A New Technique for Localization of Multiple Faults in Conformance Testing

2P−1

Hideki YUKITOMO Yoshiaki KAKUDA Shinji KUSUMOTO Tohru KIKUNO

Dept. of Information and Computer Sciences, Faculty of Engineering Science, Osaka University

1 Introduction

Conformance testing for communication protocols is to check whether a protocol implementation conforms to a given protocol specification, and it is performed by observing output sequences outside by giving input sequences. Such input and output sequences are called *test sequences*. Many researchers have developed techniques for generation of *test sequences*. However, the previous techniques can only find the existence of differences (that are called *faults*) between a protocol specification and its implementation. Recently, a test sequence generation technique is proposed in [2], which can localize a fault under the assumption that the implementation includes only one fault.

This paper proposes a new technique for localization of faults by generating test sequences interactively, even when the protocol implementation includes more than one fault.

2 Conformance Testing

This paper assumes that a protocol is modeled as a *deterministic finite-state machine*, so it can be represented by a directed graph where *nodes* and *edges* correspond to *states* and transitions of the protocol, respectively.

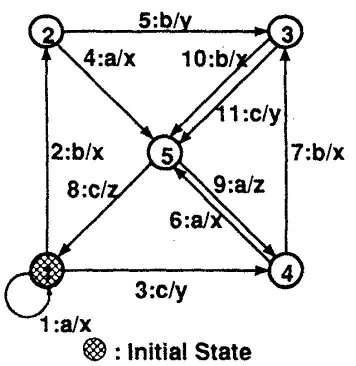

Figure 1 Graph representation of a protocol.

An example of the protocol is shown in Figure 1. Consider an edge from *node* 2 to *node* 3 with a label "5 : b/y". It implies that when the protocol receives an input "b" at state 2, it sends an output "y" and enters state 3. The edge is identified by the number 5 uniquely.

Let G_s and G_i be graphs of protocols defined by specification and its implementation, respectively. Then differences between graphs G_s and G_i of a protocol are called *faults* in the protocol implementation.

Conformance testing for a protocol is to determine whether faults exist in the protocol implementation and to localize faults by observing output sequence for a given input sequence. Such input and output sequences are called *test sequences*.

Faults on a transition T in the protocol implementation are classified into the following two types:

(1) Transfer fault: This fault occurs when a state entered by transition T in protocol implementation G_i is different from a state entered by a corresponding transition in protocol specification G_s.

(2) Output fault: This fault occurs when an output by transition T in protocol implementation G_i is different from an output by the corresponding transition in protocol specification G_s.

However, it is difficult to identify states in the protocol implementation, since they are not observable directly from outside.

A *UIO* sequence of state S is defined as "$i_1/o_1; i_2/o_2; .., i_n/o_n$" such that the protocol sends a sequence of outputs "$o_1; o_2; ...o_n$" only when it receives a sequence of inputs "$i_1; i_2; ...; i_n$" at state S [1]. Here, ";" denotes a concatenation. The *UIO* sequence of state S is abbreviated as UIO_S .

In Figure 1 "$c/y; a/z$" is UIO_1 because the protocol never sends any output sequence "$x; z$" for an input sequence "$b; c$" at any other state except state 1.

Then test sequences can be constructed using *UIO sequence*, to detect the above faults. All transitions in the protocol implementation should be tested by applying test sequences.

3 Localization of Faults

In order to test a transition T, the following procedures (1) ~ (4) are necessary. In the following, let S and D denote *source* and *destination states* of T, respectively, and let "i" and "o" represent input and output for T in the protocol specification.

(1) Set the state S as the current state.

(2) Give an input "i" to the state S.

(3) Check whether an observed output is equal to the expected output "o".

(4) Check whether the next state is an expected destination state D.

Transfer faults and *output faults* can be localized by procedures (3) and (4) respectively, if all transitions associated with UIO_D are correctly implemented according to the protocol specification.

If a *transfer fault* or an *output fault* exists on some transition associated with UIO_D, then it cannot be determined the location where faults occur (that is on

適合性試験において複数フォールトの位置を決定する方法
行友 英記，角田 良明，楠本 真二，菊野 亨
大阪大学基礎工学部情報工学科

the transition T and/or one of transitions associated with UIO_D).

Consider transition 8 in Figure 1. A label input/output is "c/z" and a UIO_1 is "$c/y; a/x$". Then transition 8 in the protocol implementation is tested whether the output and the destination state are equal to "z" and state 1, respectively, as expected in the protocol specification (shown in Figure 1).

If an output sequence "$z; y; x$" is observed for a given input sequence "$c; c; a$", then transition 8 is correctly implemented. However, suppose that an output sequence "$z; y; z$" is observed. In this case there are many possible cases that correspond to faulty implementations. Two of them are shown in Figure 2.

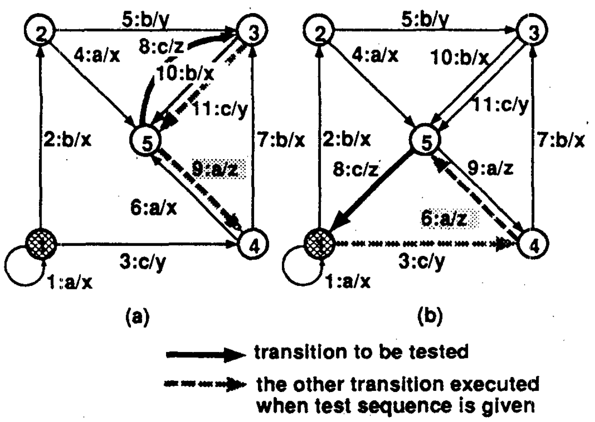

Figure 2 Two types of faulty implementations.

First, as shown in Figure 2(a), a *transfer fault* occurs on transition 8 in the protocol implementation. Because of this fault, the protocol leaves state 5 and enters state 4 via states 3 and 5 against the specification (in Figure 1).

Next as shown in Figure 2(b), an *output fault* occurs on transition 6. The protocol leaves state 5 and enters state 5 again via states 1 and 4 as expected. However, the output sequence "$z; y; z$" is different from "$z; y; x$" in the specification.

As illustrated in the above example, localization of faults is more difficult than determining whether faults exist in the protocol implementation.

4 New Technique

In order to cope with the difficulty of fault localization, all transitions with labels constructing a *UIO* sequence for each transition should be tested before the transition itself is tested.

Let T_0 be a transition to be tested, and $T_1, T_2, \ldots, T_n$ be a sequence of all transitions necessary for test of T_0 (that is, labels of $T_1, T_2, .., T_n$ are used to construct *UIO* sequence for T_0). Let "i_i/o_i" represent a label input/output of transition T_i. So, test sequence for transition T_0 is denoted by "$i_0/o_0; i_1/o_1; i_2/o_2; \ldots; i_n/o_n$".

Using such test sequence, the following two kinds of faults can be localized if $T_1, T_2, \ldots, T_n$ are confirmed to be correctly implemented by their test.

Let O_x be an observed output for input "i_0" and W_x be an output sequence for input sequence "$i_1; i_2; \ldots; i_n$".

(a) If $O_x =$ "o_0" and $W_x \neq$ "$o_1; o_2; \ldots o_n$", then a *transfer fault* on T_0 is localized.

(b) If $O_x \neq$ "o_0" and $W_x =$ "$o_1; o_2; \ldots o_n$", then an *output fault* on T_0 is localized.

For example, consider the faults shown in Figure 2. Because of the assumption that transitions associated with UIO_1 (that is transitions 3 and 6) are confirmed to be correctly implemented, a *transfer fault* on transition 8 in Figure 2(a) is localized.

To realize the above technique for fault localization, test sequences must be interactively generated referring not only the protocol specification but also interim results for $T_1, T_2, \ldots, T_n$.

The outline of the proposed algorithm for the interactive conformance testing is as follows:

(1) Pick up a transition (say T) from the given protocol specification.

(2) Test T using a test sequence for T.

(3) If an output sequence is observed as expected, attach a label "tested" to T.

(4) Otherwise, suspend test of T and, as attribute of T, attach all transitions (say $T_1, T_2, .., T_m$) associated with the *UIO* sequence except those with 133.1.236.35 labels "tested".

(5) If labels "tested" are attached to all the transitions $T_1, T_2, .., T_m$, then resume test of T according to fault localization technique described above. As a result, a fault on T is localized.

(6) If there remain untested transitions, then go to (1). Otherwise stop.

The detail of the algorithm is omitted in this paper due to the limited space and is described in [3].

5 Conclusions

This paper has proposed a new technique for localization of multiple faults in conformance testing. Experimental study will follow this paper for evaluation of availability of the proposed technique as the future research.

References

[1] A. V. Aho, A. T. Dahbura, D. Lee, and M. U. Uyar, "An optimization technique for protocol conformance test generation based on UIO sequences and rural Chinese postman tours", *Proc. of the 9th Int'l Symp. on Protocol Specification Testing and Verification*, pp. 75–86, 1988.

[2] A. Ghedamsi and G. V. Bochmann, "Test result analysis and diagnostics for finite state machines", *Proc. of ICDCS-12*, pp. 244–251, May 1992.

[3] H. Yukitomo, Y. Kakuda, S. Kusumoto, and T. Kikuno, "Interactive conformance testing for localization of multiple faults in communication protocols", *TECHNICAL REPORT OF IEICE. CPSYS92-68,FTS92-41*, pp. 17–24, Dec. 1992, in Japanese.