Hindawi
Security and Communication Networks
Volume 2018, Article ID 3723691, 13 pages
https://doi.org/10.1155/2018/3723691

Research Article

Test Sequence Reduction of Wireless Protocol Conformance Testing to Internet of Things

Weiwei Lin,[1,2,3] **Hongwei Zeng**,[1] **Honghao Gao**,[4,5] **Huaikou Miao**,[1] **and Xiaolin Wang**[1,2]

[1] *School of Computer Engineering and Science, Shanghai University, Shanghai 200444, China*
[2] *Shanghai Key Laboratory of Computer Software Testing and Evaluating, Shanghai 201114, China*
[3] *Department of Information Science and Technology, Taishan University, Taian 271000, China*
[4] *Computing Center, Shanghai University, Shanghai 200444, China*
[5] *Shanghai Key Laboratory of Intelligent Manufacturing and Robotics, Shanghai 200072, China*

Correspondence should be addressed to Honghao Gao; gaohonghao@shu.edu.cn

Received 23 August 2018; Accepted 2 October 2018; Published 1 November 2018

Guest Editor: Lianyong Qi

Wireless communication protocols are indispensable in Internet of Things (IoT), which refer to rules and conventions that must be followed by both entities to complete wireless communication or service. Wireless protocol conformance testing concerns an effective way to judge whether a wireless protocol is carried out as expected. Starting from existing test sequence generation methods in conformance testing, an improved method based on overlapping by invertibility and multiple unique input/output (UIO) sequences is proposed in this paper. The method is accomplished in two steps: first, maximum-length invertibility-dependent overlapping sequences (IDOSs) are constructed, then a minimum-length rural postman tour covering the just constructed set of maximum-length IDOSs is generated and a test sequence is extracted from the tour. The soundness and effectiveness of the method are analyzed. Theory and experiment show that desirable test sequences can be yielded by the proposed method to reveal violations of wireless communication protocols in IoT.

1. Introduction

Wireless communication is essential and critical to Internet of Things (IoT) [1, 2]. The reliability of wireless communication transmission largely depends on whether the wireless communication protocol is implemented as specified. Conformance testing [3, 4] is widely used to check whether an implementation conforms to its specification in areas such as traditional communication protocols and reactive systems; i.e., there must be the same behavior in the implementation for any I/O behavior observed in the specification. In FSM-based conformance testing, a protocol is called a specification and is expressed as a Finite State Machine (FSM), while an implementation under test is considered to be a "black box", the I/O behavior of which can only be observed. A test sequence is an I/O sequence such that whether an implementation conforms to the specification may be concluded by delivering the input sequence of the test sequence to the implementation and comparing the resulting output sequence with the output sequence in the test sequence.

Test techniques based on state identification [5–10] are well-known in FSM-based conformance testing. Test techniques based on state identification are supposed to identify every state and verify every transition of the specification in the implementation. A state is said to be identified when a state identifier is delivered to the state. A transition is said to be verified when the end state of the transition is identified. A state identifier of a state is a nonempty set of input sequences for the state such that the set of corresponding output sequences can characterize the state. Unique input/output (UIO) sequence is a popular state identifier such that UIO-based techniques are commonly used for test sequence generation.

Since wireless communication protocols can also be modeled by FSMs, FSM-based conformance testing is also applicable to wireless protocol conformance testing in

Table 1: UIO sequence of FSM corresponding to the connection and release process of Zigbee.

No.	States	UIO(s)
Ex1	s_1	i_1/o_1
Ex2	s_2	i_6/o_1
Ex3	s_3	i_5i_6/o_1
Ex4	s_4	i_2i_3/o_1o_2 i_2i_4/o_1o_3

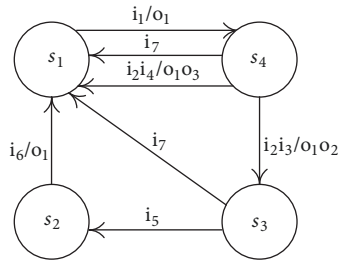

Figure 1: FSM corresponding to the connection and release process of Zigbee.

IoT. Zigbee is a typical wireless communication protocol and is divided into four layers: physical layer (PHY), media access control lay (MAC), network layer (NWK), and application layer (APL). The connection and release process of nodes in the MAC layer is modeled by the FSM in Figure 1 and UIO sequences of every state are listed in Table 1. On this basis, test sequences based on FSMs can be constructed using UIO sequences as state identifiers. There is a test sequence $s_1(i_1/o_1)s_4(i_2i_3/o_1o_2)s_3(i_5)s_2(i_6/o_1)s_1(i_1/o_1)s_4(i_2i_4/o_1o_3)s_1(i_1/o_1)s_4(i_7)s_1(i_1/o_1)s_4(i_2i_3/o_1o_2)s_3(i_7)s_1(i_1/o_1)s_4$ of the FSM in Figure 1 based on UIO sequences in Table 1. When the input sequence of this test sequence is applied to an implementation, every state of the FSM in Figure 1 can be identified and every transition of the FSM in Figure 1 can be verified in the implementation. Meanwhile, an output sequence can be obtained. It can be concluded whether the connection and release process of nodes is executed as the specification specifies by comparing the obtained output sequence with the expected one in the test sequence.

Test sequence reduction has long been an active research topic in FSM-based conformance testing. One approach is to convert the problem into the Rural Chinese Postman Problem from which a test sequence is extracted. For the purpose of transition verification every transition of an FSM is followed by an appended UIO sequence in the test sequence. Bo Yang et al. reduced test sequences by overlapping and multiple UIO sequences [11]. Benefiting from overlapping, more than one transition may be verified by a single appended UIO sequence. Hierons improved the method of UIO sequences through the use of an invertibility criterion, thereby achieving more overlapping [12], i.e., verifying even more transitions of an FSM with a single appended UIO sequence. However, multiple UIO sequences which are conducive to test sequence reduction are not considered by Hierons.

In order to reduce the cost of testing while assuring the effectiveness, this paper presents an improved method to generate reduced test sequences for wireless protocol conformance testing of IoT. Transitions in a test sequence are of three kinds: a copy of transitions in an FSM, UIO sequences which have been appended to the test sequence for the purpose of transition verification for the FSM, and the transitions which have been appended to the test sequence. In the improved method, invertibility is taken into account to leverage as much overlapping as possible such that all the transitions of an FSM can be verified with as few appended UIO sequences as possible. Rural symmetric augmentation is the main measure for test sequence generation. The replicated transitions during the rural symmetric augmentation are exactly the transitions for concatenation of test subsequences in the test sequence. More options of rural symmetric augmentation can be supplied by multiple UIO sequences than those supplied by single UIO sequences; i.e., a sensible choice of UIO sequences can lead to a minimum number of replicated transitions during a rural symmetric augmentation, and a minimum number of transitions for concatenation are achieved in the test sequence. In this way, further reduced test sequences are obtained by the proposed method.

The rest of the paper is organized as follows. In Section 2, the basic concepts and assumptions used in this paper are presented. The improved method is described with simple examples in Section 3, and the soundness and effectiveness of the method are discussed. Experimental evaluation is set out in Section 4. Then, the related work about the testing of IoT is reviewed in Section 5 and the paper is concluded briefly in Section 6.

2. Preliminaries

In this section, we introduce the definitions related to FSMs and graphs, together with the assumptions necessary for FSM-based conformance testing.

2.1. Definitions. An FSM is formally defined as a 6-tuple $M = (I, O, S, s_0, \delta, \lambda)$ where

(i) I and O are finite and nonempty sets of input symbols and output symbols, respectively;

(ii) S is a finite and nonempty set of states;

(iii) $s_0 \in S$ represents the initial state of M;

Table 2: UIO sequence of M_1.

No.	States	UIO(s)
Ex1	s_1	$a/0$
Ex2	s_2	$b/1$ $(a/1)(a/1)$ $(a/1)(b/1)$
Ex3	s_3	$(a/1)(a/0)$

(iv) $\delta : S \times I \longrightarrow S$ denotes the state transition function;

(v) $\lambda : S \times I \longrightarrow O$ is the output function.

According to this definition, when an input symbol i is delivered to the current state s, M moves to the state $s' = \delta(s, i)$ with an output produced by $\lambda(s, i)$. The transition function and output function can be extended to finite input sequences, i.e., for an input symbol i, an input sequence $\beta \in I^*$ (where I^* is the set of finite sequences of input symbols), and a state s, $\delta(s, i\beta) = \delta(\delta(s, i), \beta)$, and $\lambda(s, i\beta) = \lambda(s, i)\lambda(\delta(s, i), \beta)$ where concatenation is denoted by juxtaposition.

A transition t is defined by a tuple $(s, s', i/o)$ where s is the start state, i is an input of s, $o = \lambda(s, i)$ is the associated output, and $s' = \delta(s, i)$ is the end state. A transition $(s, s', i/o)$ is invertible if it is the unique transition ending at s' with input i and output o.

A UIO sequence of a state is an input/output sequence such that the input/output behavior exhibited by the state is unique, i.e., given a UIO sequence β of a state s, and the input sequence of β is denoted by β_{in}; for any state $s' \neq s$, it is always true that $\lambda(s, \beta_{in}) \neq \lambda(s', \beta_{in})$. A UIO sequence is irreducible if it is not a UIO sequence anymore when the end i/o symbol of the sequence is deleted. UIO sequences in this paper will be supposed irreducible unless otherwise specified. There may be more than one UIO sequence for a state and the UIO sequences may be of different length.

An example FSM M_1 is described in Figure 2 where $S = \{s_1, s_2, s_3\}$, $I = \{a, b\}$, $O = \{0, 1\}$ and s_1 is the initial state of M_1. There are four invertible transitions $(s_1, s_2, a/0)$, $(s_2, s_2, a/1)$, $(s_2, s_3, b/1)$, and $(s_3, s_1, a/1)$. The UIO sequences for every state are shown in Table 2.

An FSM M can be perceived as a labeled, directed graph G. A state of M is represented as a node of G, and there is an edge labeled with i/o from node s to s' in G if and only if $\delta(s, i) = s'$ and $\lambda(s, i) = o$ in M, where s and s' are the start and end node of the edge. The numbers of incoming and outgoing edges of a node are called the in-degree and out-degree of the node, respectively. If the in-degree equals out-degree at each node then the directed graph is symmetric. Suppose that G is an asymmetric directed graph; if G^* is a symmetric directed graph generated from G by making copies of the edges then G^* is a symmetric augmentation of G. When the total cost of copies of edges is minimized, G^* is said to be a minimal symmetric augmentation of G. Suppose that E is the set of edges of G and $E' \subseteq E$, if G^* is a symmetric directed graph generated from G by making copies of the edges such that each edge in E' is included in G^* at least once and the total cost of copies of edges is minimized then G^* is called a rural

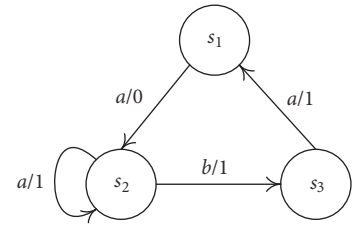

Figure 2: FSM M_1.

symmetric augmentation of G. A sequence of contiguous edges $\beta = (s_1, s_2, i_1/o_1)(s_2, s_3, i_2/o_2) \ldots (s_{k-1}, s_k, i_{k-1}/o_{k-1})$ forms a path of G where s_1 and s_k are the start and end node of the path, respectively. A path that starts and ends at the same node forms a tour, furthermore, if the tour traverses every edge of G exactly once then it is an Euler tour. A rural postman tour is a tour that traverses a given set of edges at least once. The Rural Chinese Postman Problem is to find a minimum-length rural postman tour for a given set of edges. FSMs and their directed graph representations are used interchangeably throughout this paper.

2.2. Assumptions. Given a specification FSM M with n states, the fault domain $\Psi(I)$ of M is the set of all possible implementations of M over the input alphabet I of M. $\Psi_n(I)$ refers to the implementations with up to n states in $\Psi(I)$. This paper only focuses on implementations in $\Psi_n(I)$.

An FSM M is strongly connected if for any two distinct states s and s' there is an input sequence β that takes M from s to s'; i.e., $\forall s, s' \in S, s \neq s', \exists \beta \in I^* \cdot (\delta(s, \beta) = s')$.

Two states s and s' are equivalent if for any input sequence there are always the same output sequences from s and s'. An FSM M without equivalent states is said to be minimal or reduced; otherwise M is reducible by joining equivalent states.

An FSM M is deterministic if at any state for any input there is at most one transition leading to the next state. Otherwise, M is nondeterministic.

Only strongly connected, minimal and deterministic FSMs are considered in this paper. In addition, it is assumed that UIO sequences for each state of an FSM are available and are derived from successor trees in advance. It is noted that only state transition functions in forms of $S \times I \longrightarrow S$ are considered; i.e., state transitions without any input in IoT are out of the scope of this paper.

3. Test Sequence Reduction

3.1. Key Properties of the Method

3.1.1. Overlapping by Invertibility

Definition 1 (invertibility-dependent UIO sequence). If a transition $(s, s', i/o)$ is invertible and β is a UIO sequence of s', then $(i/o)\cdot\beta$ is an invertibility-dependent UIO sequence of s.

The transition $(s_1, s_2, a/0)$ of M_1 in Figure 2 is invertible and $(a/1)(b/1)$ is a UIO sequence of s_2, then $(a/0)(a/1)(b/1)$ is an invertibility-dependent UIO sequence of s_1.

Definition 2 (invertibility-dependent overlapping sequence). Given a transition sequence $(s_1, s_2, i_1/o_1)(s_2, s_3, i_2/o_2)\dots(s_j, s_{j+1}, i_j/o_j)$ and a UIO sequence β of s_{j+1}, if every transition $(s_k, s_{k+1}, i_k/o_k)(1 \leqslant k \leqslant j-1)$ is verified by an invertibility-dependent UIO sequence $(i_{k+1}/o_{k+1}) \cdot (i_{k+2}/o_{k+2})\dots(i_j/o_j) \cdot \beta$ when $(s_j, s_{j+1}, i_j/o_j)$ is verified by β, then $(s_1, s_2, i_1/o_1), (s_2, s_3, i_2/o_2)\dots(s_j, s_{j+1}, i_j/o_j)$ is an invertibility-dependent overlapping sequence (IDOS).

There is a transition sequence $s_1(a/0)s_2(a/1)s_2(b/1)s_3(a/1)s_1$ of M_1 in Figure 2 and a UIO sequence $a/0$ of s_1. When the last transition $(s_3, s_1, a/1)$ of $s_1(a/0)s_2(a/1)s_2(b/1)s_3(a/1)s_1$ is verified by the UIO sequence $a/0$ of s_1, by working backward $(s_2, s_3, b/1)$ is verified by $(a/1)(a/0)$, $(s_2, s_2, a/1)$ is verified by $(b/1)(a/1)(a/0)$, and $(s_1, s_2, a/0)$ is verified by $(a/1)(b/1)(a/1)(a/0)$; i.e., all the other transitions except the last one are verified by invertibility-dependent UIO sequences. As a result, $s_1(a/0)s_2(a/1)s_2(b/1)s_3(a/1)s_1$ is an IDOS.

For $(i_{k+1}/o_{k+1}) \cdot (i_{k+2}/o_{k+2})\dots(i_j/o_j) \cdot \beta$ $(1 \leqslant k \leqslant j-1)$ to be invertibility-dependent UIO sequences, indispensable requirements for transitions in $(s_1, s_2, i_1/o_1)(s_2, s_3, i_2/o_2)\dots(s_j, s_{j+1}, i_j/o_j)$ are put forward.

Theorem 3. *Given a transition sequence $(s_1, s_2, i_1/o_1)(s_2, s_3, i_2/o_2)\dots(s_j, s_{j+1}, i_j/o_j)$ and a UIO sequence β of s_{j+1}, $(s_1, s_2, i_1/o_1)(s_2, s_3, i_2/o_2)\dots(s_j, s_{j+1}, i_j/o_j)$ is an IDOS if and only if every transition $(s_k, s_{k+1}, i_k/o_k)$ $(2 \leqslant k \leqslant j)$ is invertible.*

Proof. The sufficiency of the condition is proved inductively by working backward. In the base case when $k = j$, $(s_{j-1}, s_j, i_{j-1}/o_{j-1})$ is sure to be verified by an invertibility-dependent UIO sequence $(i_j/o_j) \cdot \beta$ since $(s_j, s_{j+1}, i_j/o_j)$ is invertible. In the inductive step, assume that $(s_{k-1}, s_k, i_{k-1}/o_{k-1})$ $(3 \leqslant k \leqslant j-1)$ is verified by an invertibility-dependent UIO sequence γ, then $(s_{k-2}, s_{k-1}, i_{k-2}/o_{k-2})$ is definitely verified by an invertibility-dependent UIO sequence $(i_{k-1}/o_{k-1}) \cdot \gamma$ since $(s_{k-1}, s_k, i_{k-1}/o_{k-1})$ is invertible. In this way, every transition $(s_k, s_{k+1}, i_k/o_k)$ $(1 \leqslant k \leqslant j-1)$ is backward verified inductively by an invertibility-dependent UIO sequence $(i_{k+1}/o_{k+1}) \cdot (i_{k+2}/o_{k+2})\dots(i_j/o_j) \cdot \beta$ when $(s_j, s_{j+1}, i_j/o_j)$ is verified by β. Thus, it is a sufficient condition for a transition sequence $(s_1, s_2, i_1/o_1)(s_2, s_3, i_2/o_2)\dots(s_j, s_{j+1}, i_j/o_j)$ to be an IDOS that every transition $(s_k, s_{k+1}, i_k/o_k)$ $(2 \leqslant k \leqslant j)$ is invertible.

Next, the necessity of the condition is proved by contradiction. As shown in Figure 3, suppose that $(s_1, s_2, i_1/o_1)(s_2, s_3, i_2/o_2)\dots(s_j, s_{j+1}, i_j/o_j)$ is an IDOS with a noninvertible transition $(s_k, s_{k+1}, i_k/o_k)$ $(2 \leqslant k \leqslant j)$; i.e., there is another transition $(s_{k'}, s_{k+1}, i_k/o_k)$ ending at s_{k+1} with the same input and output. Obviously, s_k cannot be identified by $(i_k/o_k) \cdot (i_{k+1}/o_{k+1})\dots(i_j/o_j) \cdot \beta$; i.e., $(s_{k-1}, s_k, i_{k-1}/o_{k-1})$ cannot be verified by $(i_k/o_k) \cdot (i_{k+1}/o_{k+1})\dots(i_j/o_j) \cdot \beta$. This contradicts with the assumption

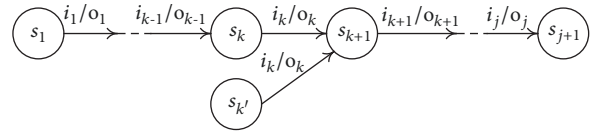

Figure 3: A fragment of an FSM.

that $(s_1, s_2, i_1/o_1)(s_2, s_3, i_2/o_2)\dots(s_j, s_{j+1}, i_j/o_j)$ is an IDOS. Accordingly, it is a necessary condition for a transition sequence $(s_1, s_2, i_1/o_1)(s_2, s_3, i_2/o_2)\dots(s_j, s_{j+1}, i_j/o_j)$ to be an IDOS that every transition $(s_k, s_{k+1}, i_k/o_k)$ $(2 \leqslant k \leqslant j)$ is invertible. □

Definition 4 (set of IDOSs). Given a set T in which every sequence is an IDOS of an FSM M, if every transition of M is included in one and only one IDOS of T then T is a set of IDOSs of M.

Of all the IDOSs from a state, a maximum-length IDOS is the one contains no fewer transitions than any other ones. There is no doubt that the longer IDOSs are, the more overlapping can be achieved, and the shorter test sequences will be obtained. For maximum overlapping, this paper is only interested in maximum-length IDOSs.

The set $\{s_1(a/0)s_2(a/1)s_2(b/1)s_3(a/1)s_1\}$ is a set of IDOSs of M_1 in Figure 2. There is only one IDOS in the set since the only IDOS $s_1(a/0)s_2(a/1)s_2(b/1)s_3(a/1)s_1$ already covers all the transitions of M_1. Obviously, $s_1(a/0)s_2(a/1)s_2(b/1)s_3(a/1)s_1$ is also a maximum-length IDOS starting from s_1.

3.1.2. Multiple UIO Sequences. In the improved method, for any maximum-length IDOS $(s_1, s_2, i_1/o_1)(s_2, s_3, i_2/o_2)\dots(s_j, s_{j+1}, i_j/o_j)$, the associated test subsequence is expressed as $(s_1, s_2, i_1/o_1)(s_2, s_3, i_2/o_2)\dots(s_j, s_{j+1}, i_j/o_j) \cdot \beta$ where β is a UIO sequence of s_{j+1}. A minimum-length rural postman tour covering all the test subsequences is subsequently constructed by a rural symmetric augmentation and a test sequence is obtained from the tour. Accordingly, transitions for concatenation of test subsequences in the test sequence are derived from the transition replications during the rural symmetric augmentation. It is noted that different choice of UIO sequences may result in different rural symmetric augmentations; i.e., a minimum number of transition replications can be achieved by a sensible choice of UIO sequences during the rural symmetric augmentation. The minimum number of transition replications during the rural symmetric augmentation indicates the minimum number of transitions for concatenation of test subsequences in the test sequence, leading to a reduced test sequence. In other words, the result of using single UIO sequences can only in best-case scenarios obtain the same length of minimum-length rural postman tours as that of using multiple UIO sequences.

3.2. Design of the Method. It is known from Theorem 3 that noninvertible transitions restrict the generation of IDOSs.

From this point of view, FSMs can be partitioned into two subsets. One is FSMs with only invertible transitions and the other is FSMs with noninvertible transitions. For FSMs with only invertible transitions, if the FSMs are symmetric test sequences can be obtained directly. Otherwise, the FSMs should be augmented firstly. So FSMs with only invertible transitions can also be partitioned into two subsets. One is symmetric FSMs with only invertible transitions and the other is asymmetric FSMs with only invertible transitions. Generally, FSMs are classified into three categories: symmetric FSMs with only invertible transitions, asymmetric FSMs with only invertible transitions and FSMs with noninvertible transitions. The improved method is described in two steps for each type of FSMs and the detail varies for different types.

Step 1. Construct the set of maximum-length IDOSs.

Step 2. With the consideration of multiple UIO sequences, generate a minimum-length rural postman tour covering the set of maximum-length IDOSs and extract a test sequence from the tour.

3.2.1. Symmetric FSMs with Only Invertible Transitions

Step 1 (maximum-length IDOSs generation). There is an Euler tour in a directed graph if and only if the directed graph is strongly connected and symmetric. Under the assumption that all the FSMs are strongly connected, there must be Euler tours for symmetric FSMs with only invertible transitions. An Euler tour starting from and ending at the initial state is definitely an IDOS since there are only invertible transitions; furthermore, it is a maximum-length IDOS from the initial state since there is no other one containing more transitions. In other words, an Euler tour of a symmetric FSM with only invertible transitions is the only sequence in the set of maximum-length IDOSs.

Note that an Euler tour starting from and ending at the initial state may not be unique; i.e., there may be nonunique sets of maximum-length IDOSs. Nonetheless, all the sets of maximum-length IDOSs hold the following properties.

(i) There is only one maximum-length IDOS covering all the transitions of the FSM in every set of maximum-length IDOSs.

(ii) The start state of the maximum-length IDOS is the initial state of the FSM in every set of maximum-length IDOSs.

(iii) The end state of the maximum-length IDOS is the initial state of the FSM in every set of maximum-length IDOSs.

Step 2 (test sequence generation). For symmetric FSMs with only invertible transitions, test sequences are denoted by $(s_1, s_2, i_1/o_1)(s_2, s_3, i_2/o_2)\dots(s_j, s_1, i_j/o_j) \cdot \beta$ where $(s_1, s_2, i_1/o_1)(s_2, s_3, i_2/o_2)\dots(s_j, s_1, i_j/o_j)$ is an Euler tour staring from and ending at s_1 and β is a minimum-length UIO sequence of s_1. It can be inferred that test sequences from different sets of maximum-length IDOSs are always of the same length when there is more than one set of maximum-length IDOSs. As a result, a randomly generated set of maximum-length IDOSs will do for test sequence generation of symmetric FSMs with only invertible transitions.

M_1 in Figure 2 is a symmetric FSM with only invertible transitions and $s_1(a/0)s_2(a/1)s_2(b/1)s_3(a/1)s_1$ is an Euler tour starting from and ending at s_1. It is known from Step 1 that $\{s_1(a/0)s_2(a/1)s_2(b/1)s_3(a/1)s_1\}$ is a set of maximum-length IDOSs of M_1. A test sequence $s_1(a/0)s_2(a/1)s_2(b/1)s_3(a/1)s_1(a/0)s_2$ is resulted from Step 2.

3.2.2. Asymmetric FSMs with Only Invertible Transitions

Step 1 (maximum-length IDOSs generation). An FSM is asymmetric which refers to the fact that there are nodes whose out-degree does not equal in-degree. It is known that $\sum_{s\in S} d_{out}(s) = \sum_{s\in S} d_{in}(s) = |E|$ where $d_{out}(s)$ and $d_{in}(s)$ denote the out-degree and in-degree of a node, respectively [13]. The notation $|E|$ refers to the number of edges in a directed graph. It is concluded from $\sum_{s\in S} d_{out}(s) = \sum_{s\in S} d_{in}(s)$ that $\sum_{s\in S^-} d_{out>in}(s) = \sum_{s'\in S^+} d_{in>out}(s')$ where S^- and S^+ are the sets of nodes whose out-degree outnumbers in-degree and in-degree outnumbers out-degree, respectively. Correspondingly, $d_{out>in}(s)$ and $d_{in>out}(s')$ represent the amount of out-degree over in-degree for a node in S^- and the amount of in-degree over out-degree for a node in S^+, respectively. The relation $\sum_{s\in S} d_{out}(s) = \sum_{s\in S} d_{in}(s) = |E|$ implies that if a path passes through as many edges as possible exactly once then the path consumes as many out-degree and in-degree as possible. Every path through a node takes up one incoming edge as well as one outgoing edge, so if the out-degree outnumbers in-degree at a node, its outgoing edges cannot be traversed completely by paths passing through the node; i.e., some of its outgoing edges must instead be traversed by paths starting from the node. Thus, it is advisable to start the paths from nodes whose out-degree outnumbers in-degree if all the edges should be traversed exactly once with as few paths as possible. Moreover, it can be proved by contradiction that the paths from nodes whose out-degree outnumbers in-degree must end at nodes whose in-degree outnumbers out-degree otherwise the paths will continue to extend.

According to the above analysis, maximum-length IDOSs generation for asymmetric FSMs with only invertible transitions is performed as follows: start from a node $s \in S^-$ and go down along an outgoing edge until a node $s' \in S^+$ is reached and all the outgoing edges of s' have been traversed. Thus a maximum-length IDOS from s to s' is obtained. Add the maximum-length IDOS to the set of maximum-length IDOSs and delete the corresponding edges in the directed graph. Repeat the above process until all the nodes of the directed graph are isolated which implies that the set of maximum-length IDOSs is obtained.

Similarly, there may be nonunique sets of maximum-length IDOSs for asymmetric FSMs with only invertible transitions and all the sets of maximum-length IDOSs hold the following properties:

(i) For any set of maximum-length IDOSs, S^- is the set of start states for maximum-length IDOSs.

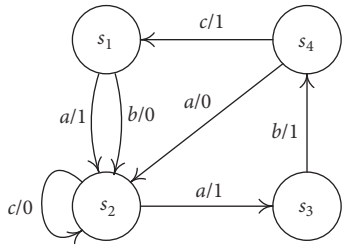

Figure 4: FSM M_2.

(ii) For any set of maximum-length IDOSs, S^+ is the set of end states for maximum-length IDOSs.

(iii) For any node $s \in S^-$, the number of maximum-length IDOSs starting from s is $d_{out>in}(s)$ in every set of maximum-length IDOSs.

(iv) For any node $s' \in S^+$, the number of maximum-length IDOSs ending at s' is $d_{in>out}(s')$ in every set of maximum-length IDOSs.

(v) The total number of maximum-length IDOSs is $\sum_{s\in S^-} d_{out>in}(s)$ which equals $\sum_{s'\in S^+} d_{in>out}(s')$ in each set of maximum-length IDOSs.

Step 2 (test sequence generation). With the set of maximum-length IDOSs, the same method as that of Bo Yang et al. is used to construct test sequences and the detail of the method is described in Algorithm 1. The core of the method is the rural symmetric augmentation over an FSM augmented by maximum-length IDOSs and the multiple UIO sequences for the end states of maximum-length IDOSs. The core of the symmetric augmentation is the states involved. According to the above properties, for any set of maximum-length IDOSs, there is no difference about the states involved in the symmetric augmentation such that a random set of maximum-length IDOSs will do for asymmetric FSMs with only invertible transitions when there is more than one set of maximum-length IDOSs.

M_2 in Figure 4 is an asymmetric FSM with only invertible transitions and its UIO sequences are shown in Table 3. The nonunique sets of maximum-length IDOSs for M_2 are listed as follows:

$\{s_1(a/1)s_2(c/0)s_2(a/1)s_3(b/1)s_4(c/1)s_1(b/0)s_2, s_4(a/0)s_2\}$,
$\{s_1(a/1)s_2(c/0)s_2(a/1)s_3(b/1)s_4(a/0)s_2, s_4(c/1)s_1(b/0)s_2\}$,
$\{s_1(b/0)s_2(c/0)s_2(a/1)s_3(b/1)s_4(c/1)s_1(a/1)s_2, s_4(a/0)s_2\}$,
$\{s_1(b/0)s_2(c/0)s_2(a/1)s_3(b/1)s_4(a/0)s_2, s_4(c/1)s_1(a/1)s_2\}$.

Clearly, all the sets of maximum-length IDOSs satisfy the properties in this section. The augmentation of M_2 using a random set of maximum-length IDOSs $\{s_1(a/1)s_2(c/0)s_2(a/1)s_3(b/1)s_4(c/1)s_1(b/0)s_2, s_4(a/0)s_2\}$ is shown in Figure 5 and the associated test sequence of M_2 is $s_1(a/1)s_2(c/0)s_2(a/1)s_3(b/1)s_4(c/1)s_1(b/0)s_2(a/1)s_3(b/1)s_4(a/0)s_2(c/0)s_2$.

3.2.3. FSMs with Noninvertible Transitions

Step 1 (maximum-length IDOSs generation). To address an FSM with noninvertible transitions in a similar way to the first two types of FSMs, noninvertible transitions are removed from the FSM and saved to a set. The remainder excluding noninvertible transitions is either an FSM or more than one connected component with only invertible transitions. And thus maximum-length IDOSs of the remainder excluding noninvertible transitions are generated in the same way as symmetric or asymmetric FSMs with only invertible transitions. Naturally, it comes to the same conclusion as the first two types of FSMs that a random set of maximum-length IDOSs will do when there are nonunique sets of maximum-length IDOSs. Next, a union of maximum-length IDOSs of the remainder excluding noninvertible transitions and all the noninvertible transitions is derived; moreover, whenever the start state of a maximum-length IDOS is the end state of a noninvertible transition, the maximum-length IDOS is concatenated with the noninvertible transition. If a maximum-length IDOS is an Euler tour then a node which is the end state of a noninvertible transition is preferred to be the start state of the tour. The set after all possible concatenations is a set of maximum-length IDOSs of the FSM with noninvertible transitions.

Table 3: UIO sequence of M_2.

No.	States	UIO(s)
Ex1	s_1	$b/0$ $(a/1)(c/0)$ $(a/1)(a/1)$
Ex2	s_2	$(a/1)(b/1)$ $c/0$
Ex3	s_3	$b/1$
Ex4	s_4	$a/0$ $c/1$

Given a union of maximum-length IDOSs of the remainder excluding noninvertible transitions as well as all the noninvertible transitions, there may be nonunique sets of maximum-length IDOSs for the FSM with noninvertible transitions because of different concatenation. The properties of nonunique sets of maximum-length IDOSs for FSMs with noninvertible transitions are described as follows:

(i) In any set of maximum-length IDOSs, for any state s which is the start state of a maximum-length IDOS, if there are j maximum-length IDOSs starting from s then there must be j maximum-length IDOSs starting from s in every other set of maximum-length IDOSs.

(ii) In any set of maximum-length IDOSs, for any state s which is the end state of a maximum-length IDOS, if there are k maximum-length IDOSs ending at s then there must be k maximum-length IDOSs ending at s in every other set of maximum-length IDOSs.

Step 2 (test sequence generation). For FSMs with noninvertible transitions, test sequence generation is in the same way as asymmetric FSMs with only invertible transitions, i.e., by means of rural symmetric augmentation over an FSM augmented by maximum-length IDOSs and the multiple UIO sequences for the end states of maximum-length IDOSs.

Require:
FSM M with n states $\{s_1, s_2, \ldots, s_n\}$ in which s_1 is the initial state;
Set of non-invertible transitions $P = \phi$;
Set of maximum-length IDOSs $Q = \phi$;
Set of end states for non-invertible transitions $S' = \phi$;
UIO sequences of M;
A minimum-length UIO sequence γ of s_1;
Ensure:
Reduced test sequence β of M;
(1) **if** M is an *FSM* with m non-invertible transitions **then**
(2) **for** i=1 to m **do**
(3) $P = P \cup t_i$ where t_i denotes a non-invertible transition;
(4) $M = M \setminus t_i$;
(5) $S' = S' \cup e_i$ where e_i denotes the end state of t_i;
(6) **end for**
(7) **for** each state whose out-degree > in-degree in M **do**
(8) $Q=Q \cup \alpha$ where α is a maximum-length IDOS of j transitions generated from the state;
(9) $M=M \setminus t_i$ $(1 \leqslant i \leqslant j)$;
(10) **end for**
(11) **for** each connected component with Euler tours **do**
(12) **if** States in S' can be found in an Euler tour t with k transitions **then**
(13) Choose a state in S' as the initial state of t;
(14) **end if**
(15) $Q=Q \cup t$;
(16) $M=M \setminus t_i$ $(1 \leqslant i \leqslant k)$;
(17) **end for**
(18) $Q=Q \cup P$;
(19) Concatenate non-invertible transitions and maximum-length IDOSs as long as the former's end state is the latter's start state;
(20) $M = M \cup V^*$ where V^* is a new state set;
(21) $M = M \cup T$ where T is a new transition set;
(22) $M = M \cup U$ where U is a new transition set;
(23) Construct a minimum-length rural postman tour over Q in the augmented M and extract a test sequence starting from s_1;
(24) Remove the transition sequence follows the UIO sequence of the maximum-length IDOS which is verified last in the minimum-length tour;
(25) **else**
(26) **if** M is a symmetric FSM with only invertible transitions **then**
(27) $Q = Q \cup t$ where t is an Euler tour starting from and ending at s_1;
(28) $\beta = t \cdot \gamma$;
(29) **else**
(30) execute lines (7) through (9);
(31) Execute lines (20) through (24);
(32) **end if**
(33) **end if**

Algorithm 1: Improved method of test sequence reduction.

The core of rural symmetric augmentation is still the states involved. It is known from the above properties that for any set of maximum-length IDOSs the states involved in the rural symmetric augmentation are the same such that a random concatenation will do.

Considering M_3 in Figure 6 with UIO sequences in Table 4, the following nonunique union of maximum-length IDOSs of the remainder excluding noninvertible transitions and noninvertible transitions confirm that a random union will do.

$\{s_1(a/1)s_2(c/0)s_2(a/1)s_3(b/1)s_4(c/1)s_1(b/0)s_2, s_4(a/0)s_2, s_1(d/0)s_4, s_3(d/0)s_4\}$,

$\{s_1(a/1)s_2(c/0)s_2(a/1)s_3(b/1)s_4(a/0)s_2, s_4(c/1)s_1(b/0)s_2, s_1(d/0)s_4, s_3(d/0)s_4\}$,

$\{s_1(b/0)s_2(c/0)s_2(a/1)s_3(b/1)s_4(c/1)s_1(a/1)s_2, s_4(a/0)s_2, s_1(d/0)s_4, s_3(d/0)s_4\}$,

$\{s_1(b/0)s_2(c/0)s_2(a/1)s_3(b/1)s_4(a/0)s_2, s_4(c/1)s_1(a/1)s_2, s_1(d/0)s_4, s_3(d/0)s_4\}$.

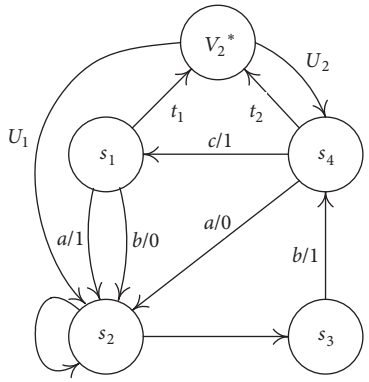

Figure 5: Augmentation of M_2.

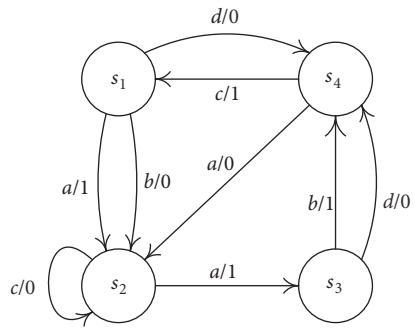

Figure 6: FSM M_3.

Table 4: UIO sequence of M_3.

No.	States	UIO(s)
Ex1	s_1	$b/0$ $(a/1)(c/0)$ $(a/1)(a/1)$
Ex2	s_2	$(a/1)(b/1)$ $c/0$
Ex3	s_3	$b/1$
Ex4	s_4	$a/0$ $c/1$

Take a random union of maximum-length IDOSs of the remainder excluding noninvertible transitions and noninvertible transitions $\{s_1(a/1)s_2(c/0)s_2(a/1)s_3(b/1)s_4(c/1)s_1(b/0)s_2$, $s_4(a/0)s_2$, $s_1(d/0)s_4$, $s_3(d/0)s_4\}$; nonunique sets of maximum-length IDOSs from different concatenations $\{s_1(a/1)s_2(c/0)s_2(a/1)s_3(b/1)s_4(c/1)s_1(b/0)s_2$, $s_3(d/0)s_4(a/0)s_2, s_1(d/0)s_4\}$ and $\{s_1(a/1)s_2(c/0)s_2(a/1)s_3(b/1)s_4(c/1)s_1(b/0)s_2$, $s_1(d/0)s_4(a/0)s_2$, $s_3(d/0)s_4\}$ confirm that a random concatenation will do. The augmentation of M_3 using a randomly concatenated set of maximum-length IDOSs $\{s_1(a/1)s_2(c/0)s_2(a/1)s_3(b/1)s_4(c/1)s_1(b/0)s_2$, $s_3(d/0)s_4(a/0)s_2$, $s_1(d/0)s_4\}$ is shown in Figure 7 and the resulting test sequence is $s_1(d/0)s_4(c/1)s_1(a/1)s_2(c/0)s_2(a/1)s_3(b/1)s_4(c/1)$ $s_1(b/0)s_2(c/0)s_2(a/1)s_3(d/0)s_4(a/0)s_2(c/0)s_2$.

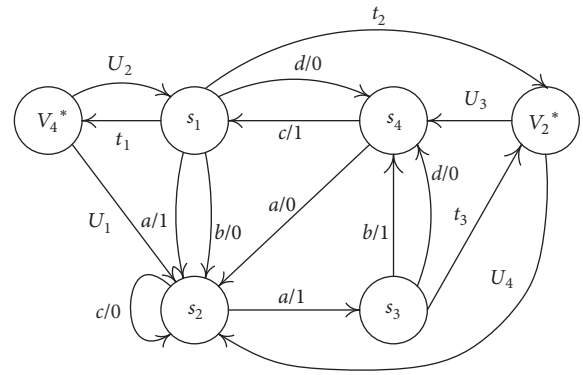

Figure 7: Augmentation of M_3.

Algorithm 1 describes the detail of the improved method for all types of FSMs. Note that self-loops are always given priority to traverse in the process of maximum-length IDOSs generation. When creating the new state set V^* in M, for the end state of every maximum-length IDOS, there is a state in V^*. When creating the new transition set T in M, for every maximum-length IDOS, there is a transition from the start state of the maximum-length IDOS to state $v_i{}^*$ labeled with the corresponding label of the maximum-length IDOS. When creating a new transition set U in M, for every UIO sequence of the end state of every maximum-length IDOS, there is a transition from $v_i{}^*$ to the end state of every UIO sequence labeled with the corresponding UIO sequence.

Theorem 5. *Given an FSM M, suppose that Q is a set of maximum-length IDOSs and β is a sequence over Q generated from Algorithm 1, then β is a reduced test sequence of M.*

Proof. The soundness of β is first proved; i.e., β is a test sequence of M. Then the effectiveness of β is assessed in terms of test generation and test execution cost, respectively. From a general standpoint, the notion of cost in the context of testing is complex and can be related to many factors. In our context, the effort required for generating test sequences is measured in terms of test sequence computational complexity. Test sequence execution cost is measured by the length of test sequences. Although these are clearly approximation methods, for practical reasons such methods have been commonly used in a number of testing studies [14–16]. □

(1) Soundness Analysis

(i) Check whether every transition defined in M is verified in the implementation

All the maximum-length IDOSs in Q are included in β since β is a sequence over Q. Algorithm 1 indicates that every maximum-length IDOS in β is followed by a UIO sequence that verifies the last transition of the sequence. According to Definition 2, i.e., for any maximum-length IDOS $(s_1, s_2, i_1/o_1)(s_2, s_3, i_2/o_2)\ldots(s_j, s_{j+1}, i_j/o_j)$, if $(s_j, s_{j+1}, i_j/o_j)$ is verified then every transition $(s_k, s_{k+1}, i_k/o_k)$ $(1 \leqslant k \leqslant j-1)$ is verified, it is inferred that transitions of all the

maximum-length IDOSs in Q are verified in β. According to Definition 4, i.e., every transition in M is included in one and only one IDOS of Q, it is concluded that every transition defined in M is verified in the implementation by β.

(ii) Check whether every state in M is defined in the implementation

M is supposed to be strongly connected such that for any state s of M there is at least one transition ending at s. It is proved that every transition defined in M is verified in β by identifying the end state of the transition; i.e., every state of M is checked in the implementation by β.

(2) Effectiveness Analysis

(i) Analysis of Computational Complexity. The test sequence β has the same computational complexity as those of Aho et al. and Hierons since all the three test sequence generation methods are based on the max flow/min cost problem and the networks used in every method are of the same order.

(ii) Analysis of Length Reduction. As mentioned above, transitions in a test sequence are of three kinds and the cost of a test sequence comes from the UIO sequences which have been appended for the purpose of transition verification and the transitions which have been appended for the purpose of concatenation of test subsequences to generate a minimum-length rural postman tour.

For symmetric FSMs with only invertible transitions, β is in the form of $(s_1, s_2, i_1/o_1)(s_2, s_3, i_2/o_2)\ldots(s_j, s_1, i_j/o_j)\cdot\gamma$ where $(s_1, s_2, i_1/o_1)(s_2, s_3, i_2/o_2)\ldots(s_j, s_1, i_j/o_j)$ is an Euler tour staring from and ending at s_1 and γ is a minimum-length UIO sequence of s_1; i.e., the cost of β only comes from one appended minimum-length UIO sequence of s_1.

For asymmetric FSMs with only invertible transitions, all the transitions are divided into a least number of maximum-length IDOSs; i.e., all the transitions are verified by a least number of appended UIO sequences such that the cost of β from the appended UIO sequences is minimal. For any maximum-length IDOS $(s_1, s_2, i_1/o_1)(s_2, s_3, i_2/o_2)\ldots(s_j, s_{j+1}, i_j/o_j)$, there is a test subsequence $(s_1, s_2, i_1/o_1)(s_2, s_3, i_2/o_2)\ldots(s_j, s_{j+1}, i_j/o_j)\cdot\gamma$ where γ is a UIO sequence of s_{j+1}. β is obtained from a minimum-length rural postman tour covering all the test subsequences. The minimum-length rural postman tour is generated by the rural symmetric augmentation and thus the cost of β for concatenation comes from the replicated transitions during the rural symmetric augmentation. Benefiting from multiple UIO sequences, a minimum number of transition replications is reached by a sensible choice of UIO sequences during the rural symmetric augmentation; i.e., the cost of β from the transitions for concatenation is minimal.

For FSMs with noninvertible transitions, transitions excluding noninvertible ones are divided into a least number of IDOSs. According to Theorem 3, the acquired IDOSs concatenate with noninvertible transitions as much as possible such that a set of maximum-length IDOSs with a least number of maximum-length IDOSs is obtained; i.e., all the transitions of an FSM with noninvertible transitions are verified by a least number of appended UIO sequences such that the cost of β from the appended UIO sequences is minimal. Similarly, a minimum number of transition replications during the rural symmetric augmentation is reached by a sensible choice of UIO sequences; i.e., the cost of β from the transitions for concatenation is minimal.

In short, for all types of FSMs, the execution cost is reduced effectively without increasing the generation cost such that β is a reduced test sequence of M.

4. Case Study

We experiment with the aforementioned M_1, M_2, and M_3 which are random generation of different types of FSMs. M_4 and M_5 are also randomly generated FSMs; moreover, they are example FSMs used by Bo Yang et al. and Hierons, respectively. While the experimental results shed some light on how the improved method behaves with randomly generated FSMs, they bring no insight on the test sequence reduction of FSMs that are produced by software designers. For this reason, experiments on FSMs modeling INRES protocol [17], GUI for password modification in a property management system [18], page function of Gmail system [19] and the connection and release process of MAC layer in Zigbee protocol are carried out, and the FSM modeling page function of Gmail system is adjusted to satisfy the strong connectivity and UIO availability assumptions.

The FSMs used in the experiment are admittedly small. However, it is important to note that FSMs are mostly used to model the behavior of complex classes or class clusters, particularly complex control classes in protocols or reactive systems. They are rarely used to model entire systems which will result in large and unmanageable models for software engineers and testers. Completeness degree is a general factor to reveal the complexity of both large and small FSMs such that test sequence reduction of large FSMs can to some extent be learned through relatively small FSMs with the same distribution of completeness degree. Given an FSM with x inputs, y transitions and n states, completeness degree of M is denoted by $y/(x \times n)$; moreover, the higher the completeness degree is, the more complex the FSM is. In this section, completeness degrees of FSMs range from 0.24 to 1.

The associated data of the experiment is illustrated in Table 5, $|Input|$, $|State|$ and $|Transition|$ denote the number of inputs, states and transitions of every FSM. Completeness degree of every FSM is calculated and listed. $T_{FOTS+MUIO}$, T_{inv}, and $T_{inv+MUIO}$ are test sequences resulting from Bo Yang et al., Hierons, and the improved method, respectively. $|T_{FOTS+MUIO}|$, $|T_{inv}|$, and $|T_{inv+MUIO}|$ represent the length of the corresponding test sequences.

For symmetric FSMs with only invertible transitions, T_{inv} and $T_{inv+MUIO}$ are both Euler tours starting from the initial state followed by a minimum-length UIO sequence of the initial state. Thus, $|T_{inv}|$ and $|T_{inv+MUIO}|$ are always the same for symmetric FSMs with only invertible transitions. When the Euler tour conforms to the definition of fully overlapping transition sequences (FOTSs) [11], $|T_{FOTS+MUIO}|$ is the same as $|T_{inv}|$ and $|T_{inv+MUIO}|$; otherwise $|T_{FOTS+MUIO}|$ tends to be longer because of more appended UIO sequences as well as the possible extra transitions for concatenation.

Table 5: Experimental objects and associated data.

FSMs	Information of FSMs				Length of Test Sequences		
	$\lvert Input\rvert$	$\lvert State\rvert$	$\lvert Transition\rvert$	Completeness Degree	$\lvert T_{FOTS+MUIO}\rvert$	$\lvert T_{inv}\rvert$	$\lvert T_{inv+MUIO}\rvert$
M_1	2	3	4	0.67	8	5	5
M_2	3	4	7	0.58	10	12	10
M_3	4	4	9	0.56	15	15	13
M_4	3	5	11	0.73	17	17	16
M_5	2	5	10	1	33	21	20
INRES	5	4	16	0.8	13	12	12
GUI	7	8	25	0.45	74	31	31
Gmail	9	7	15	0.24	43	27	27
Zigbee	7	4	7	0.25	15	14	14

Table 6: Single sample K-S check.

	$\lvert T_{FOTS+MUIO}\rvert$	$\lvert T_{inv}\rvert$	$\lvert T_{inv+MUIO}\rvert$
N	9	9	9
Mean Value	25.333	17.111	16.444
Standard Deviation	21.535	8.054	8.263
Kolmogorov – Smirnov Z	.952	.517	.564
Asympotic Significance (2 – tailed)	.325	.952	.908

Table 7: Paired samples statistics.

Pair	Mean	N	Standard Deviation	Standard Error Mean
$\lvert T_{FOTS+MUIO}\rvert$	25.333	9	21.535	7.178
$\lvert T_{inv+MUIO}\rvert$	16.444	9	8.263	2.754
$\lvert T_{inv}\rvert$	17.111	9	8.054	2.685
$\lvert T_{inv+MUIO}\rvert$	16.444	9	8.263	2.754

For asymmetric FSMs with only invertible transitions, $\lvert T_{FOTS+MUIO}\rvert$ and $\lvert T_{inv+MUIO}\rvert$ are the same if the maximum-length IDOSs comply with the definition of FOTS. Otherwise, $\lvert T_{FOTS+MUIO}\rvert$ tends to be longer because of more appended UIO sequences as well as the possible extra transitions for concatenation. T_{inv} is an Euler tour from a rural symmetric augmentation of an asymmetric FSM with only invertible transitions followed by a minimum-length UIO sequence of the initial state. The number of appended transitions in the rural symmetric augmentation is a deciding factor of $\lvert T_{inv}\rvert$.

For FSMs with noninvertible transitions, $\lvert T_{FOTS+MUIO}\rvert$ is longer than $\lvert T_{inv+MUIO}\rvert$ since every noninvertible transition is verified individually by an appended UIO sequence. $\lvert T_{inv}\rvert$ is the same as $\lvert T_{inv+MUIO}\rvert$ at best otherwise $\lvert T_{inv}\rvert$ tends to be longer than $\lvert T_{inv+MUIO}\rvert$.

As shown in Table 5, test sequences generated from different test methods conform to the above theoretical analysis. In this section, *two-sample t-test* is performed to compare test methods in terms of the length of the associated test sequences. *Single sample K-S check* in SPSS is used to verify the normality of the data studied since *t-test* is a parametric test and requires data to be normally distributed. Normality results are reported whenever samples deviate significantly from the normal distribution and the result of single sample *K-S* check in Table 6 shows that all the data in this experiment follow normal distribution. The paired samples statistics in Table 7 indicates that the average length of test sequences from the improved method is superior to those of the other two methods.

Vargha-Delaney effect size measure($\widehat{A}_{12}$) is also calculated to get more credible conclusions. When comparing two methods, $\widehat{A}_{12}$ measures the probability that one method would perform better than the other method. A value of 0.5 would mean that the two methods have equal probability of performing better than the other. The Vargha-Delaney effect size measure ($\widehat{A}_{12}$) from comparing $\lvert T_{inv+MUIO}\rvert$ to $\lvert T_{FOTS+MUIO}\rvert$ and $\lvert T_{inv}\rvert$ is shown in Table 8. The results show that $\lvert T_{inv+MUIO}\rvert$ is statistically 60.5% and 54.3% of the time significantly shorter than $\lvert T_{FOTS+MUIO}\rvert$ and $\lvert T_{inv}\rvert$, respectively.

5. Related Work

There are many works on IoT testing. Xiaoping Che et al. presented a logic-based approach to test the conformance and performance of XMPP protocol which is gaining momentum in IoT through real execution traces and formally specified

Table 8: Statistical results from $(\widehat{A}_{12})$.

	$\lvert T_{inv+MUIO}\rvert$ *versus*	$(\widehat{A}_{12})$
Ex1	$\lvert T_{FOTS+MUIO}\rvert$	0.395
Ex2	$\lvert T_{inv}\rvert$	0.457

properties [20]. Dimitrios Serpanos et al. introduced testing for security for IoT systems and especially fuzz testing, which is a successful technique to identify vulnerabilities in systems and network protocols [21]. Martin Tappler et al. presented a model-based approach to test IoT communication via active automata learning [22]. Combining Model-Based Testing (MBT) and a service-oriented solution, Abbas Ahmad et al. presented Model-Based Testing As A Service (MBTAAS) for testing data and IoT platforms [23]. Hiun Kim et al. introduced IoT testing as a Service-IoT-TaaS which is composed of remote distributed interoperability testing, scalable automated conformance testing, and semantics validation testing components adequate for testing IoT devices [24]. John Esquiagola et al. used the current version of their IoT platform to perform performance testing [25]. Daniel Kuemper et al. described how concepts for semantically described web services can be transferred into the IoT domain [26]. Philipp Rosenkranz et al. propose a testing framework which supports continuous integration techniques and allows for the integration of project contributors to volunteer hardware and software resources to the test system [27]. A connective and semantic similarity clustering algorithm (CSSCA) and a hierarchical combinatorial test model based on FSM are proposed by Kai Cui et al. [28].

Test sequence generation and reduction has long been an active research topic. Porto et al. used identification sets which are subsets of a characterizing set to identify states and obtained reduced test sequences [29]. Locating sequences are used to make sure that every element of a characterizing set is applied to the same state. Jourdan et al. generated shorter test sequences by means of reducing the number of locating sequences [30]. Baumgartner et al. proposed a mixed integer nonlinear programming (MINLP) model to formalize how the total cost of testing depends on the sequence and the parameters of the elementary test steps [31]. To provide an efficient formalization of the scheduling problem and avoid difficulties due to the evaluation of an objective function during the relaxation of the integer variables, the MINLP was formulated as a process network synthesis problem. Hierons et al. affirmed the importance of invertibility in test sequence reduction and considered three optimisation problems associated with invertible sequences [32]. Petrenko et al. addressed the problem of extending the checking experiment theory to cover a class of FSMs with symbolic extensions [33]. They also reported the results that further lift the theory of checking experiments for Mealy machines with symbolic inputs and symbolic outputs. Hierons et al. described an efficient parallel algorithm that uses many-core GPUs for automatically deriving UIOs from Finite State Machines [34]. The proposed algorithm uses the global scope of the GPU's global memory through coalesced memory access and minimizes the transfer between CPU and GPU memory. Song et al. introduced a practical conformance testing tool that generates high-coverage test input packets using a conformance test suite and symbolic execution. This approach can be viewed as the combination of conformance testing and symbolic execution [35]. Bokil et al. presented an automated black box test suite generation technique for reactive systems [36]. The technique is based on dynamic mining of specifications in form of an FSM from initial runs. The set of test cases thus produced contain several redundant test cases, many of which are eliminated by a simple greedy test suite reduction algorithm to give the final test suite.

6. Conclusions

Taking Zigbee protocol as an example, this paper introduces how FSM-based conformance testing works in wireless protocol conformance testing of IoT. An improved method in which both overlapping by invertibility and multiple UIO sequences are considered is proposed to achieve test sequence reduction for wireless protocol conformance testing of IoT. Based on invertibility, transitions of all types of FSMs are verified with as few appended UIO sequences as possible. Multiple UIO sequences contribute to generate a shorter test sequence by means of reducing transitions for concatenation of test subsequences in the test sequence. Moreover, test sequences can be further reduced by removing the transition sequence which follows the UIO sequence of the maximum-length IDOS that is verified last in the tour. Theory and experiment indicate that the execution cost is reduced effectively by the improved method under the premise of not increasing the generation cost. The improved method is also applicable to traditional protocol conformance testing as well as reactive systems. Numerous experimental data and practical examples about wireless protocols of IoT will be gathered in the future work to analyze the effectiveness of the method.

Data Availability

The data used to support the findings of this study are included within the article.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

Acknowledgments

This work is supported by National Natural Science Foundation of China (Grant no. 61572306 and Grant no. 61502294), the IIOT Innovation and Development Special Foundation of Shanghai (Grant no. 2017-GYHLW-01037), and the CERNET Innovation Project (NGII20170513 and NGII20170206).

References

[1] L. Qi, X. Zhang, W. Dou, and Q. Ni, "A distributed locality-sensitive hashing-based approach for cloud service recommendation from multi-source data," *IEEE Journal on Selected Areas in Communications*, vol. 35, no. 11, pp. 2616–2624, 2017.

[2] L. Qi, X. Xu, X. Zhang et al., "Structural balance theory-based e-commerce recommendation over big rating data," *IEEE Transactions on Big Data*, vol. 4, no. 3, pp. 301–312, 2018.

[3] K. Zaniewski and W. Pedrycz, "A hybrid optimization approach to conformance testing of finite automata," *Applied Soft Computing*, vol. 23, pp. 91–103, 2014.

[4] S. J. S., K. H., and P. S., "Enhancing conformance testing using symbolic execution for network protocols," *IEEE Transactions on Reliability*, vol. 64, no. 3, pp. 1024–1037, 2015.

[5] P. Liu, H.-K. Miao, H.-W. Zeng, and Y. Liu, "FSM-based testing: Theory, method and evaluation," *Jisuanji Xuebao/Chinese Journal of Computers*, vol. 34, no. 6, pp. 965–984, 2011.

[6] R. Dorofeeva, K. El-Fakih, S. Maag, A. R. Cavalli, and N. Yevtushenko, "Experimental evaluation of FSM-based testing methods," in *Proceedings of the 3rd IEEE International Conference on Software Engineering and Formal Methods (SEFM '05)*, pp. 23–32, IEEE, Koblenz, Germany, September 2005.

[7] X. Zhang, M. Yang, J. Zhang, H. Shi, and W. Zhang, "A study on the extended unique input/output sequence," *Information Sciences*, vol. 203, pp. 44–58, 2012.

[8] I. Ahmad, F. M. Ali, and A. S. Das, "Synthesis of finite state machines for improved state verification," *Computers & Electrical Engineering*, vol. 32, no. 5, pp. 349–363, 2006.

[9] J. Shujuan, J. Xiaolin, W. Xingya, L. Haiyang, Z. Yamei, and L. Yingqi, "Measuring the importance of classes using uio sequence," *Chinese Journal of Electronics*, vol. 43, no. 10, pp. 2062–2068, 2015 (Chinese).

[10] É. F. D. Souza, V. A. D. Santiago Júnior, and N. L. Vijaykumar, "H-Switch Cover: a new test criterion to generate test case from finite state machines," *Software Quality Journal*, vol. 25, no. 2, pp. 373–405, 2017.

[11] B. Yang and H. Ural, "Protocol conformance test generation using multiple uio sequences with overlapping," *ACM SIGCOMM Computer Communication Review*, vol. 20, no. 4, pp. 118–125, 1990.

[12] R. M. Hierons, "Extending test sequence overlap by invertibility," *The Computer Journal*, vol. 39, no. 4, pp. 325–330, 1996.

[13] J. Bang-Jensen and G. Z. Gutin, *Digraphs: Theory, Algorithms and Applications*, Springer, New York, NY, USA, 2002.

[14] S. Mouchawrab, L. C. Briand, Y. Labiche, and M. Di Penta, "Assessing, comparing, and combining state machine-based testing and structural testing: a series of experiments," *IEEE Transactions on Software Engineering*, vol. 37, no. 2, pp. 161–187, 2011.

[15] K. El-Fakih, A. Simao, N. Jadoon, and J. C. Maldonado, "An assessment of extended finite state machine test selection criteria," *The Journal of Systems and Software*, vol. 123, pp. 106–118, 2017.

[16] A. T. Endo and A. Simao, "Evaluating test suite characteristics, cost, and effectiveness of FSM-based testing methods," *Information and Software Technology*, vol. 55, no. 6, pp. 1045–1062, 2013.

[17] J. F. Cutigi, A. Simao, and S. R. Souza, "Reducing fsm-based test suites with guaranteed fault coverage," *The Computer Journal*, vol. 59, no. 8, pp. 1129–1143, 2016.

[18] W.-W. Lin and H.-W. Zeng, "A chain algorithm for conformance testing based on uio sequences," in *Proceedings of the International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing*, pp. 1–6, IEEE, 2015.

[19] P. Liu, H.-K. Miao, H.-W. Zeng, and J. Mei, "DFSM-based minimum test cost transition coverage criterion," *Ruan Jian Xue Bao/Journal of Software*, vol. 22, no. 7, pp. 1457–1474, 2011.

[20] X.-P. Che and S. Maag, "A passive testing approach for protocols in internet of things," in *Proceedings of the 2013 IEEE International Conference on Green Computing and Communications and IEEE Internet of Things and IEEE Cyber, Physical and Social Computing*, pp. 678–684, IEEE, 2013.

[21] D. Serpanos and M. Wolf, "Security testing iot systems," *Internet-of-Things (IoT) Systems*, pp. 77–89, 2018.

[22] M. Tappler, B. K. Aichernig, and R. Bloem, "Model-Based Testing IoT Communication via Active Automata Learning," in *Proceedings of the 10th IEEE International Conference on Software Testing, Verification and Validation, ICST 2017*, pp. 276–287, IEEE, March 2017.

[23] A. Ahmad, F. Bouquet, E. Fourneret, F. L. Gall1, and B. Legeard, "Model-based testing as a service for iot platforms," in *Proceedings of the International Symposium on Leveraging Applications of Formal Methods*, pp. 727–742, Springer, 2016.

[24] H. Kim, A. Ahmad, J. Hwang et al., "IoT-TaaS: Towards a Prospective IoT Testing Framework," *IEEE Access*, vol. 6, pp. 15480–15493, 2018.

[25] J. Esquiagola, L. Costa, P. Calcina, G. Fedrecheski, and M. Zuffo, "Performance testing of an internet of things platform," in *Proceedings of the 2nd International Conference on Internet of Things, Big Data and Security, IoTBDS 2017*, pp. 309–314, Portugal, April 2017.

[26] K. Daniel, E. Reetz, and R. Tönjes, "Test derivation for semantically described iot services," in *Proceedings of the 2013 Future Network & Mobile Summit*, pp. 1–10, 2013.

[27] P. Rosenkranz, M. Wählisch, E. Baccelli, and L. Ortmann, "A Distributed Test System Architecture for Open-source IoT Software," in *Proceedings of the 2015 Workshop on IoT challenges in Mobile and Industrial Systems*, pp. 43–48, ACM, Florence, Italy, May 2015.

[28] K. Cui, K.-j. Zhou, T. Qiu, M.-c. Li, and L.-m. Yan, "A hierarchical combinatorial testing method for smart phone software in wearable iot systems," *Computers and Electrical Engineering*, vol. 61, pp. 250–265, 2017.

[29] F. R. Porto, A. T. Endo, and A. Simao, "Generation of checking sequences using identification sets," in *Proceedings of the International Conference on Formal Engineering Methods*, pp. 115–130, Springer, 2013.

[30] G.-V. Jourdan, H. Ural, and H. Yenigun, "Reduced checking sequences using unreliable reset," *Information Processing Letters*, vol. 115, no. 5, pp. 532–535, 2015.

[31] J. Baumgartner, Z. Sle, B. Bertk, and J. Abonyi, "Test-sequence optimisation by survival analysis," *Central European Journal of Operations Research*, pp. 1–19, 2018.

[32] R. M. Hierons, M. R. Mousavi, M. K. Thomsen, and U. C. Trker, "Hardness of deriving invertible sequences from finite state machines," in *Proceedings of the International Conference on Current Trends in Theory and Practice of Informatics*, pp. 147–160, Springer, 2017.

[33] A. Petrenko, "Toward testing from finite state machines with symbolic inputs and outputs," *Software and Systems Modeling*, pp. 1–11, 2017.

[34] R. M. Hierons and U. C. Turker, "Parallel Algorithms for Testing Finite State Machines:Generating UIO Sequences," *IEEE Transactions on Software Engineering*, vol. 42, no. 11, pp. 1077–1091, 2016.

[35] J. Song, H. Kim, and S. Park, "Enhancing Conformance Testing Using Symbolic Execution for Network Protocols," *IEEE Transactions on Reliability*, vol. 64, no. 3, pp. 1024–1037, 2015.

[36] P. Bokil, P. Krishnan, and R. Venkatesh, "Achieving effective test suites for reactive systems using specification mining and test suite reduction techniques," *ACM SIGSOFT Software Engineering Notes*, vol. 40, no. 1, pp. 1–8, 2015.