequences with the wide fault coverage still cannot guarantee the correctness of its decision. In this paper, we show that commonly used test sequences such as UIO and DS fail to test a correct state transition, and focus on the scheme to guarantee the completeness of the verdict of a test sequence.

Concerning the incorrect behavior of test sequences, we define a new problem instance, and propose the input test sequence generating scheme to resolve the problem. To achieve this goal, *k-strong* FSM is defined for a certain characterization sequence. Additionally, a characterization sequence which makes an FSM to be *k-strong* is proposed using extended UIO, and its fault detection capability is examined. For the construction of an extended UIO, multiple UIO sequences are adopted [8]. We show that the length of the test sequence satisfying the completeness can be reduced using the cyclic characterization sequence [11]. Conformance test procedure with the proposed test sequence may reduce a tester's job, since the test sequence guarantees the validity of its verdict [1], [4], [6]. Lastly, we illustrate our technique on a specific example.

2. A New Problem Instance

In this section, we describe the situation where the commonly used characterization sequences such as UIO and DS fail to test a correct state transition. Figure 1 shows that the transition (si, sj) is considered faulty due to the output fault or the transition fault even though implemented correctly. That is, an output fault or a transition fault in the characterization sequence can induce the wrong verdict.

Definition 1 (Completeness Criteria)
When a tester makes verdict that a tested edge in an implementation under test is incorrectly implemented compared to the specification FSM, actually the edge is not conformed to the specification. This is called completeness criteria.

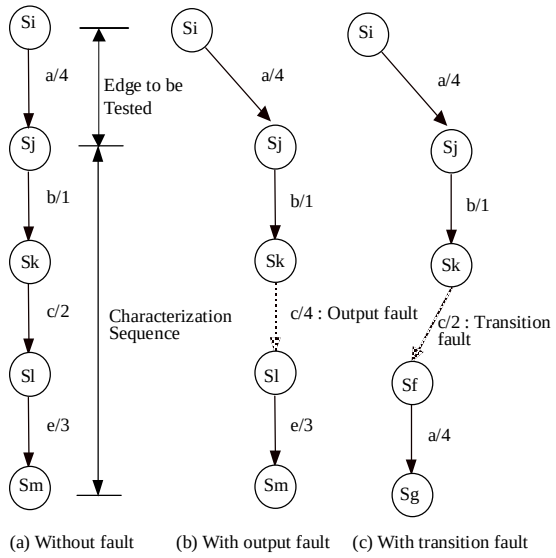

Figure 1. Incorrect behavior of commonly used test sequences

Thus, consideration on the completeness criteria should be taken into a generation of test sequence.

3. A Test Sequence Considering the Problem

We briefly explain the test sequence which can verify the correct state transition here. A key observation is that a characterization sequence whose input/output behavior is completely different from other state's can discriminate whether the output fault arises in the segment being tested or in the characterization sequence. So the new test sequence can correctly verdict the conformity of the implementation. Now, we describe the test sequence with more detail.

Definition 2 (Degree of Difference)
Degree of Difference between a walk A and a walk B, DoD(A, B) is defined as the number of output differences between A and B, where A and B have the same input sequences. In case A and B have the different input sequences, then DoD(A,B) = ∞. In counting the number of differences, previously visited transitions in loops are not counted, since it already contributes to DoD.

Definition 3 (*k-strong* FSM)
An FSM is called *k-strong* for a characterization sequence CS if there exists any characterization sequence CS which satisfies

$$2k+1 <= \text{Min}(\text{DoD}(CS, W)) < 2k+3$$

for all walks W which have the same input sequence as CS in the FSM. (1)

Intuitively, at most k output faults in verifying sequence do not affect the fault detection capability of the test sequence in testing *k-strong* FSM. This is because k output faults do not make a CS look like the other CS in *k-strong* FSM.

4. Proper CS Selection Criteria for k-strong FSM

W-set, UIO, DS, and their variants can be used as the characterization sequence satisfying (1). For each characterization sequence, *k-strong* FSM may exist or not. For example, UIO and DS which satisfy the condition (1) require rather strict conditions of an FSM.

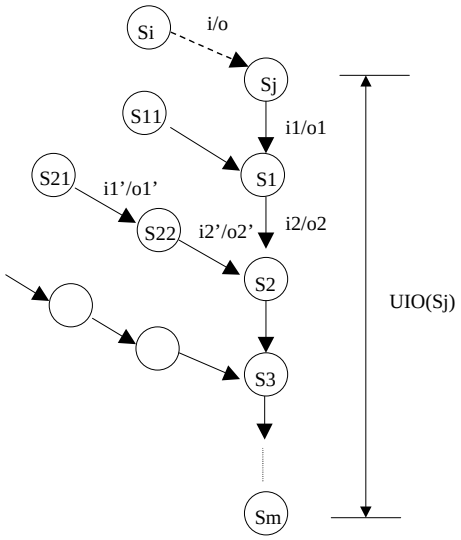

Figure 2. Illustration of Converging walks

Definition 4 (Converging walks)
Converging walks, W1 and W2 are those walks which converge into the same state with the same input sequence. In this case, the output sequence is not considered. Assume that the edge (Si, Sj) is to be tested in Figure 2. For example, (Sj, S1, S2; i1/o1 i2/o2) and (S21, S22, S2; i1'/o1' i2'/o2') converge into the same state S2. If i1 = i1' and i2 = i2', then they are called converging walks and the state S2 is a convergent state.

For a CS to make an FSM *k-strong*, following criteria should be satisfied.

For n = 1, 2, 3, … , 2k
n-th(CS(Sj)) ≠ i/o for all i/o which incomes to δ(Sj, *n*-th(CS(Sj)))
δ(s, i) : Transition function
n-th(A) : *n*-th transition of walk A

For n = 2k+1, 2k+2, …
2k+1 ≤ DoD(*n*-Head(CS(Sj)), A) < 2k+3
for all A, where "A" is an incoming I/O sequence into Sn whose length is *n.*
n-Head(X) : First n I/O sequences of walk X

The first part of the criteria explains that the *k-strong* FSM does not have converging walks whose length is less than or equal to 2k. Because there is less than 2k output results when the k numbers of faults occur, we cannot know where the current state is, nor the correctness of the edge being tested. The second part of the criteria says that CS which makes an FSM *k-strong* must have at least 2k+1 different outputs. If the CS has 2k+3 output differences with all other converging walks, it makes an FSM k+1 strong. Because there is more than 2k+1 output results, there should be a UIO sequence which makes it different from all other walks. Thus, we know where the current state is and where the fault occurs, even when k faults occur.

In the information theoretical sense, this criteria is similar to the error correction capability of Hamming code. Proof of the above criteria can be constructed with the similar way of the error correction capability of the Hamming code [10].

Testing with *k-strong* FSM is performed as following:

The edge (si, sj) is correctly implemented when

DoD(CSsj, Expected Sequence) <= k,

where CSsj is the characterization sequence for a state sj.

The above passing criteria for an edge is the generalization of all other existing test methods, where k is equal to "0". When k is equal to "0", the FSM under testing is *0-strong* and cannot verdict exactly whether the tested edge is correct or not. However, when k is greater than 0, a tester can decide whether the tested edge is correctly implemented or not even when there are k-output faults in verifying sequence. Or the testing with *k-strong* FSM satisfies the completeness criteria explained in Definition 1.

Like most testing methods, the overall test sequence is composed of

(ri)@(Shortest Path to si)@(i/o)@CSsj,

where ri is a reset input. (2)

Instead of (2), test sequences using graph touring such as RCPT can be used. Following section describes the optimizing technique of the proposed test sequence generation scheme satisfying the completeness criteria.

5. Test Sequence Optimizing Technique

In this section, we observe that our test sequence can be reduced. Our test sequence generation method has the property that it exactly verdicts the correctness of the state verifying sequence. Let's assume that the edge under test is (si, sj; i/o), state verifying sequence of sj is (sj, s1, s2, s3, s4; i1/o1, i2/o2, i3/o3, i4/o4), and the FSM is *1-strong*. The result of the test is (PASS, PASS, PASS, FAIL, PASS). With the result, we are certain that the edge under test is correctly implemented even though there is one output fault in the verifying sequence. Here, note that the other edges do not need to be tested, since our test sequence has the completeness property. More specifically, because there are no converging walks in paths from sj to s4, we do not need to compute s1, s2, s3 and s4's UIOs and test them either.

With the optimization technique, we can improve the time complexity of our test sequence. When there are enough characterization sequences which do not have converging walks in an FSM, we can greatly reduce the overall test sequence length. Edges which are not tested using the method can be tested using classical conformance testing methods. We will illustrate the technique in the following example.

6. Construction of k-strong FSM with Extended UIOs

As explained in the previous section, characterization sequences which make an FSM *k-strong* may be W-set, UIO, DS, and their variants. In this section we propose a characterization sequence using UIO, which makes the FSM *k-strong*.

Generally, minimal UIO sequence is applied in testing an FSM, but in most cases it cannot make an FSM *k-strong*. Thus we need to extend minimal UIO so that it may make the FSM *k-strong*. For this purpose, we suggest the new testing sequences called EUIO. EUIO is constructed by concatenating two UIOs as following:

For the verification of (si, sj),
EUIO = UIO(sj) @ UIO(TAIL(UIO(sj)),

where TAIL(UIO(sj)) means the tail state of i/o sequence
UIO(sj).

EUIO reduces the overall problem space. We don't need to consider all walks in FSM, but only walks which incomes to TAIL(UIO(sj)).

Alternatively, we can use multiple UIO to construct EUIO [8]. By using multiple UIO, there are more chances to make an FSM *k-strong*. Even better, the overall length of a test sequence is shortened.

7. Example k-strong FSM for EUIO

Here, we illustrate the *k-strong* FSM with a sample FSM. For the FSM of figure 1, the edge (Si, Sj;a/4) will be tested using minimal UIO and extended UIO. We will show our scheme satisfies the completeness criteria.

First, minimal UIO for Sj is computed: b/1 c/2 c/2 b/2. Then degree of difference of each state is computed by definition 2:

S1
DoD((Sj, S1), (S4, S1)) = ∞
S2
DoD((Sj, S1, S2), (S5, S4, S2)) = ∞
DoD((Sj, S1, S2), (S4, S3, S2)) = ∞
DoD((Sj, S1, S2), (S4, S1, S2)) = ∞
S3
DoD((Sj, S1, S2, S3), (Sj, S5, S4, S3)) = ∞
DoD((Sj, S1, S2, S3), (S6, S5, S4, S3)) = ∞
S4
DoD((Sj, S1, S2, S3, S4), (S6, Si, S6, S5, S4)) = 4

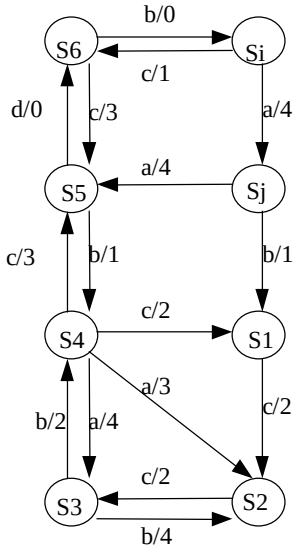

Figure 3. Example of *k-strong* FSM

As seen in the above result, Min(DoD(UIO(Sj), W)) = 4 for all walks W ≠ UIO(Sj) in FSM. Since 2*__1__ + 1 < 4

< 2*__1__ + 3, the FSM is 1-strong for the extended UIO by definition 2.

Now, assume that the edge (S2, S3;c/2) has been incorrectly implemented with an output fault as (S2, S3; c/3). After evaluation of DoDs, output sequences by each method are as following:

Using Minimal UIO,
Expected output sequence: 1 2 2 2
Observed output sequence: 1 2 3 2
=> Verdict as FAIL.
Using Extended UIO,
Expected output sequence: 1 2 2 2
Observed output sequence: 1 2 3 2
=> Since DoD = 1 ≤ 1, Verdict as PASS

In case of the minimal UIO scheme, test sequence could not decide whether the output fault arises in the edge being tested or one of edges in the UIO sequence, whereas our scheme can tell where the fault arises with high probability. Thus, a tester can infer that the edge being tested is conformed to the specification.

Besides the fault correction capability, the length of the test sequence can be reduced. For example, by testing the edge (Si, Sj; a/4), we do not need to test the edges in verifying sequence, (Sj, S1, S2, S3, S4; b/1, c/2, c/2, b/2). Thus, computing UIOs of S1, S2, S3 and S4 is not required.

In this example, since the FSM is 1-strong for the extended UIO, the test sequence has only one fault correction capability.

8. Conclusion

In this paper, we present a problem of commonly used characterization sequences and propose a new test sequence to resolve the problem. The proposed test sequence generation method can be applied to various test sequences such as UIO-, DS-, W-, and their variants, though only a construction strategy using UIO sequence is presented. The proposed test sequence could decide whether the output fault arises in the edge being tested or one of edges in the UIO sequence. Additionally, the fault coverage is much wider than other test sequence generation methods, since the test segment for a tail state verification is longer than others [7].

However, the length of the proposed test sequence is about twice longer than the original one if the simple test sequence construction algorithm is adopted from β-sequence. To resolve this inefficiency, we take advantage of the completeness property of the test sequence.

A test sequence generation method which makes an FSM *k-strong* with higher probability should be studied. The proposed scheme is also applicable to software testing and the conformance testing of sequential circuits.

References

[1] Alfred V. Aho, T. Dahbura, David Lee, M. Umit Uyar, “An Optimization Technique for Protocol Conformance Test Generation Based on UIO Sequences and Rural Chinese Postman Tours,” IEEE Trans. on Commun., vol. 39, no. 11, Nov, 1991, pp 1604-1615

[2] Krishan Sabnani, Anton Dahbura, “A Protocol Test Generation Procedure,” Computer Networks and ISDN Systems, 1988, vol. 15, pp 285-297

[3] Deepinder P. Sidhu, and Ting-Kau Leung, “Formal Methods for Protocol Testing: A Detailed Study,” IEEE Trans. on SE. vol. 15, no. 4, Apr., 1989, pp 413-426

[4] F. Lombardi and Y.-N. Shen, “Evaluation and Improvement o Fault Coverage of Conformance Testing by UIO Sequences,” IEEE Trans. on Commun., vol. 40, no. 8, Aug., 1992, pp 1288-1293

[5] Hasan Ural and Kequin Zhu, “Optimal Length Test Sequence Generation Using Distinguishing Sequences,” IEEE Trans. on Networking, vol. 1, no. 3, Jun., 1993, pp 358-371

[6] D. Rayner, “OSI Conformance Testing,” Computer Networks and ISDN Systems, vol. 14, 1987, pp 79-98

[7] Kshinrasagar Naik, “Fault-tolerant UIO Sequences in Finite State Machines,” IWPTS’95, 1995, pp 207-220

[8] Y. –N. Shen, F. Lombardi, and A. T. Dahbura, “Protocol Conformance Testing Using Multiple UIO Sequences”, IEEE Trans. on Commun., vol 40, no. 8, Aug., 1992, pp 1282-1287

[9] Raymond E. Miller, Fellow, IEEE, and Sanjoy Paul, Member, IEEE, “Structural Analysis of Protocol Specifications and Generation of Maximal Fault Coverage Conformance Test Sequences,” IEEE/ACM Trans. on Networking, vol. 2, no. 5, Oct. 1994, pp 457-470

[10] Robert B. Ash, “Information Theory,” Dover Publications, Inc., 1990, pp 89-90

[11] Samuel T. Chanson and Jinsong Zhu, “A Unified Approach to Protocol Test Sequence Generation,” Proc. In IEEE INFOCOM’93, 1993, pp 106-114