17
A weighted random walk approach for conformance testing of a system specified as communicating finite state machines

Deukyoon Kang, Sungwon Kang, Myungchul Kim and Sangjo Yoo
Protocol Engineering Team, Technical Standards & Requirements Laboratory
Korea Telecom Research & Development Group
17 Woomyun-dong Sucho-ku, Seoul, Korea
Tel: +82-2-526-5156, Fax: +82-2-526-5567
{dykang, kangsw, mckim, sjyoo}@sava.kotel.co.kr

ABSTRACT

It is very important to test protocol implementations to verify conformance to their specifications (standards) in order to promote interoperability between them. This kind of testing is referred to as conformance testing. For that purpose, a kind of test scenario need prepared in advance and the involved work is called test generation. On the other hand, often a protocol can be specified succinctly and in an understandable way as a collection of communicating finite state machines. In this paper, we propose a test generation scheme called weighted random walk that can be applied to the test generation of communicating finite state machines. The proposed scheme

Formal Description Techniques and Protocol Specification, Testing and Verification
T. Mizuno, N. Shiratori, T. Higashino & A. Togashi (Eds.)

is applied to an example protocol and some results of comparison with existing schemes such as pure random walk and guided random walk are presented. Our scheme is superior to the existing schemes in that it tends to test communicating finite state machines with fewer external test inputs. In an illustrated example in the paper, our scheme shows about 48% improvement over the existing schemes in terms of the number of necessary external test inputs.

Key words
Conformance testing, protocol testing, communicating finite state machines, random walk

1 INTRODUCTION

Conformance testing checks if an implementation conforms to its specification. Especially, in protocol engineering area, it is considered important in order to achieve interoperable networks since they consist of communication equipment from many different vendors implementing the same specification like Q.2931, the ATM user-network signalling protocol standardized by ITU-T.

In order to perform conformance testing on an implementation, test generation should be done first. It is well known that manual test generation is error-prone and very time consuming. Thus, a large amount of research work has been done for automatic test generation from various formal specifications including Finite State Machine (FSM) models. In fact, much research effort was focused on test generation from a single FSM model and produced concrete results [3,4]. However, often communication protocols can be specified more succinctly and in an understandable way as a collection of communicating FSM's [1,2].

We can classify conformance testing into structured testing and non-structured testing. In the case of structured testing, a test is generated based on the structure of a single FSM which is composed from a set of communicating FSM's in terms of I/O behaviour. Otherwise the test is non-structured. Furthermore, conformance testing can be classified into static testing or adaptive testing. In the static testing, a test campaign is carried out based on a pre-determined test sequence. For example, the test generation methods such as TT, UIO and W are for static testing. In the case of adaptive testing, there is no such pre-determined test sequence. Only when the current state of an IUT is known, the next external input to be applied is selected. Adaptive testing is very useful to cope with the difficulties arising from non-deterministic behaviour of the IUT since for a non-deterministic FSM we cannot know in advance which transition would be exercised at a given state.

A naive approach to test generation for a system specified as a collection of communicating FSM's would be to first compose them into a single FSM, then apply to it existing test generation methods such as TT, UIO,

W and so on. But the well-known state explosion problem [1,2,4] would be encountered in the process of composing the communicating FSM's into a single FSM. Thus, structured testing would be difficult to apply in practice. In fact, some test generation approaches were proposed to avoid the state explosion problem. They attempt to test in respect of each of communicating FSM's rather than a single FSM. In other words, they attempt to test each of communicating FSM's separately instead of composing them into a single FSM and then testing the one complex FSM. However, even in this approach, static testing would be impractical due to inherent difficulties to be described in next section.

In this paper, we propose a heuristic test generation scheme for a system specified as a collection of communicating FSM's, which is based on random walk and enhanced with weight information. A closely related idea is the guided random walk approach [2]. But its drawback is to converge to random walk rapidly resulting in inefficient test generation as a test campaign goes on. Meanwhile, our approach keeps working effectively with the help of the weight information.

The paper is organized as follows: in Section 2, background concepts and existing test generation approaches based on random walk are explained briefly. Also fundamental difficulties in conformance testing of communicating FSM's are described. In Section 3, we present in detail our test generation approach called weighted random walk. In Section 4, we show some results of comparison of our approach with existing ones. Finally, the paper concludes with Section 5.

2 BACKGROUND AND PREVIOUS WORK

In this paper, a protocol is specified as a set of communicating FSM's. For convenience, we refer to the protocol simply as a CFSM. Each FSM constituting the CFSM is called a component FSM F_i and defined as follows:

Definition 1. $F_i = (Q_i, M_i, \delta_i, s_0^i)$

- Q_i: *a finite set of states*
- M_i: *a finite set of I/O messages*
- δ_i: *non-deterministic transition function defined as* $Q_i X M_i \rightarrow 2^{Q_i}$
- s_0^i: *initial state of* F_i.

Hence, we represent a component FSM F_i as a directed graph (V, E) where V is a set of states (Q_i) and E is a set of edges connecting a state s_j^i to another state s_m^i. Each edge is labeled with an I/O (M_i).

A CFSM P consisting of k component FSM's is denoted as follows:

Definition 2. $P=(F_0, F_1, F_2, ..., F_{k-1})$

We assume that communication between component FSM's occurs in synchronous manner, that is, a sender is blocked until the message sent is received by a receiver and a receiver is blocked until an expected message is received. The following notation is used to represent the message exchange

between FSM's:

- To send a message M to Process (or FSM) B: B!M
- To receive a message M from Process A: A?M

For a CFSM consisting of k component FSM's, we can build a single FSM equivalent to the CFSM in respect of I/O behavior. The single FSM is referred to as a composite FSM.

In this paper, for conformance testing of a CFSM, we deal with each component FSM instead of a composite FSM derived from the CFSM to avoid the state explosion in the stage of converting the CFSM to the composite FSM. The approach is justified by the following proposition [1]:

> ***Proposition 1:*** *Given a test sequence, if it is a conformance test of each component FSM, then it is also a conformance test of the composite FSM.*

However, there is a difficulty in generating a test sequence systematically for each component FSM due to the following proposition [1]:

> ***Proposition 2:*** *Given two states in a component FSM, it is a PSPACE problem to calculate an exact test sequence leading one state to another considering side effects of other component FSM's.*

Thus, random walk based approaches are viable solutions in the situation. In past years random walk was used for the validation of an FSM [5] and some other validation purposes. However, it is not appropriate to apply to conformance test generation directly since the odds are that it produces a very long test sequence for an even very small FSM. Therefore, P-method and guided random walk were proposed in [2] and [1] respectively.

Both of them are unstructured testing in terms of a composite FSM and are based on random walk. However, the assumptions on component FSM's and the communication between them are different. The P-method assumes deterministic component FSM's and asynchronous communication between them. Whereas, the guided random walk assumes non-deterministic component FSM's and synchronous communication.

It was claimed in [2] that for given a real world protocol, it is very difficult or practically impossible to make a complete state space exploration provided that the protocol is specified as a collection of asynchronous communicating FSM's. Its proposed solution is to test more probable part of a protocol first with the aid of pre-information as to which transitions are more likely to happen. Hence, the P-method generates a set of test sequence prior to a test campaign. Thus, its approach can be said to be based on static testing.

The other scheme [1] is based on adaptive testing to deal with non-determinism. Its goal is different from that of the P-method in that it attempts to cover all transitions of each component FSM and does not generate a fixed test sequence. In this approach, when reaching a state of a CFSM during a test campaign, the next external input is dynamically selected depending on

the state. More specifically, given a state, it divides possible external input transitions into two classes, unvisited transitions and visited transitions. Then it gives higher priority to the transitions in the unvisited class than those in the visited class. Actually, it would not try to traverse transitions belonging to visited class if there exists unvisited transition(s). Notwithstanding it is an improvement on the pure random walk approach, the guided random walk loses its advantage over the pure random walk fast as the test campaign goes on. Because if many transitions are traversed, there are seldom chances to exercise the guided selection of transitions.

3 WEIGHTHED RANDOM WALK

We propose the weighted random walk approach for test generation for a CFSM. The proposed approach performs well even in the situation where the guided random walk loses its advantage over the pure random walk. It is made possible by incorporating pre-knowledge from a specification in the form of weights. In this paper, it is assumed that the minimal requirement of test generation for a CFSM is to traverse at least once all transitions of each component FSM of the CFSM.

3.1 Protocol model and assumptions

The protocol model to be used throughout this paper is basically the same as the one in [1]. It was already described in Definitions 1 and 2 in Section 2. That is, synchronous communicating non-deterministic FSM's. Now we refine the notions of transition and state in those definitions and make some additional assumptions on them.

Transitions are either external or internal. A transition is classified as an external one if it is associated with the environment. Otherwise it is classified as an internal transition. An external transition is denoted as ?M (external input transition) or !M (external output transition). Especially, only external output transitions can be observed and we can exercise limited control to external input transitions. It is limited because the external input transitions may be non-deterministic. For example, for an external input, there can be more than one corresponding transitions and it cannot be known in advance which transition would be exercised. Hence, each external input transition has an integer value, *weight*. The usage of the weight and how to determine its value will be discussed in next subsection.

There are two kinds of states a component state and a global state. A component state is defined as a state of a component FSM. A global state is defined as tuples of component states and represents a unique state of a CFSM. For example, given a CFSM consisting of k component FSM's, a global state s is defined as *k-tuples* $(s^0, s^1,...,s^{k-1})$, where s^i is a component state corresponding to each component FSM of the CFSM. Hence, global states are classified into stable global states and transient global states. Stable global states are states in which the CFSM are waiting for an external input. All

other global states are transient. Transitions can proceed without external inputs in transient global states. We assume that only stable global states can be observed. In other words, when an IUT reaches a stable global state, it would stay there as long as no external inputs are applied to. When an external input is applied to the IUT, it would make some internal input and/or output transitions and/or external output transitions while going through transient global states. Then, it would eventually reach a stable global state. And, we assume that the initial global state can be reached from any global state.

3.2 Overall architecture of weighted random walk

Our proposed testing approach is depicted in Figure 1. Note that there are three procedures such as the selection procedure, the conformity decision procedure and the test manager. The selection procedure takes weight information list as input in order to choose the next external input. The conformity decision procedure determines if observed stable global states and external outputs conform to the specification. The test manger checks if the test termination criteria is satisfied and manages some data structures to keep various information regarding a test campaign. They will be explained in detail through next subsections.

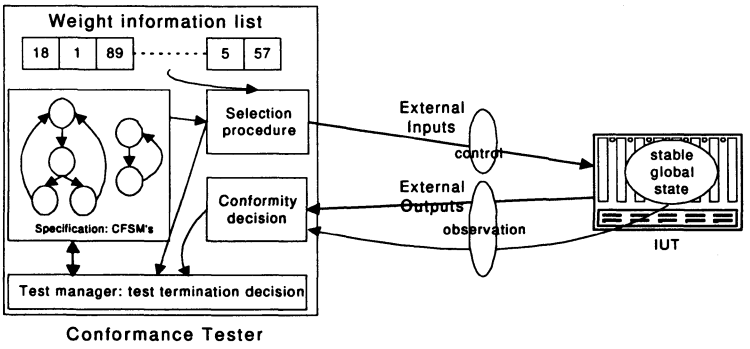

Figure 1 Overall test architecture of weighted random walk.

3.3 Selection of external inputs

When a stable global state is reached, it is very important to select an appropriate external input if more than one external inputs are applicable since it may affect the number of necessary external test inputs significantly. There are three different existing schemes for the purpose such as 1) the pure random walk, 2) the P-method and 3) the guided random walk as mentioned in the previous section.

In our scheme, we introduce an integer value, *weight,* for the decision on which external input transitions would be applied next. We represent it in

a directed graph by augmenting labels on corresponding edges as follows:

?external message (weight value)

We classify external input transitions into *unvisited* and *visited* depending on whether they were already traversed or not. However, we cannot decide for certain that an external input transition was traversed due to non-determinism. This problem will be addressed in next subsection. For a while we assume that we can classify states into *unvisited* and *visited* for certain. Given a state, a transition among the ones in unvisited class would be chosen at random as the next transition before any one in the visited class is chosen. In case that the unvisited class is empty, a transition among ones with the highest weight would be chosen at random from visited class.

The following example shows the basic idea of our approach. In the FSM given in Figure 2, the external input transition *(0,?A,1)* is on a path leading to a bigger behaviour space than the one resulting from the external input transition *(0,?B,4)*. Provided that the FSM is in the state 0 and *(0,?A,1)* and *(0,?B,4)* were already traversed, the guided random walk would choose one of them at random. However, even if we traversed both *(0,?A,1)* and *(0,B,4)*, one of States 2 and 3 may remain untraversed. Thus, it is more reasonable to apply an external input *A* than to apply *B* at State 0 if both *(0,?A,1)* and *(0,?B,4)* were traversed. That is what our approach chooses. However, as we select *(0,?A,1)* repeatedly, its advantage over *(0,?B,4)* decreases. For example, if we traversed *(0,?A,1)* more than a certain number of times, it would be better to give a chance to *(0,?B,4)* next time. Actually, when an external input transition is selected by the associated weight, the weight will be decreased by one if it is greater than *0*. Hence, the mechanism prevents the traversal of a CFSM from being caught in livelock, for example, bouncing back and forth between two global states.

3.4 Weight and weight calculation

In our weighted random walk, weight values should be prepared prior to a test campaign as shown in Figure 1. The weight values should guide selection of the next external input so that the following requirement is satisfied:

> ***Requirement:*** *When we meet with a stable global state with more than one candidate external input transitions, it should guide us to the next global state from which untraversed transitions are likely to be discovered.*

In order to satisfy the above requirement, we define *weight* as follows. Suppose that we have a composite FSM. For an external input transition e_i at the state s_i in the FSM, its weight is defined as the number of different reachable transitions from s_j to s_0, where s_j is a state reached immediately after applying e_i and s_0 is the initial state of the composite FSM. The weight w_{ei} for e_i can be obtained using the following formula:

$$w_{ei} = \sum_{s_k \in rs_j} \rho_k$$

where rs_j is *{x/x ∈ reachable states from s_j without passing through the initial state and x ≠ the initial state}* and ρ_k is the number of outgoing transitions of s_k. Table 1 shows weights obtained using the formula.

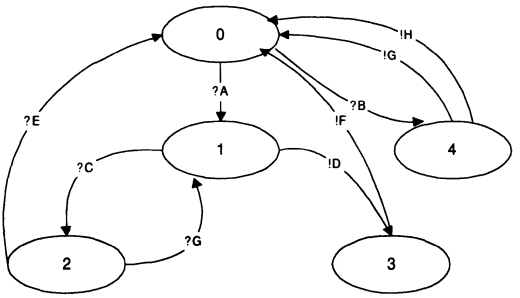

Figure 2 An example FSM.

However, in order to apply the above formula, communicating FSM's should be converted into a composite FSM in advance and we may run into the state-explosion problem again.

Table 1 Weights for the example FSM

Ext. Input transition	Weight	Ext. Input transition	Weight
(0, ?A, 1)	5	*(0, ?B, 4)*	2
(1, ?C, 2)	5	*(2, ?E, 0)*	0
(1, ?D, 3)	1	*(2, ?G, 1)*	5

Thus, instead of using the formula directly, we propose an algorithm that can satisfy the requirement. Let us assume that all transitions in a CFSM have associated weights. From the initial state $(s_0^0, s_0^1, \ldots, s_0^{k-1})$, we can construct a path by concatenating a transition at random iteratively until the path reaches the initial state. In the concatenating process, if we meet with a new transition, weights associated with transitions on the path are increased and the transition is concatenated at the end of the path. Otherwise, the transition is concatenated at the end of the path without an increment of weights. On the other hand, when reaching the initial state, we discard the path and preserves associated weights. We repeat the path construction process until all transitions are traversed. Then, it is obvious that given a weight W_t associated with a transition t, there must be at least W_t transitions reachable via the transition. Thus if a weight associated with an external input transition greater than the weights of other external input transitions, we can claim that the external input transition leads to a bigger behaviour

space than others. Therefore, the weight values obtained as described satisfy *Requirement.*

```
ALGORITHM 1 /* Weights calculation algorithm */
BEGIN
/* the CFSM consist of k component FSM's */
    initialize W; /* W is an integer array of size equal to the number of external
                     input transitions and indexed by a transition. It keeps track of
                     weights of external inputs */
    initialize Q of size MAX_Q; /* Q is a queue to keep external inputs leading the
                                   CFSM from the initial state to the initial state
                                   again */
    cur_state := (s0^0, s0^1,..., s0^(k-1)); /* initial state */

    SUBPROCEDURE transition(tr: a transition)
    BEGIN
        IF tr is an external input transition THEN
            put tr in Q;
        IF tr is a newly traversed transition THEN BEGIN
            FOR all t ∈ a set of transitions in Q DO
                W[t] := W[t]+1;
            mark tr as 'traversed';
        ENDIF;
        update cur_state according to tr;
        IF cur_state = (s0^0, s0^1,..., s0^(k-1)) or the size of Q ≥ MAX_Q THEN
            discard all elements in Q and reset Q;
        RETURN;
    ENDSUBPROCEDURE

    WHILE (there is an untraversed transition) DO BEGIN
        To := {t|t ∈ possible internal and external output transitions at
              cur_state};
        WHILE(To ≠ ∅) DO BEGIN
            to := choose one at random from To;
            transition(to);
            IF to is an internal output transition THEN BEGIN
                ti := input transition matching to;
                transition(ti);
            ENDIF;
            To := {t|t ∈ possible internal and external output transitions at
                  cur_state};
        ENDWHILE;
        /* stable global state */
        Te:= {t|t ∈ possible external input transitions at cur_state};
        te := choose one at random from Te;
        transition(te);
    ENDWHILE
    RETURN(W);
ENDALGORITHM
```

The prescribed sketch is refined in Algorithm 1. Note that the length of a path can be arbitrarily long due to the possible cycles existing in a CFSM. Thus, in the algorithm the length of a path is limited by MAX_Q. Hence, it is not necessary to keep information about internal transitions or about external output transitions. All we need is the information of how many transitions are possible from each external input transition. Thus, we construct and keep a path consisting only of external input transitions. Hence, the path is kept in a queue.

3.5 A conformity decision algorithm

It is essential in testing to decide whether observed states and output transitions from an IUT conform to its specification. Before addressing this problem, let us define spontaneous transitions as follows.

Definition 3: *A spontaneous transition is a transition having an external output or internal input/output.*

For any pair of two states, $(s_{(i-1)}^{j}, s_{(i)}^{j})$ in a component *FSM j*, we can determine if $s_{(i)}^{j}$ is reachable from $s_{(i-1)}^{j}$ through only spontaneous transitions at the cost $O(l_j)$, where l_j is the number of spontaneous transitions in the component *FSM j*. The proof is straightforward and is omitted here. For convenience, in order to denote that $s_{(i)}^{j}$ is reachable from $s_{(i-1)}^{j}$ through only spontaneous transitions, let's use the following notation:

$$s_{(i-1)}^{j} \rightarrow^{*} s_{(i)}^{j}$$

In order to denote that $s_{(i)}^{j}$ is reachable immediately from $s_{(i-1)}^{j}$ after an input or output message *t*, the following notation is used:

$$s_{(i-1)}^{j} \rightarrow^{t} s_{(i)}^{j}$$

Suppose that we observed a global state $S_{(i-1)}$ after applying an external input *Ext* to a component *FSM m* at a global state $S_{(i)}$, where $S_{(i)} = <s_{(i-1)}^{0}, s_{(i-1)}^{1}, \ldots, s_{(i-1)}^{k-1}>$ and $S_{(i-1)} = <s_{(i)}^{0}, s_{(i)}^{1}, \ldots, s_{(i)}^{k-1}>$ respectively. For convenience, we denote it as $(S_{(i-1)}, ?Ext^{(m)}, S_{(i)})$ and refer to it as a stable external input transition. The observed $(S_{(i-1)}, ?Ext^{(m)}, S_{(i)})$ is considered conforming if the following condition is satisfied:

$s_{(i-1)}^{j} \rightarrow^{*} s_{(i)}^{j}$ *for all* j, $0 \le j < k$, $j \neq m$ *and* $s_{(i-1)}^{m} \rightarrow^{Ext} s_{(t)}^{m} \rightarrow^{*} s_{(i)}^{m}$ ····**Condition 1**

For instance, if we observed $(0, ?B^{(0)}, 0)$ in an IUT implementing the FSM in Figure 2, it is considered valid because it satisfies the above condition as follows:

$$0^{(0)} \rightarrow^{?B} 4^{(0)} \rightarrow^{!G} 0^{(0)}$$

In this example, note that $k = 1$, $m = 0$.

On the other hand, for the outputs observed during the stable external input transition $(S_{(i-1)}, ?Ext^{(m)}, S_{(i)})$, we verify if they belong to a set of expected external output transitions. The expected output transitions can be obtained in the following way. First, obtain the set A as follows:

$$A=\{x/x \in s_{(t)}^{j}, \text{ where } s_{(i-1)}^{j} \rightarrow^{*} s_{(t)}^{j} \text{ for all } j,\ 0 \le j < k,\ j \neq m\}$$

Then, obtain the set B as follows:

$$B=\{x/x \in s_{(t)}^{m} \text{ or } s_{(t)}^{m'}, \text{ where } s_{(i-1)}^{m} \rightarrow^{Ext} s_{(t)}^{m}, s_{(t)}^{m} \rightarrow^{*} s_{(t)}^{m'}\}$$

Finally, we obtain *O*, a set of expected outputs, defined as follows:

$O=\{x/x \in$ *External outputs possible at* $s_i^j, s_i^j \in A \cup B\}$ ·····**Condition 2**

For instance, we can calculate A, B and O as follows for $(0, ?B^{(0)}, 0)$ in the example FSM. $A = \{\}$, $B = \{0, 4\}$, $O = \{G, H\}$.

For a set of observed outputs *O'*, if $O' \subseteq O$, then *O'* is conisidered valid.

Otherwise, invalid. Note that we do not concern with the order or the number of occurrences of external outputs.

3.6 Termination criterion

As mentioned in Section 3, given a CFSM, the minimal requirement of conformance testing is to traverse all transitions of the CFSM at least once. So, in our scheme, the termination criterion is to check if all transitions are exercised at least once. However, for all kinds of transitions such as external input transitions, external output transitions, internal input transition and internal output transition, we could not determine for certain that they are exercised since we assumed non-deterministic FSM model. Thus, we can say only with certain confidence level that they are exercised. For the purpose, a real array T is introduced. Each entry of it indicates the possibility that a corresponding transition is exercised. For the proper manipulation of the real array, we define outgoing degree as follows:

> ***Definition 3:*** *For a component state s_i^j of a component FSM j, outgoing degree ρ_i^j is defined as the number of possible spontaneous transitions at the state.*

When observing $(S_{(i-1)}, ?Ext^{(m)}, S_{(i)})$, for each transition at $s_{(t)}^j$, where $s_{(t)}^j \in A \cup B$, we increase the corresponding entry of T by $1/\rho_i^j$. If all entries of T is greater than or equal to a certain thresh-hold value, α, then the termination criteria is considered satisfied and the test campaign ends; where α is given prior to the test campaign by a test operator regarding the characteristics of a system under test.

3.7 The conformance testing procedure of weighted random walk

So far we have described the pieces constituting the weighted random walk approach one by one. The algorithm for a conformance test campaign using those pieces is elaborated in Algorithm 2.

The algorithm works in the following way. External input transitions possible at the current stable global state are classified into *class_0, class_1,* and *class_1_high.* Untraversed external input transitions belong to *class_0* and traversed ones to *class_1.* The ones with the highest weight among external input transitions in *class_1* belongs to *class_1_high.* As we noted already, we cannot determine if external input transitions were traversed for certain due to non-determinism. Nevertheless for the external input transitions with corresponding entries of T are equal to zero, we can assert that they have not been traversed for certain. Thus, we assign those external input transitions to *class_0* and others to *class_1.* When the classification is done, one of external input transitions from *class_0* is selected at random if it is not empty. Otherwise, an external input transition that belongs to *class_1_high* is selected at random. Then, the corresponding external input is

applied to the IUT. Note that it is not guaranteed that the applied external input results in the execution of the selected external input transition. It is this reason that we do not update the *class_0* and *class_1* at the time when the selection procedure occurs in Algorithm 2. The classification occurs immediately after updating the entries of *T*. The whole process is iterated until the termination criterion is satisfied.

```
ALGORITHM 2 /* the CFSM are assumed to consist of k component FSM's */
BEGIN
    /* W is a integer array to keep weights and indexed by an external input
       transition id; for e.g, W[t]. T is a real array to keep confidence level of
       traversal of a transition and indexed by a transition id; for e.g, T[t] */
    W := call ALGORITHM 1;
    initialize T;
    /* class_0, class_1, and class_1_high are sets. See Section 3.7 */
    class_0 := ∅; class_1 := ∅; class_1_high := ∅;
    observed_outputs := ∅;
    class_0:={t/t ∈ possible external input transitions at the initial state};
    cur_state := (s0^0, s0^1,..., s0^(k-1));
    WHILE termination criteria is not satisfied DO BEGIN /* see Section 3.6 */
        IF class_0 ≠ ∅ THEN
            sel_t := select one at random from class_0;
        ELSE BEGIN
            class_1_high := {t/t ∈ class_1 and ∀t' ∈ class_1 ≤ t};
            sel_t := select one at random from class_1_high;
            IF W[sel_t] > 0 THEN
        /* note that the selected external input transition may not be exercised
           in the IUT due to non-determinism. but we consider it exercised */
                W[sel_t] := W[sel_t] - 1;
        ENDELSE;
        apply to the IUT the corresponding external input for sel_t;
        observe the next stable global state and outputs;
        pre_state := current_state;
        cur_state := the observed global state;
        observed_outputs:= the observed outputs;
        expected_outputs:= calculate expected outputs regarding pre_state,
            cur_state and the selected external input;
        FOR regarding pre_state, cur_state and sel_t DO BEGIN
            IF cur_state is not reachable from pre_state THEN
                RETURN(fault); /* see Condition 1*/
            IF observed_outputs ⊄ expected_outputs THEN
                RETURN(fault); /* see Condition 2*/
            update involved entries of T properly;
            class_0 := {t/T[t]=0 and t is a valid external input transition
                at cur_state};
            class_1 := {t/T[t]>0 and t is a valid external input transition
                at cur_state};
        ENDFOR
    ENDWHILE;
    RETURN(ok);
ENDALGORITHM;
```

There is the possibility that the termination criteria cannot be satisfied due to faults existing in an IUT. To solve this problem, one may incorporate the progress observers proposed in [1]. Their role is to check if the test campaign makes a progress in a sense that new transitions are being traversed. For example, if some entries of T are zero and external inputs have been

applied more than a certain number of times, we can declare that a fault may exist in the IUT.

4 Experiments and comparison with the previous approaches

Including ours, there are now four random-walk based testing schemes. They are (1) pure random walk, (2) P-method, (3) guided random walk and (4) weighted random walk. Among them, P-method is quite different from the others in that it is based on asynchronous deterministic communicating FSM's model while the others are based on synchronous non-deterministic communicating FSM's model. Hence, P-method can be classified as static testing while the others are adaptive. Thus, we only consider the schemes (1), (3) and (4) for comparison.

For comparison, we implemented the simulators for (1), (3) and (4) with the C language on UNIX based workstation. For the purpose, we use the example protocol in Figure 2 as a specification as well as an IUT. The weight values for each of external input transitions were obtained by iterating Algorithm 1 ten times and taking the mean values. The results were calculated in terms of the number of external inputs required to traverse all transitions of each component FSM.

In case of a real IUT, the odds is very low that we can observe internal transitions as assumed in Section 3. However, for our comparison purpose, without losing generality all internal transitions can be assumed to be observable.

For an example protocol in Figure 3, external inputs were applied until all transitions of each component FSM are covered. This was done for each of the three schemes. Furthermore, each scheme was iterated 500 times with different random number seeds to get fair results. The comparison results are shown in Table 2 and Figure 4.

Table 2 shows that our scheme is superior to the two existing schemes by about 48%. Weighted random walk covers all transitions of each component FSM with the average of 29 external inputs while the pure random walk and the guided random walk cover them with the average of 57 and 56 external inputs, respectively.

Table 2 Results of the three schemes

Scheme	Avg. Number of external inputs
Pure random walk	57
Guided random walk	56
Weighted random walk	29

From Figure 4, we can observe that when the number of external inputs is between 1 and 4, all three schemes show similar performance with respect to the number of untraversed transitions and the number of applied

external inputs since most of the transitions remain untraversed.

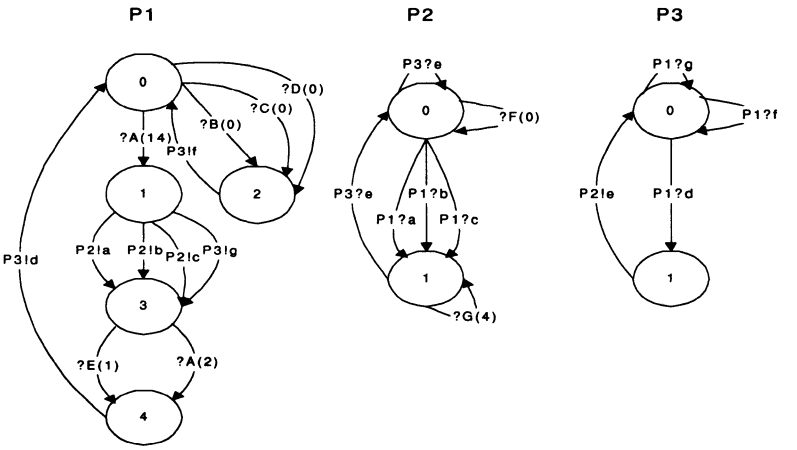

Figure 3 An example protocol.

Where the number of applied external inputs is between 4 and 8, the guided random walk and the weighted random walk are superior to the pure random walk because the former two schemes pay attention to untraversed transitions and their advantage begins to show up. Hence, the two schemes show similar performance in the range.

When the number of applied external inputs is more than 16, the guided random walk and the pure random walk show similar performance. Now the advantage of the guided random walk diminishes since that it has already achieved high traversal ratio. However, we can see that the weighted random walk still performs well by exploiting weight information.

To accomplish higher than 95% coverage in terms of the number of traversed transitions, 24 external inputs were necessary in the case of weighted random walk while 48 and 46 external inputs were necessary in the case of the pure random walk and the guided random walk, respectively. In order to achieve high coverage by random walk based schemes in conformance testing of communicating FSM's, we believe that the pre-knowledge about a CFSM such as the weight information plays an essential role.

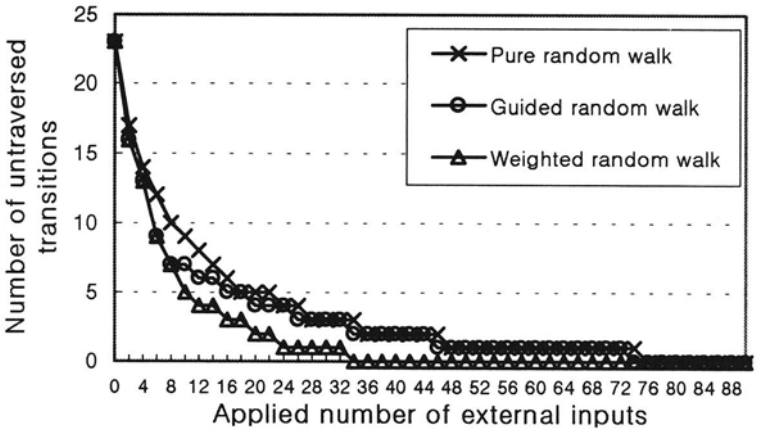

Figure 4 Comparison results of the three approaches.

5 Conclusion

In this paper, we developed a weighted random walk testing scheme for the conformance testing of a system specified as a collection of communicating FSM's. Also, we developed a heuristic method to obtain weight information which is again based on random walk. For the example protocol in Figure 3, it was shown that the proposed scheme shows about 48% improvement over the existing schemes in terms of the number of external inputs required to cover transitions of each component FSM constituting the protocol. Especially, it was addressed that our weighted random walk is expected to achieve high coverage.

The adaptive random walk approach like the guided random walk [1] and ours may at first sight look unrealistic when compared with the methodology ISO 9646 [6]. However, the conformance testing based on the ISO 9646 has the following problems:

- It is very tedious and time consuming to derive test cases from specifications.
- The size of test suites tends to be so big that it is very difficult to validate them to satisfaction and hence impractical to make them standard documents

We believe that our approach is very promising to overcome such problems. For it is straightforward to derive a specification in a set of communicating FSM's from the original protocol specification, for example, from a specification written in SDL (Specification and Description Language). Hence, we can validate the specification with various formal methods. In fact, the example protocol in this paper was validated through random walk.

For further work, we plan to apply our weighted random walk scheme to a real world protocol and to conduct a fault coverage analysis of the scheme.

Also, we look forward to implementing a test software for it on a commercial protocol tester.

6 REFERENCES

[1] D. Lee, K. Sabnani, D. M. Kristol and S. Paul, "Conformance testing of protocols specified as communicating finite state machines - A guided random walk based approach", *IEEE Trans. on Communication,* Vol.44, No. 5, May 1996.

[2] A. Chung and D. Sidhu, "Fault coverage of probabilistic test sequence", In Proc. *3rd International workshop on protocol test systems*, November 1990.

[3] D. Sidhu and T. Leung, "Formal methods for protocol testing: A detailed study", *IEEE Trans. on Software engineering,* Vol.15, No.4, April 1989.

[4] D. Lee and M. Yannakakis, "Principles and methods of testing finite state machines - A survey", Technical report, AT&T Bell Labs., September, 1995.

[5] C. West, "Protocol validation by random state exploration", In Proc. *6th International Workshop Protocol specification, testing, and verification*, 1986.

[6] ISO DIS 9646, Conformance Testing Methodology and Framework, part 1, December, 1989.

7 BIOGRAPHY

Deukyoon Kang received B.A. in electronics engineering from Kumoh Nat'l Institute of Technology in 1993 and M.S. in computer science from Pohang Institute of Science and Technology in 1995. Currently he is with Korea Telecom R&D Group as a member of technical staff. He is involved in the design and implementation of a protocol test system for the ATM/B-ISDN protocol family.

Sungwon Kang received a B.A. from Seoul National University in Korea in 1982 and received M.S. and Ph.D. in computer science from the University of Iowa in U.S.A in 1989 and 1992. Since 1993, he has been a senior researcher at Korea Telecom R&D Group. In 1995-1996, he was a guest researcher at National Institute of Standards and Technology of U.S.A. In 1997, he was co-chair of the 10th International Workshop on Testing of Communicating Systems. Currently he is the head of the Protocol Engineering Team at Korea Telecom R&D Group. His research interests include communication protocol testing, program optimization and programming languages.

Myungchul Kim received B.A. in electronics engineering from Ajou Univ. in 1982, M.S. in computer science from the Korea Advanced Institute of Science and Technology in 1984, and Ph.D. in computer science from the Univ. of British Columbia in 1992. Since 1984, he has been working for Korea Telecom. In 1997, he was co-chair of the 10th International Workshop on Testing of Communicating Systems. Currently he is the managing director of Testing Technology Research Section at Korea Telecom R&D Group and chairman of Profile Test Specifications - Special Interest Group of Asia-Oceania Workshop. His interests include protocol engineering on multimedia and telecommunications.

Sangjo Yoo received B.A. in electric communication engineering from Hanyang Univ. in 1988 and M.S. in electrical engineering from the Korea Advanced Institute of Science and Technology in 1990. Currently he is with the Korea Telecom R&D Group as a member of technical staff.