National Library of Canada
Acquisitions and Bibliographic Services Branch
395 Wellington Street
Ottawa, Ontario
K1A 0N4

Bibliothèque nationale du Canada
Direction des acquisitions et des services bibliographiques
395, rue Wellington
Ottawa (Ontario)
K1A 0N4

Your file Votre référence

Our file Notre référence

NOTICE

The quality of this microform is heavily dependent upon the quality of the original thesis submitted for microfilming. Every effort has been made to ensure the highest quality of reproduction possible.

If pages are missing, contact the university which granted the degree.

Some pages may have indistinct print especially if the original pages were typed with a poor typewriter ribbon or if the university sent us an inferior photocopy.

Reproduction in full or in part of this microform is governed by the Canadian Copyright Act, R.S.C. 1970, c. C-30, and subsequent amendments.

AVIS

La qualité de cette microforme dépend grandement de la qualité de la thèse soumise au microfilmage. Nous avons tout fait pour assurer une qualité supérieure de reproduction.

S'il manque des pages, veuillez communiquer avec l'université qui a conféré le grade.

La qualité d'impression de certaines pages peut laisser à désirer, surtout si les pages originales ont été dactylographiées à l'aide d'un ruban usé ou si l'université nous a fait parvenir une photocopie de qualité inférieure.

La reproduction, même partielle, de cette microforme est soumise à la Loi canadienne sur le droit d'auteur, SRC 1970, c. C-30, et ses amendements subséquents.

Canada

BOUNDS ON THE LENGTH OF TEST SEQUENCES FOR CONFORMANCE TESTING OF COMMUNICATION PROTOCOLS

by

Keqin Zhu

An M. Sc. Thesis

submitted to the School of Graduate Studies and Research
of the University of Ottawa
in partial fulfillment of the requirements for the degree of

Master of Computer Science*

Department of Computer Science
University of Ottawa
Ottawa, Ontario

* The Master of Computer Science program is a joint program with Carleton University, administrated by the Ottawa-Carleton Institute for Computer Science.

National Library
of Canada

Acquisitions and
Bibliographic Services Branch

395 Wellington Street
Ottawa, Ontario
K1A 0N4

Bibliothèque nationale
du Canada

Direction des acquisitions et
des services bibliographiques

395, rue Wellington
Ottawa (Ontario)
K1A 0N4

Your file Votre référence

Our file Notre référence

ISBN 0-315-80059-3

UNIVERSITÉ D'OTTAWA
UNIVERSITY OF OTTAWA

ABSTRACT

This study discusses the problem of the optimization of the length of test sequences for FSM-based protocol conformance testing. The discussion focuses on finding the extent to which the length of a test sequence for an FSM can be optimized, and obtaining algorithms to optimize the length of a test sequence for an FSM. For a special case (i.e., for a class of FSM's), we give the greatest lower bound (GLB) on the length of test sequences generated by any method employing a distinguishing sequence (D-method) without overlapping test segments. For the general case, we give lower bounds (i.e., approximations to the GLB) on the length of test sequences generated by any D-method without overlapping test segments as well as by any method employing a W-set (W-method) without overlapping test segments. We then establish a lower bound (i.e., an approximation to the GLB) on the length of test sequences generated by any D-method (or W-method) that overlaps test segments. It is observed that the reduction in the length of test sequences generated by any D-method (or W-method) due to overlapping is significant. Thus, two types of efficient algorithms based on overlapping test segments are established for both a method employing a distinguishing sequence and a method employing a W-set. The first type of efficient algorithms utilize the maximum-cardinality minimum-cost matching algorithm. The second type of efficient algorithms utilize the rural Chinese postman tour algorithm. Moreover, the sufficiency conditions for the second type of algorithms to find a minimum-length test sequence in polynomial time are indicated. Furthermore, the sufficiency conditions for the second type of algorithms to achieve the optimal length test sequences are discussed.

ACKNOWLEDGEMENTS

I would like to express my sincere gratitude to Professor Hasan Ural for his patience and guidance through the course of this study.

I especially wish to thank my wife Ling for her continuous encouragement.

I also would like to acknowledge Telecommunication Research Institute of Ontario (TRIO) & Natural Sciences and Engineering Research Council of Canada (NSERC) for their financial supports and the School of Graduate Studies and Research of the University of Ottawa for the Entrance Scholarship.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

1. INTRODUCTION

1.1 Background

Protocols are rules and conventions that define the possible interactions among the communicating entities in a communication network. In order to eliminate ambiguities and incompleteness in the protocol specification, protocols are typically modeled as extended finite-state machines [BOC82] where the control portion is a finite-state machine (FSM) and the data portion consists of program segments. The discussions on the data portion of protocols can be found in [URA91b][SAR87]. In this thesis, we will focus on the control portion of a communication protocol (henceforth referred to as protocol or *FSM-based protocol* for simplicity). The state of a protocol is defined as a stable condition in which the protocol rests until a stimulus, called an *input*, is applied. The protocol generates a response to the stimulus, called an *output*, (which may be null) when an input is applied, and moves into the next state (which may be the same as the previous state) where it stays until the next input is applied.

To ensure the successful exchange of information between the communicating entities, every protocol implementation must be tested for conformance against the specification of the protocol. This procedure is called *protocol conformance testing*. It is usually carried out by applying, by means of an upper and a lower tester, a selected sequence of inputs to an implementation under test (IUT) and verifying that the corresponding sequence of outputs is as expected according to the specification of the protocol [SAR87][AHO88][SID89b].

Practical limitations make the protocol conformance testing impossible to be exhaustive, and economic considerations may restrict testing still further [RAY87]. Therefore, only a partial behavior of an implementation will be checked for conformance with the protocol specification. It is assumed that by checking a limited set of behaviors of a protocol implementation, its conformance to the specification can be assured to an acceptable level [SID90b]. In general, the transition level behavior is chosen as the partial behavior to be checked for the implementation of an FSM [KOH78].

Various formal methods have been proposed to generate a sequence of inputs and outputs from an FSM-based protocol specification, known as a *test sequence*, which, when applied to an implementation of the FSM, is capable of verifying all specified transitions. These formal methods can be viewed as consisting of three phases. In the first phase, one may determine and derive some special input/output (I/O) sequences, called *state identification/verification (SIV) sequences*, for all states in the FSM such that each state is uniquely characterized by its SIV sequence(s). Some commonly used SIV sequences are derived from *distinguishing sequences (DS)* [GON70][KOH78], *characterizing sequences* [GIL62][CHO78][KOH78], and *unique input/output (UIO) sequences* [SAB85][SAB88]. In the second phase, a test segment for each transition in the FSM is formed. A *test segment* for a transition is a sequence of I/O's which consists of an input (with its expected output) that forces the FSM undergo the transition, and an SIV sequence that checks whether the FSM has reached the expected state. Finally, in the third phase, the test segments are joined to construct a test sequence for the FSM.

According to whether SIV sequences are used or not, or the kind of SIV sequences utilized, these formal methods can be classified as the *transition tour method* (or *T-method*) [NAI81][UYA86] where no SIV sequences are used, the *distinguishing sequence method* (or *DS method* or *D-method*) [KOH78][GON70], the *characterizing sequence method* (or *W-set method* or *W-method*) [CHO78][KOH78], and the *unique input/output sequences method* (or *UIO-method* or *U-method*) [SAB85][SAB88][AHO88] where the SIV sequences which are derived from the distinguishing sequence, the characterizing sequences and the unique input/output sequences, respectively, are used.

Except the T-method, all other formal methods reveal faults both in the output and in the next state function of the FSM. The test sequences generated by the T-method can only detect faults in the output function [SID89a]. The ability of detecting faults for a test sequence generated by a formal test sequence generation method is called the *fault coverage* of the method [SID89a]. In general, a test sequence generated by formal methods is expected to be as short as possible, while to cover as many faults as possible. For example, a modified W-method called *Wp-method* [FUJ91] is considered to be better than the W-method since the Wp-method generates shorter test sequences than those generated by the W-method while maintaining the same fault coverage as the W-method. Similarly, the UIO-method is considered to be better than the D- and W-methods [SID89a]. However, contrary to this claim, it is found that the actual fault coverage of the UIO-method is weaker than that of the D-method and the W-method [CHA89], since the uniqueness of UIO sequences may not hold in a faulty implementation of an FSM. A method called UIOv-method has been proposed to overcome the weakness of the UIO-method [CHA89].

Reductions in the length of the test sequences have been achieved by integrating graph theoretic approaches into the third phase of the formal methods, that is, combining test segments optimally, while maintaining the fault coverage of the formal methods. The method proposed by Uyar & Dahbura [UYA86] (henceforth called the *T/CPT method*) combines the T-method with the solution for the Chinese Postman Problem (CPP) [KUA62], for which efficient algorithms exist [EDM73]. The method proposed by Aho *et al.* [AHO88] (henceforth called the *SUIO/RCPT-method*) employs the solution for the Rural Chinese Postman Problem (RCPP) [KUA62] to optimize the test sequence generated by the UIO-method. Finding the solution for the Rural Chinese Postman Problem is known to be NP-complete in the most general case [LEN76]. However, it is shown in [AHO88] that a polynomial-time algorithm exists for determining a minimum-cost test sequence if the graph representation of the protocol possesses certain structure (i.e. the protocol satisfies certain sufficiency conditions). The solution for the Rural Chinese Postman Problem has also been applied to the D-method and the W-method, which forms the *D/RCPT-method* and the *W/RCPT-method*, respectively [URA91a].

Shen *et al.*'s method [SHE89] (henceforth called the *MUIO/RCPT-method*) attempts to further reduce the length of the test sequences generated by the SUIO/RCPT-method. It improves the SUIO/RCPT-method by calculating multiple UIO sequences for each state of a given FSM in the first phase of the formal method, and forming test segments through selecting the proper UIO sequence to follow each transition in the second phase of the formal method. The result is a reduction of 4%-37% in the length of the test sequences required in [AHO88] with no noticeable increase in the time required to generate the test sequence [SHE89]. However, there are problems remained unsolved due to the manner multiple UIO sequences are used in the MUIO/RCPT-method, such as the sufficiency condition for a polynomial-time algorithm to solve RCPP in the SUIO/RCPT-method is not met in the MUIO/RCPT-method, and the systematic selection of appropriate UIO's proposed in the MUIO/RCPT-method does not necessarily minimize the length of the resulting test sequence. These problems are resolved by an improved MUIO/RCPT-method proposed by Ural & Lu [URA90], which constructs test sequences that are at least as short as those that can be obtained by the MUIO/RCPT-method [URA90].

As indicated in [AHO88], further reductions in the length of a test sequence can be achieved by overlapping test segments. However, it has been proved that the problem of constructing the optimal length test sequence through finding maximal overlapping of test segments is NP-complete [BOY91]. Different heuristic approaches to generate optimal length test sequences by overlapping test segments for a given FSM have been proposed by Yang and Ural [YAN90], M.S. Chen *et al.* [CHE90], and Miller and Paul [MIL90].

1.2 Motivation & Objectives

The results achieved in optimizing the length of test sequences have been promising in reducing the time required to test a protocol implementation [SID90a]. In this thesis, we study the extent to which the length of test sequences for FSM-based protocol conformance testing can be optimized and provide algorithms to optimize the length of test sequences for FSM-based protocol conformance testing. Our objectives are to:

a) find lower bounds on the length of test sequences generated by any D-method and any W-method with or without overlapping test segments.

b) develop polynomial-time algorithms that are based on overlapping test segments and approach the lower bounds given in a).

1.3 Contributions

The main contributions of this thesis are as follows:

a) lower bounds on the length of test sequences generated by any D-method and any W-method with or without overlapping test segments are established.

b) two types of efficient algorithms based on overlapping test segments are established for both a method employing a DS and a method employing a W-set.

    (1) The first type of efficient algorithms utilize the *maximum-cardinality minimum-cost matching algorithm* [PAP82].

    (2) The second type of efficient algorithms utilize the *rural Chinese postman tour algorithm* [AHO88].

c) the sufficiency conditions for the second type of algorithms to find a minimum-length test sequence in polynomial-time are indicated.

d) the sufficiency conditions for the second type of algorithms to achieve the optimal length test sequences are discussed.

1.4 Organization of the Thesis

The organization of this thesis is as follows: In chapter 2, related terminology is introduced, and protocol conformance testing is reviewed. Chapter 3 gives the proofs on the lower bounds of the length of test sequences generated by any D-method with or without overlapping test segments, and presents two types of algorithms for a method employing a DS that overlaps test segments. Chapter 4 provides the proofs on the lower bounds of the length of test sequences generated by any W-method with or without overlapping test segments, and describes two types of algorithms for a method employing a W-set that overlaps test segments. Finally, conclusions and directions for future research are given in chapter 5.

2. PRELIMINARIES

2.1 FSM Model and its Graph Representation

An FSM M can be characterized by a quintuple M=(S, I, O, δ, λ), where

S : is a finite set of states $\{s_1, s_2, \ldots, s_n\}$;

I : is a finite set of input symbols $\{i_1, i_2, \ldots, i_q\}$;

O : is a finite set of output symbols $\{o_1, o_2, \ldots, o_r\}$;

δ : is a mapping S x I -> S, called *state transition function*;

λ : is a mapping S x I -> O, called *output function*;

A state transition from state s_j to state s_k in M is denoted by $t_{jk} = (s_j, s_k; i/o)$. The *initial state* $s_1 \in S$ of M is a designated state for M. Often, there is a designated input *ri* $\in$ I, called *reset input*, which brings M to s_1 from every state with a single transition producing a *null* output, i.e., null $\in$ O. An FSM M which has such a reset input is said to possess *reset feature*. An FSM M is said to possess *self-loop feature* if for every state in M, there is a transition starting and ending at that state.

An FSM M is said to be *minimal* if every pair of its states does not produce identical output sequences when excited by any sequence of input symbols [KOH78]. An FSM M is said to be *deterministic*, if for each input i $\in$ I, there is at most one transition defined at each state of M. An FSM M is said to be *completely specified*, if for each input i $\in$ I, there is a transition defined at each state of M.

An FSM M can be represented by a directed graph G=(V, E) where a set of vertices $V=\{v_1,v_2,\cdots,v_n\}$ represents the set S of states of M, and a set of directed edges $E=\{(v_j,v_k;i/o): v_j,v_k \in V\}$ represents all specified transitions of M. Each edge $e_{jk}=(v_j,v_k;i/o) \in E$, represents a state transition from s_j to s_k with input i and output o, where v_j, v_k, and i/o are called the *head*, the *tail*, and the *label* of the edge e_{jk}, and denoted by Head(e_{jk}), Tail(e_{jk}) and Label(e_{jk}), respectively. A state transition is called *a-transition* if the input symbol associated with the label of the transition is *a*.

In the following, G and M, vertices and states, edges and transitions are used interchangeably.

2.2 Graph Theoretic Definitions

In this section, definitions related to a directed graph D=(V, E) will be given. It is assumed that the label of an edge in D can be empty (denoted by Ø) or a sequence of input/output pairs, $i_1/o_1, i_2/o_2, \ldots, i_k/o_k$, k > 0, denoted by L. The *length of the label L* of an edge in D is defined as the number of input/output pairs in L.

Let L, L_1 and L_2 be sequences of input/output pairs:

$|L|$ represents the *length* of L (i.e., the number of input/output pairs in L).

$L_1@L_2$ represents the *concatenation* of L_1 and L_2, which forms a new sequence such that L_1 is followed by L_2.

L_1-L_2 represents the *deduction* of L_2 from L_1, which forms a new sequence from L_1 such that L_2 in L_1 is removed.

$|Q|$, $Q_1@Q_2$, Q_1-Q_2 are defined similarly for Q, Q_1, Q_2 that denote sequences of only input or only output symbols.

Given a directed graph D=(V, E), the *indegree* and *outdegree* of a vertex v of D is defined as: $d_{in}^{E}(v) = |\{(w,v; L) : (w,v; L) \in E\}|$ and $d_{out}^{E}(v) = |\{(v,w; L) : (v,w; L) \in E\}|$. The *index* $\xi^{E}(v)$ of a vertex $v \in V$ is defined as: $\xi^{E}(v) = d_{in}^{E}(v) - d_{out}^{E}(v)$. D is said to be *symmetric*, if for every $v \in V$, $d_{in}^{E}(v) = d_{out}^{E}(v)$, i.e., $\xi^{E}(v) = 0$.

A *path* $P=(v_1, v_2; L_1)(v_2, v_3; L_2) \ldots (v_{k-1}, v_k; L_{k-1})$, $k > 1$, is a finite sequence of adjacent and not necessarily distinct edges, where v_1, v_k, and $L_1@L_2@\ldots@L_{k-1}$ are called the ***head***, the ***tail***, and the ***label*** of the path P, and denoted by Head(P), Tail(P) and Label(P), respectively.

D is said to be *strongly connected*, if for every pair of vertices v_j and v_k, there exists a path from v_j to v_k. D is said to be *weakly connected* if its underlying undirected graph is connected.

The *cost* or *length of each edge* of D is equal to the length of its label. The *cost of a path* (or *length of a path*) P in D is the sum of the costs (or lengths) of edges included in P. A path with the minimum length among all paths from v_s to v_t is called a *shortest path* from v_s to v_t.

A *tour* in D is a path in D which starts and ends at the same vertex of D. An *Euler tour* of D is a tour which contains every edge in E exactly once. A *postman tour* (PT) of D is a tour which contains every edge in E at least once. A *rural postman tour* (RPT) of $D=(V, E)$ over a set $E_C \subseteq E$ is a tour traversing every edge in E_C at least once. A *Chinese postman tour* (CPT) of D is a minimum-cost tour which contains every edge in E at least once. A *rural Chinese postman tour* (RCPT) of $D=(V, E)$ over a set $E_C \subseteq E$ is a minimum-cost tour traversing every edge in E_C at least once.

A graph $D=(V, E)$ is a *subgraph* of $D'=(V', E')$ if $V \subseteq V'$ and $E \subseteq E'$. A subgraph D which contains all the vertices of a graph D' (i.e., V=V') is called a *spanning subgraph* of the graph D'. An *edge-induced subgraph* $D[E_C]$ of D' is the subgraph of D' whose vertex set is the set of heads and tails of edges in E_C, and whose edge set is E_C where $E_C \subseteq E'$ [AHO88]. An edge-induced subgraph $D[E_C]$ of D' is called an *edge-induced spanning subgraph* of D' if its vertex set is V'.

A *matching* H of a directed graph D=(V, E) is a subset of the edges with the property that no two edges of H share the same vertex [PAP82].

A *bipartite graph* is defined as a directed graph D=(V, E), where $V=X \cup Y$, $X \cap Y = \phi$ and each edge has its head in X and its tail in Y. D is said to be a *complete bipartite graph* if D is a bipartite graph, and each $v_i \in X$ is connected to each $v_j \in Y$ by an edge.

2.3 Optimization Problems on Graphs

2.3.1 The Rural Chinese Postman Problem

Given a directed graph D=(V, E) and a set $E_C \subseteq E$, the *Rural Chinese Postman Problem* (RCPP) is to find an RCPT of D over the set E_C [AHO88].

It is known that computing an RCPT is NP-complete in the most general case [LEN76]. Nevertheless, if D is strongly connected and the edge-induced spanning subgraph D[E_C] is weakly connected, then any *minimum-cost rural symmetric augmentation* D* of D over E_C is strongly connected [AHO88]. Here, D*=(V*, E*) is a *minimum-cost rural symmetric augmentation* (RSA) of D over E_C if D* is symmetric, V*=V, and E* contains every edge in E_C at least once and every edge in E-E_C zero or more times such that the total cost of edges in E* is minimized. Since a strongly connected and symmetric digraph has an Euler tour [EDM73], an RSA D* has an Euler tour which is an RCPT of D over E_C. Therefore, in this special case, the RCPP can be solved in two steps:

(1) constructing an RSA D* of D over E_C;

(2) finding an Euler tour of D*.

Efficient algorithms exist for both of the above steps [AHO88].

If $D[E_C]$ is not weakly connected, edges from E can be added to E_C to obtain E_C^* such that $D[E_C^*]$ is weakly connected. An Euler tour of the RSA D^* of D over E_C^* is then an RPT of D over E_C, but not necessarily an RCPT.

2.3.2 The Maximum-Cardinality Minimum-Cost Matching Problem

Given a bipartite graph $D=(V, E)$ where $V=X \cup Y$, $|X|=|Y|$ (*i.e.*, a complete bipartite graph with two sets of nodes that are equal in size), the *maximum-cardinality minimum-cost matching problem* is to find a matching with the maximal cardinality $|X|$ and the smallest possible sum of costs [PAP82].

As indicated in [PAP82], a bipartite graph that is not complete can always be transformed into a complete bipartite graph by simply letting the costs of those edges that were missing in the noncomplete bipartite graph be equal to a number that is large enough. Then, for a complete bipartite graph with two sets of nodes that are equal in size, the maximum-cardinality minimum-cost matching problem can be solved in $O(|V|^3)$ [PAP82].

2.4 Protocol Conformance Testing

Protocol conformance testing is to check if a protocol implementation under test (IUT) has the same input/output behavior as that of the specification of the protocol (henceforth denoted by FSM). This is often carried out (1) in a test architecture, where the IUT is tested as a black-box by two remote testers, called the *lower tester* and the *upper tester*; (2) by applying, by means of the upper and the lower testers, a selected sequence of inputs to the IUT and verifying that the corresponding sequence of outputs is as expected according to the FSM. If two sequences of outputs match, the IUT is said to *conform* to the FSM. Otherwise, the IUT is said to be *faulty*.

2.4.1 A Testing Procedure

In general, a testing procedure will only check a limited set of behaviors of an IUT for conformance with an FSM, since an exhaustive testing of a protocol (i.e., checking all possible input/output behaviors) is not feasible. However, it is not always clear which subset of behaviors is meaningful to ensure the conformance of an IUT to the FSM it implements [SID90].

Current research on protocol conformance testing has included only those behaviors that correspond to testing of the states and individual transitions of an FSM-based protocol. For example, a transition-level approach originating from sequential circuit checking experiment [KOH78] has been adopted by most formal methods for generating a selected sequence of inputs and corresponding outputs. Accordingly, the procedure of checking whether a transition $t_{jk}=(s_j,s_k;i/o)$ of a given FSM is correctly implemented by an IUT consists of three steps:

1) The IUT is brought to state s_j;
2) Input i is applied to the IUT and the output produced by the IUT is checked whether it is o;
3) The state the IUT reaches after the application of input i is checked whether it is s_k.

Steps 2) and 3) are used to detect the errors in the output function (i.e., output errors) and in the state transition function (i.e., transition or tail state errors), respectively. However, this procedure is complicated by the limitations on the controllability and the observability of the IUT due to the black box nature of protocol conformance testing. Because of the limited controllability, the IUT cannot be directly put into a desired state and thus a *transfer sequence* (i.e., the shortest sequence of inputs to bring the IUT to a desired state, s_j in step 1)) is usually required. Limited observability prevents the direct observation of the state of the IUT, and thus an SIV sequence (i.e., a sequence of inputs and outputs to verify that the IUT reached a designated state, s_k in step 3)) is required.

Therefore, the input sequence required to test the transition $t_{jk}=(s_j,s_k;i/o)$ consists of

T_a = an optional transfer sequence to bring the IUT from its current state to state s_j;
T_b = input symbol i for stimulating transition t_{jk};
T_c = the input portion of an SIV sequence for state s_k.

The sequence of inputs (together with their expected outputs) that force an IUT to undergo all transitions of a given FSM is called a ***test sequence*** (TS) for the FSM. The sequence of inputs (together with their expected outputs) which cover T_b and T_c portions of an input sequence to test a transition is called a ***test segment*** (denoted by S_i, $1 \leq i \leq |E|$) for the transition. For convenience, sometimes only the input sequence is mentioned as a test sequence or test segment.

2.4.2 SIV Sequences and Formal Methods

As mentioned in section 1.1, the most commonly used SIV sequences are derived from distinguishing sequences (DS), characterizing sequences and unique input/output (UIO) sequences.

A sequence of inputs is said to be a ***distinguishing sequence (DS)*** of FSM M if the output sequence produced by M in response to the input sequence is different for each state of M. A distinguishing sequence may not exist for every FSM [KOH78]. An SIV sequence for a state based on a DS for the given FSM (henceforth denoted by $DS(s_i)$ for state s_i) is a sequence of inputs and corresponding outputs where the sequence of inputs is the DS.

A set of *characterizing sequences (W-set)* for an FSM M is a set of input sequences $w_1,w_2,\ldots,w_\mu$ such that each state is uniquely identified by the set of output sequences produced by M in response to the set of input sequences [GIL62][CHO78]. There always exists a set of characterizing sequences for a given FSM. A W-set for an FSM M is said to be a *minimal W-set* if none of its components is a prefix of another component [GIL62]. A set of SIV sequences for a state can be derived from a W-set (henceforth denoted by W-set(s_i) for state s_i), each of which is a sequence of inputs and corresponding outputs (henceforth denoted by $w_k(s_i)$, $k=1,2,\ldots,\mu$ for state s_i) where the sequence of inputs is one of the input sequences w_k in the W-set.

A *unique input/output (UIO) sequence* for a state of FSM M is an I/O behavior that is not exhibited by any other state of M. UIO sequences may not exist for some states of a given FSM [SAB88]. In that case, a *unique signature* [SAB88] is constructed for each state which does not have a UIO sequence. An SIV sequence for a state based on a UIO sequence or a unique signature for that state (henceforth denoted by UIO(s_i) for state s_i) is simply a UIO sequence or a unique signature for that state.

If an SIV sequence SIV is used in a formal method for test sequence generation, the corresponding formal method is classified as SIV-method. In general, the test segment for the transition $t_{jk}=(s_j,s_k;i/o)$ is defined as i/o@SIV(s_k) where @ is the string concatenation operator.

2.4.3 Optimization of Test Sequences

It should be noted that the formation of a test sequence may be subject to certain constraints, such as the type of SIV sequences and the number of SIV's for each state employed to construct test segments, the existence of a requirement on the starting and terminating states of the test sequence, and whether overlapping between test segments is allowed. Based on these constraints, different optimization problems may be formed. Therefore, a test sequence for an FSM M is called *optimal* if no shorter test sequence for M satisfying the same set of constraints exists.

In the following, we assume that a deterministic, minimal, completely specified FSM M is represented by a strongly connected digraph G=(V, E) in which the cost (or length) of each edge is one. Also, it is assumed that the test sequence derived from an FSM M is a tour, which starts and ends at the initial state s_1 of M. This implies the assumption that a given IUT always starts from its initial state in the conformance testing. Finally, *minimum-length* or *minimum-cost* test sequence will be specifically used to refer to a test sequence derived by an algorithm using certain optimization approaches, and a minimum-length or minimum-cost test sequence may or may not be an optimal test sequence.

There are two kind of optimization problems for test sequences with respect to whether overlapping between test segments is allowed. The first one is to find an optimal test sequence for an FSM M in which no two test segments overlap and only one SIV for each state is used (henceforth called *OTS problem*). To solve the OTS problem, a new directed graph G' =(V', E') is constructed from the directed graph G=(V, E) representing a given FSM M, where:

$V'=V$,

$E'=E \cup E_C$,

$E_C=\{(v_j, v_z; i/o@SIV(v_k)) : (v_j, v_k; i/o) \in E, i \in I, o \in O, \text{ and } Tail(SIV(v_k))=v_z \in V\}$

(i.e., $e_{jz} \in E_C$ represents a test segment for the transition $e_{jk} \in E$).

The cost or length of $(v_j, v_z; i/o@SIV(v_k))$ is equal to the length of $SIV(v_k)$ plus one according to the definition given in section 2.2. Therefore, the minimum-cost test sequence which contains all test segments in M such that no two test segments are overlapped corresponds to the minimum-cost tour of G' such that each edge in E_C is traversed at leas once. Such a tour is a rural Chinese postman tour (RCPT) of G' over E_C. Computing such a tour is known to be NP-complete in the most general case [LEN76]. Nevertheless, as discussed in section 2.3.1, if the edge-induced spanning subgraph G[E_C] of G' is weakly connected, then finding an RCPT of G' over E_C is reduced to:

(1) constructing an RSA G* of G' over E_C, where G* is symmetric, V*=V', and E* contains every edge in E_C at least once and every edge in E zero or more times such that the total cost of edges in E* is minimized;

(2) finding an Euler tour on G*.

Both problems can be solved by efficient algorithms [AHO88]. As it was shown in [AHO88][SID90b], as long as G possesses reset feature, or self-loop feature, G[E_C] of G' is a weakly connected spanning subgraph of G'. Thus, an Euler tour of the RSA G* of G' is an RCPT of G' over E_C which in turn is a minimum-cost test sequence for G.

The other optimization problem is to find an optimal test sequence for an FSM M in which test segments are allowed to overlap and one or several SIV's for each state can be used (henceforth called *OTSO problem*). Given the graph representation G of an FSM M, a *checking path* for a state $v_k \in V$ is a path which starts from v_k and corresponds to an SIV sequence for v_k. A *checking traversal* of G is a tour which includes at least one occurrence of every transition $e_{jk}=(v_j,v_k;i/o) \in E$ which is immediately followed by a checking path of v_k.

In graph-theoretic terms, the OTSO problem can be defined as follows: given the graph representation G of an FSM M and, for each state, one or more checking paths, find the shortest checking traversal of G which starts and ends with v_1. It was shown [BOY91] that OTSO problem is NP-complete. Therefore, only heuristic approaches may be found to reduce the length of a test sequence by attempting to overlap test segments.

One such heuristic approach is to reduce the OTSO problem to a maximum-cardinality minimum-cost matching problem. It was first proposed in [CHE90] where a single UIO sequence is used as an SIV sequence for each state. This approach can be extended to use other SIV sequences for each state. To formulate the OTSO problem, a new directed graph $G'=(V', E')$ is constructed from the graph representation G of an FSM M, where

$V'=V \cup U$,

$U=\{u_i : v_i \in V\}$,

$E'=E \cup E_C \cup E_v \cup E_o$,

$E_C=\{e_{jz}=(u_j,u_z;i/o@SIV(v_k)) : (v_j, v_k; i/o) \in E, i \in I, o \in O, \text{and } Tail(SIV(v_k))=v_z \in V\}$,

$E_v=\{(v_j,u_j;\emptyset) : v_j \in V, u_j=Head(e_{jz}) \in U, e_{jz} \in E_C\} \cup$

$\{(u_z,v_z;\emptyset) : v_z \in V, u_z=Tail(e_{jz}) \in U, e_{jz} \in E_C\}$,

$E_o=\{(u_z,u_j;\emptyset) : u_z,u_j \in U, e_{xz}, e_{jy} \in E_C, u_z=Tail(e_{xz}), u_j=Head(e_{jy})$, the test segment corresponding to e_{jy} overlaps the test segment corresponding to $e_{xz}\}$.

A test segment S_j is said to *overlap* another test segment S_i if the possible overlapping part between S_i and S_j is not null when S_i is immediately followed by S_j. The cost or length of $(u_j, u_z; i/o@SIV(v_k)) \in E_C$ is equal to the length of $SIV(v_k)$ plus one, and the cost or length of $(v_j,u_j;\emptyset)$ or $(u_z,v_z;\emptyset) \in E_v$ is zero, but the cost or length of $(u_z,u_j;\emptyset) \in E_o$ is defined as a negative number equal to the length of the corresponding overlapping part.

Informally, each edge $e_{jz} \in E_C$ represents a test segment for each transition $e_{jk} \in E$, each edge $(v_j,u_j;\emptyset)$ or $(u_z,v_z;\emptyset) \in E_v$ represents a link between U and V, and each edge $(u_z,u_j;\emptyset) \in E_o$ represents the possible overlapping part with which the test segment $e_{jy} \in E_C$ can overlap the test segment $e_{xz} \in E_C$. This formulation enforces the constraint that a tour on G' must have the segment S_i immediately followed by the segment S_j if the overlapping part with which S_j can overlap S_i is used. Therefore, the optimal test sequence which contains all test segments in FSM M (allowing two segments overlapped) corresponds to an RCPT of G' over E_C. However, G[E_C] is no longer weakly connected because of the additional nodes from U, and the optimum solution is in general not obtainable [CHE90]. Nevertheless, if segments have significant overlapping, efficient heuristics that utilize overlapping are still likely to yield better results than the exact solutions used in solving OTS problem [CHE90]. It is shown in [CHE90] that the so formulated RCPP can be solved as a maximum-cardinality minimum-cost matching problem in a bipartite graph. If the solution is a tour, then it is a valid and optimal test sequence. Otherwise, it is possible that only a minimum-cost test sequence can be obtained via heuristics to connect these resulting disconnected tours.

3. D-METHOD

In the D-method [GON70, KOH78, SAR84, SID89a], a test segment for each transition consists of the transition and the distinguishing sequence (DS) which follows the tail of the transition. The test sequence for an FSM M is then constructed by combining test segments for all transitions of M. Some transfer sequences may be required for connecting two consequent test segments. If test segments are overlapped, the result of overlapping two test segments will not only eliminate the transfer sequence for connecting them, but also the overlapping part of two test segments will be used only once.

3.1 Lower Bounds on the Length of Test Sequences

The main goal of this section is to find a lower bound on the length of test sequences generated by any method employing a DS with overlapping, i.e., a lower bound on the length of test sequences for the OTSO problem employing a DS. The OTSO problem employing a DS is called the *OTSO_D problem* henceforth, and the lower bound on the length of test sequences for the OTSO_D problem is denoted by LB_{OTSO_D}. Similarly, the OTS problem employing a DS is called the *OTS_D problem*, and the lower bound on the length of test sequences for the OTS_D problem is denoted by LB_{OTS_D}.

In fact, the OTSO_D problem is related to the OTS_D problem since overlapping of two test segments in the OTSO_D problem will not only eliminate the possible transfer sequence for connecting them which is required in the OTS_D problem, but also the overlapping part of two test segments will be used only once. Therefore, we reduce the problem of finding LB_{OTSO_D} to the problem of

(a) finding a lower bound on the length of test sequences for the OTS_D problem, i.e., LB_{OTS_D}.

(b) calculating the sum of the length of the possible overlapping parts between any two test segments (denoted by $TSOP_D$), and the sum of the length of the possible transfer sequences eliminated by overlapping (denoted by $TROP_D$).

Thus, $LB_{OTSO\_D} = LB_{OTS\_D} - TSOP_D - TROP_D$.

The following theorem establishes a lower bound on the length of test sequences for the OTS_D problem.

Theorem 1: Given a strongly-connected digraph G=(V, E) representing a deterministic, minimal, and completely-specified FSM M=(S, I, O, δ, λ), |S|=n, |I|=q, with a DS of length L. Let $E_C=\{(v_j, v_z; i/o@DS(v_k)) : (v_j, v_k; i/o) \in E, i \in I, o \in O, \text{ and } Tail(DS(v_k))=v_z \in V\}$,

(a) If $G[E_C]$ is weakly connected, then the length of a test sequence for the OTS_D problem can not be below

$$nq(1+L) + \alpha$$

where

$\alpha=|\{e : e \in E^* - E_C\}|$; that is, the total length of those edges added to $G[E_C]$ for constructing an RSA $G^*=(V^*, E^*)$ of G' over E_C.

(b) Otherwise, the length of a test sequence for the OTS_D problem can not be below

$$nq(1+L) + \tau - \beta$$

where

$\tau = \sum |\xi^{E_C}(v_i)|,\ v_i \in V,\ \xi^{E_C}(v_i) < 0;$

$\beta = 1$, if $\xi^{E_C}(v_1) < 0;$

$\beta = 0$, if $\xi^{E_C}(v_1) \geq 0.$

***Proof*:** nq is the number of transitions, and (1+L) is the length of the test segment for each transition. Without overlapping, the total length of the test sequence should be at least equal to the total length of all test segments, nq(1+L).

(a) From the discussion in section 2.4, the minimum-length test sequence is equal to a minimum-length tour in G' that traverses every edge in $G[E_C]$ at least once; i.e., it is equal to a rural Chinese postman tour (RCPT) of G' over E_C. Computing such a tour is known to be NP-complete in the most general case [LEN76].

However, due to the fact that $G[E_C]$ is weakly connected, any minimum-cost rural symmetric augmentation (RSA) G* of G' over E_C is strongly connected [AHO88]. Since a strongly connected and symmetric digraph has an Euler tour [EDM73], the problem of finding an RCPT of G' over E_C is reduced to constructing an RSA G* of G' over E_C and finding an Euler tour of G*. So, at least a total length of $|\{e : e \in E^* - E_C\}|$ of transfer sequences has to be used for constructing a test sequence.

Therefore, the length of a test sequence for the OTS_D problem can not be below

$$nq(1+L) + \alpha \qquad \text{where } \alpha = |\{e : e \in E^* - E_C\}|.$$

(b) For $v_i \in V$, and $\xi^{E_C}(v_i) < 0$, to test all transitions initiating from v_i, $|\xi^{E_C}(v_i)|$ transfer sequences, each of which consists of at least one transition, will be employed to transfer M to v_i. Thus, at least an extra length $\sum |\xi^{E_C}(v_i)|$, $v_i \in V$, $\xi^{E_C}(v_i) < 0$, of transfer sequences has to be used for constructing a test sequence. If $\xi^{E_C}(v_1) < 0$, 1 should be deducted from the total added length of transfer sequences since no transfer sequence is required for the first test segment starting from v_1.

Therefore, the length of a test sequence for the OTS_D problem can not be below

$$\begin{array}{ll} nq(1+L) + \tau - 1 & \text{if } \xi^{E_C}(v_1) < 0, \\ nq(1+L) + \tau & \text{if } \xi^{E_C}(v_1) \geq 0, \end{array}$$

where $\tau = \sum |\xi^{E_C}(v_i)|$, $v_i \in V$, $\xi^{E_C}(v_i) < 0$. ♦

In Theorem 1, (a) gives the greatest lower bound on the length of test sequences if $G[E_C]$ is weakly-connected. However, if $G[E_C]$ is not weakly-connected, it is not possible to find the greatest lower bound on the length of the test sequences. Therefore, Theorem 1(b) gives an approximation to the greatest lower bound in the general case, including the case where $G[E_C]$ is weakly-connected. Furthermore, the way to calculate the lower bound given in Theorem 1(b) provides the relevant information for finding the lower bound on the length of test sequences for the OTSO_D problem. So, the lower bound given in Theorem 1(b) will be used as LB_{OTS_D} henceforth.

In fact, the lower bound on the length of test sequences consists of two parts: the *minimum requirement on the length of test segments* (denoted by MRTS) and the *minimum* *requirement on the length of transfer sequences* (denoted by MRTR). In Theorem 1, MRTR=α , α=|{e : e $\in$ E* - E_C }| if G[E_C] is weakly-connected; otherwise, MRTR= τ - β where $\tau=\sum |\xi^{E_C}(v_i)|$, $v_i \in$ V, $\xi^{E_C}(v_i) < 0$ (henceforth denoted by $MRTR_{OTS\_D}$). Meanwhile, MRTS is always equal to nq(1+L) (henceforth denoted by $MRTS_{OTS\_D}$).

It is obvious that if G[E_C] is strongly-connected and symmetric, MRTR will be 0 and the lower bound becomes nq(1+L). Indeed, this is the best case for the test sequences generated by any method employing a DS without overlapping.

Since $LB_{OTSO\_D} = LB_{OTS\_D} - TSOP_D - TROP_D$, what need to be found next is $TSOP_D$ and $TROP_D$ for establishing the lower bound on the length of test sequences for the OTSO_D problem, i.e., LB_{OTSO_D}. Before we give a theorem to establish LB_{OTSO_D}, we will make some observations highlighting the analysis required for the derivation of $TSOP_D$ and $TROP_D$, and define the terms that will be used in the theorem. First, the following observations are made:

1) If test segment S_b can overlap test segment S_a with a length at most l_{op}, then the length l_{op} should be added to $TSOP_D$. l_{op} can be determined in two steps:

    a) In the first step, only the input portion of each test segment for an FSM is considered. Thus, the prefix of each test segment with which the test segment can potentially overlap another test segment can be identified. Such a prefix of the maximum length for each test segment is called the *potential maximal overlapping part* of the test segment. The corresponding length is called the *potential maximal overlapping length.*

    b) In the second step, test segments that can never overlap other test segments with their potential maximal overlapping length are identified. Thus, an *adjustment factor for the potential maximal overlapping length* for each of these test segments are determined and deducted from its potential maximal overlapping length. The result is the l_{op} for the test segment.

2) If test segment S_b can overlap test segment S_a, and S_b starts from $v_i \in V$, $\xi^{E_C}(v_i) < 0$, then the length of the transfer sequence which is required for the OTS_D problem and counted in $MRTR_{OTS\_D}$ of LB_{OTS_D} should be added to $TROP_D$ since the transfer sequence will be eliminated due to the overlapping.

In order to make use of these observations, the following definitions are needed: let the set of input symbols I in an FSM be $\{a_1, a_2, \ldots, a_q\}$. Then, there are only q forms of input portions a_j@DS (j=1,...,q) among nq test segments for an FSM. Henceforth, a test segment with the input portion a_j@DS is called a *test segment with the form* a_j@DS (denoted by test segment [a_j@DS]).

Formally, we introduce the following terms for a DS for a given FSM. Z_{ji} is said to be the *potential overlapping suffix* of the DS w.r.t. a_j@DS ($1 \le j \le q$), if Z_{ji} is the suffix of the DS and the prefix of a_j@DS, where $1 \le i \le b_j$. b_j is the number of the potential overlapping suffixes of the DS w.r.t. a_j@DS ($1 \le j \le q$). Z_{ji} (j=1,...,q, i=1,...,b_j) are listed in such a way that $|Z_{j1}| > |Z_{j2}| > \ldots > |Z_{jb_j}|$. Z_{j1} is said to be the *potential maximal overlapping suffix* of a DS w.r.t. a_j@DS (henceforth denoted by Z_j). The corresponding length is called the *potential maximal overlapping length* of the DS w.r.t. a_j@DS. Accordingly, X_j is said to be the *intrinsically nonoverlapping prefix* of a DS w.r.t. a_j@DS, if X_j is a prefix of a DS and X_j = DS - Z_j.

Similarly, we introduce the following terms for each test segment [a_j@DS] ($1 \le j \le q$) for a given FSM. Z_j is said to be the *potential maximal overlapping part* of the test segment [a_j@DS] if Z_j is a maximum length prefix of the test segment with which the test segment can potentially overlap another test segment. The corresponding length is called the *potential maximal overlapping length* of the test segment [a_j@DS], and denoted by z_j. It is clear that n test segments with the same form a_j@DS ($1 \le j \le q$) initiating from n different states will have the same potential maximal overlapping length.

A DS of length L can only be in one of the following three formats:

(1) $a_1 a_1 \cdots a_1$, that is, the DS consists of L repeated input symbols (without loss of generality, it can be assumed that the input symbol is a_1);

(2) X_1@a_1, where X_1 is the intrinsically nonoverlapping prefix of the DS w.r.t. a_1@DS (without loss of generality, it can be assumed that the last input symbol in the DS is a_1).

(3) D_{k1}@D_{k2}=D_{k3}@D_{k1}=X_k@a_k@D_{k1} (k=1,...,h), where $|D_{k2}|=|D_{k3}| > 1$; at least one D_{k1} satisfying $|D_{k1}| \ge 1$; at most one $D_{k1}=\emptyset$; and X_k is the intrinsically nonoverlapping prefix of the DS w.r.t. a_k@DS (without loss of generality, it can be assumed that $k=1,\ldots,h \le q$, i.e., a_k (k=1,...,h) are the first h input symbols in I, and if there is one $D_{k1}=\emptyset$, then $D_{h1}=\emptyset$);

It is clear that a DS in format (2) can be written as X_1@a_1@D_{11}, where $D_{11}=\emptyset$. Also, a DS in format (1) can be written as X_1@a_1@D_{11}, where $X_1=\emptyset$, $D_{11}=a_1 \cdots a_1$.

The following theorem gives a lower bound on the length of test sequences for the OTSO_D problem and formally presents the observations given above in detail.

Theorem 2: Given a strongly-connected digraph G=(V, E) representing a deterministic, minimal, and completely-specified FSM $M=(S, I, O, \delta, \lambda)$, $|S|=n$, $|I|=q$, with a DS of length L. The length of a test sequence for the OTSO_D problem can not be below

$$[nq(1+L) + \tau - \beta] - [(n \sum_{j=1}^{q} z_j - z_r) - \sum_{j=1}^{q} \sum_{i=1}^{n} y_{ji}] - t$$

where

$\tau=\sum |\xi^{Ec}(v_i)|,\ v_i \in V,\ \xi^{Ec}(v_i) < 0;$

$\beta=1$, if $\xi^{Ec}(v_1) < 0;$
$\beta=0$, if $\xi^{Ec}(v_1) \geq 0;$

z_j, $1 \leq j \leq q$, is the potential maximal overlapping length of each test segment $[a_j@DS]$, $a_j \in I\ (1 \leq j \leq q)$;

y_{ji}, $1 \leq j \leq q$, $1 \leq i \leq n$, is the adjustment factor for the potential maximal overlapping length of each test segment $[a_j@DS]$, $a_j \in I\ (1 \leq j \leq q)$ initiating from each state $v_i \in V\ (1 \leq i \leq n)$;

$z_j - y_{ji}$, $1 \leq j \leq q$, $1 \leq i \leq n$, is the l_{op} of each test segment $[a_j@DS]$, $a_j \in I\ (1 \leq j \leq q)$ initiating from each state $v_i \in V\ (1 \leq i \leq n)$;

$z_r=\min\{z_j : j=1,\ldots,q\}$, $1 \leq r \leq q$;

$t= \sum_{v_i \in V;\ \xi^{Ec}(v_i)<0} \min(|\xi^{Ec}(v_i)|, |\{z_j - y_{ji} : z_j - y_{ji} \neq 0, j=1,\ldots,q\}|).$

Proof: There are three possible cases for a given DS.

Case 1: DS is in the format of $a_1a_1\cdots\cdots a_1$, that is, the DS consists of L repeated input symbols (without loss of generality, it can be assumed that the input symbol is a_1).

There are q test segments $[a_j@DS]$ $(a_j \in I, 1 \leq j \leq q)$ initiating from each state $v_i \in V\ (1 \leq i \leq n)$. The potential maximal overlapping length of each test segment $[a_1@DS]$ initiating from each $v_i \in V\ (1 \leq i \leq n)$ is L, i.e.,

z_1=L. On the other hand, each test segment [a_j@DS] ($2 \le j \le q$) initiating from $v_i \in V$ ($1 \le i \le n$) cannot overlap any other test segment, i.e., z_j=0.

Since DS is in format (1), $X_1=\emptyset$, and thus, the adjustment factors for the potential maximal overlapping length for all test segments are 0 (i.e., y_{ji}=0, j=1,...,q, i=1,...,n).

In the best case, each test segment [a_1@DS] can overlap another test segment with its potential maximal overlapping length L, and the first test segment in a test sequence initiating from the initial state corresponds to a transition other than an a_1-transition. Since, there are n test segments [a_1@DS], the maximal saving of length nL on test segments can be obtained, compared with $\text{MRTS}_{\text{OTS\_D}}$, i.e., TSOP_D = nL.

Meanwhile, due to the overlapping, no transfer sequences are required for the test segments [a_1@DS] initiating from $v_i \in V$ and $\xi^{\text{Ec}}(v_i) < 0$ ($1 \le i \le n$). Therefore, a saving of length $t=|\{i : v_i \in V, \xi^{\text{Ec}}(v_i) < 0\}|$ on transfer sequences can be obtained, compared with $\text{MRTR}_{\text{OTS\_D}}$, i.e., TROP_D = t.

Therefore, the length of a test sequence for the OTSO_D problem can not be below

$$(\text{MRTS}_{\text{OTS\_D}} - \text{TSOP}_\text{D})+(\text{MRTR}_{\text{OTS\_D}} - \text{TROP}_\text{D})$$
$$= [nq(1+L) + \tau - \beta] - nL - t, \quad \text{where } t = |\{i : v_i \in V, \xi^{\text{Ec}}(v_i) < 0\}|.$$

Since z_1=L, z_j=0 (j=2,...,q) and y_{ji}=0 (j=1,...,q, i=1,...,n), we get

$$z_r = \min\{z_j : j=1,\ldots,q\} = 0,\ 2 \le r \le q, \text{ and}$$
$$(n \sum_{j=1}^{q} z_j - z_r) - \sum_{j=1}^{q} \sum_{i=1}^{n} y_{ji} = nL.$$

Meanwhile, we get

$|\{z_j - y_{ji} : z_j - y_{ji} \neq 0, j=1,\ldots,q\}| = |\{z_1\}| = 1.$

Thus,

$$t = \sum_{v_i \in V;\ \xi^{E_C}(v_i)<0} \min(|\xi^{E_C}(v_i)|, |\{z_j - y_{ji} : z_j - y_{ji} \neq 0, j=1,\ldots,q\}|)$$
$$= \sum_{v_i \in V;\ \xi^{E_C}(v_i)<0} 1 = |\{i : v_i \in V, \xi^{E_C}(v_i)<0\}|.$$

So, the lower bound is proved in this case.

Case 2: DS is in the format of $X_1@a_1$, where X_1 is the intrinsically nonoverlapping prefix of the DS w.r.t. $a_1@DS$ (without loss of generality, it can be assumed that the last input symbol in the DS is a_1).

There are q test segments $[a_j@X_1@a_1]$ ($a_j \in I$, $1 \leq j \leq q$) initiating from each state $v_i \in V$ ($1 \leq i \leq n$). The potential maximal overlapping length of each test segment $[a_1@X_1@a_1]$ initiating from each $v_i \in V$ ($1 \leq i \leq n$), is 1, i.e., $z_1=1$. On the other hand, each test segment $[a_j@X_1@a_1]$ ($2 \leq j \leq q$) initiating from $v_i \in V$ ($1 \leq i \leq n$) cannot overlap any other test segment, i.e., $z_j=0$ ($2 \leq j \leq q$).

However, if $v_i \in V$ and $d_{in}^{E_{X_1}}(v_i)=0$, then the test segment $[a_1@DS]$ initiating from v_i cannot overlap any other test segment with its potential maximal overlapping length (i.e., 1). Here, E_{X_1} is the set of edges from a new graph $G[E_{X_1}]=(V, E_{X_1})$, where $E_{X_1}=\{(v_j,v_r;X_1(v_j)) : X_1$ is the intrinsically nonoverlapping prefix of the DS w.r.t. $a_1@DS$, $X_1\neq\emptyset$, $Tail(X_1(v_j))=v_r \in V\}$, and $|E_{X_1}|=|V|$. Thus, $v_i \in V$ and $d_{in}^{E_{X_1}}(v_i)=0$ means that there is no path in G corresponding to X_1 that can reach v_i.

Accordingly, the test segment [a_1@DS] initiating from v_i cannot overlap any other test segment a_j@X_1@a_1 with its potential maximal overlapping length (i.e., 1).

Thus, the adjustment factor for the potential maximal overlapping length for each of the test segments [a_1@DS] initiating from $v_i \in V$ and $d_{in}^{EX_1}(v_i)=0$ should be 1, i.e., $y_{1i}=1$ ($i \in \{i : v_i \in V$ and $d_{in}^{EX_1}(v_i)=0\}$). Also, the adjustment factor for the potential maximal overlapping length for each of the test segments [a_1@DS] initiating from $v_i \in V$ and $d_{in}^{EX_1}(v_i)\neq 0$ or for each of the test segments [a_j@DS] ($j=2,\ldots,q$) initiating from each $v_i \in V$ ($i=1,\ldots,n$) should be 0, i.e., $y_{1i}=0$ ($i \in \{i : v_i \in V$ and $d_{in}^{EX_1}(v_i)\neq 0\}$) or $y_{ji}=0$ ($j=2,\ldots,q$, $i=1,\ldots,n$).

In the best case, each test segment [a_1@DS] can overlap another test segment with its potential maximal overlapping length except the test segments [a_1@DS] initiating from $v_i \in V$ and $d_{in}^{EX_1}(v_i)=0$, and the first test segment in a test sequence initiating from the initial state corresponds to a transition other than an a_1-transition. There are n test segments [a_1@DS], among which $p=|\{i : v_i \in V$ and $d_{in}^{EX_1}(v_i)=0\}|$ test segments cannot overlap any other test segment with its potential maximal overlapping length (i.e., 1). So, the maximal saving of length (n - p) on test segments can be obtained, compared with $MRTS_{OTS\_D}$, i.e., $TSOP_D = n - p$.

Meanwhile, due to the overlapping, no transfer sequences are required for the test segments [a_1@DS] initiating from $v_i \in V$, $\xi^{E_C}(v_i) < 0$ and $d_{in}^{EX_1}(v_i)\neq 0$ ($1 \leq i \leq n$). Therefore, a saving of length $t=|\{i : v_i \in V$, $\xi^{E_C}(v_i) < 0$ and $d_{in}^{EX_1}(v_i)\neq 0\}|$ on transfer sequences can be obtained, compared with $MRTR_{OTS\_D}$, i.e., $TROP_D = t$.

Therefore, the length of a test sequence for the OTSO_D problem can not be below

$$(MRTS_{OTS\_D} - TSOP_D) + (MRTR_{OTS\_D} - TROP_D)$$
$$= [nq(1+L) + \tau - \beta] - (n - p) - t,$$

where $p = |\{i : v_i \in V \text{ and } d_{in}^{Ex_1}(v_i)=0\}|$, $t = |\{v_i : v_i \in V, \xi^{Ec}(v_i) < 0 \text{ and } d_{in}^{Ex_1}(v_i) \neq 0\}|$.

Since $z_1=1$, $z_j=0$ $(j=2,\ldots,q)$ and
$y_{1i}=1$ $(i \in \{i : v_i \in V \text{ and } d_{in}^{Ex_1}(v_i)=0\})$,
$y_{1i}=0$ $(i \in \{i : v_i \in V \text{ and } d_{in}^{Ex_1}(v_i)\neq 0\})$,
$y_{ji}=0$ $(j=2,\ldots,q,\ i=1,\ldots,n)$,
we get
$z_r = \min\{z_j : j=1,\ldots,q\}=0$, $2 \leq r \leq q$, and
$(n \sum_{j=1}^{q} z_j - z_r) - \sum_{j=1}^{q} \sum_{i=1}^{n} y_{ji} = n - |\{i : v_i \in V \text{ and } d_{in}^{Ex_1}(v_i) = 0\}| = n - p.$

Meanwhile, we get

$|\{z_j - y_{ji} : z_j - y_{ji} \neq 0, j=1,\ldots,q\}| = |\{z_1 - y_{1i} : z_1 - y_{1i} \neq 0\}| =$
1 when $i \in \{i : v_i \in V \text{ and } d_{in}^{Ex_1}(v_i)\neq 0\}$; and
0 when $i \in \{i : v_i \in V \text{ and } d_{in}^{Ex_1}(v_i)=0\}$.

Thus,

$$t = \sum_{v_i \in V;\ \xi^{Ec}(v_i)<0} \min(|\xi^{Ec}(v_i)|, |\{z_j - y_{ji} : z_j - y_{ji} \neq 0, j=1,\ldots,q\}|)$$
$$= \sum_{v_i \in V;\ \xi^{Ec}(v_i)<0} 1 \text{ when } i \in \{i : v_i \in V \text{ and } d_{in}^{Ex_1}(v_i)\neq 0\}$$
$$= |\{i : v_i \in V, \xi^{Ec}(v_i) < 0 \text{ and } d_{in}^{Ex_1}(v_i)\neq 0\}| = t.$$

So, the lower bound is proved in this case.

Case 3: $D_{k1}@D_{k2}=D_{k3}@D_{k1}=X_k@a_k@D_{k1}$ (k=1,...,h), where $|D_{k2}|=|D_{k3}| > 1$; at least one D_{k1} satisfying $|D_{k1}| \geq 1$; at most one $D_{k1}=\emptyset$; and X_k is the intrinsically nonoverlapping prefix of the DS w.r.t. a_k@DS (without loss of generality, it can be assumed that $k=1,\ldots,h \leq q$, i.e., a_k (k=1,...,h) are the first h input symbols in I, and if there is one $D_{k1}=\emptyset$, then $D_{h1}=\emptyset$). Note that $|D_{k2}|=|D_{k3}| > 1$; otherwise, it should belong to Case 1.

There are q test segments [a_j@DS] ($a_j \in$ I, $1 \leq j \leq q$) initiating from each state $v_i \in$ V ($1 \leq i \leq n$). The potential maximal overlapping length of each test segment [$a_k@D_{k1}@D_{k2}$] ($1 \leq k \leq h$) initiating from each $v_i \in$ V ($1 \leq i \leq n$), is $|a_k@D_{k1}|$, i.e., $z_k=|D_{k1}|+1$ ($1 \leq k \leq h$). On the other hand, each test segment [a_j@DS] ($h+1 \leq j \leq q$) initiating from $v_i \in$ V ($1 \leq i \leq n$) cannot overlap any other test segment, i.e., $z_j= 0$ ($h+1 \leq j \leq q$).

However, if $v_i \in$ V and $d_{in}^{E_{X_k}}(v_i)=0$ ($1 \leq k \leq h$), then the test segment [a_k@DS] initiating from v_i cannot overlap any other test segment with its potential maximal overlapping length $|D_{k1}|+1$ (i.e., $|Z_k|$). Here, E_{X_k} is the set of edges from a new graph $G[E_{X_k}]=(V, E_{X_k})$, where $E_{X_k}=\{(v_j,v_r;X_k(v_j)) : X_k$ is the intrinsically nonoverlapping prefix of the DS w.r.t. a_k@DS, $X_k \neq \emptyset$, $Tail(X_k(v_j))=v_r \in V\}$, and $|E_{X_k}|=|V|$. Thus, $v_i \in$ V and $d_{in}^{E_{X_k}}(v_i)=0$ means that there is no path in G corresponding to X_k that can reach v_i. Accordingly, the test segment [a_k@DS] initiating from v_i cannot overlap any other test segment [$a_j@X_k@a_k@D_{k1}$] with its potential maximal overlapping length $|D_{k1}|+1$ (i.e., $|Z_k|$).

Therefore, the adjustment factor for the potential maximal overlapping length for each of the test segments [a_k@DS] ($1 \leq k \leq h$)

initiating from $v_i \in V$ and $d_{in}^{Ex_k}(v_i)=0$ should not be 0, i.e., $y_{ki} \neq 0$ $(1 \leq k \leq h,\ i \in \{i : v_i \in V \text{ and } d_{in}^{Ex_k}(v_i)=0\})$. Also, the adjustment factor for the potential maximal overlapping length for each of the test segments $[a_k@DS]$ $(1 \leq k \leq h)$ initiating from $v_i \in V$ and $d_{in}^{Ex_k}(v_i) \neq 0$ or for each of the test segments $[a_j@DS]$ $(h+1 \leq j \leq q)$ initiating from each $v_i \in V$ $(i=1,\ldots,n)$ should be 0, i.e., $y_{ki}=0$ $(1 \leq k \leq h,\ i \in \{i : v_i \in V \text{ and } d_{in}^{Ex_k}(v_i) \neq 0\})$ or $y_{ji}=0$ $(j=h+1,\ldots,q,\ i=1,\ldots,n)$.

For the same reason, if $v_i \in V$ and $d_{in}^{Ex_{ku}}(v_i)=0$ $(1 \leq k \leq h,$ $2 \leq u \leq b_k$, b_k is the number of the potential possible overlapping suffixes of the DS w.r.t. $a_k@DS)$, then the test segment $[a_k@DS]$ initiating from v_i cannot overlap any other test segment with the potential overlapping length $|Z_{ku}|$ of the DS w.r.t. $a_k@DS$. Here, $Ex_{ku}(1 \leq k \leq h,\ 2 \leq u \leq b_k)$ is the set of edges from a new graph $G[Ex_{ku}]=(V, Ex_{ku})$, where $Ex_{ku}=\{(v_j,v_r;X_{ku}(v_j)) : X_{ku}=DS-Z_{ku}$, Z_{ku} is the potential overlapping suffix of the DS w.r.t. $a_k@DS$, $Tail(X_{ku}(v_k))=v_r \in V\}$, and $|Ex_{ku}|=|V|$.

Thus, the adjustment factor for the potential maximal overlapping length for each of the test segments $[a_k@DS]$ $(1 \leq k \leq h)$ initiating from $v_i \in V$ and $d_{in}^{Ex_{ku}}(v_i) \neq 0$ $(2 \leq u \leq b_k)$, but $d_{in}^{Ex_k}(v_i)=0$ and $d_{in}^{Ex_{ks}}(v_i)=0$ $(s < u)$ should be the difference between the potential overlapping length $|Z_{ku}|$ of the DS w.r.t. $a_k@DS$ and its potential maximal overlapping length $|Z_k|$, i.e., $y_{ki} = |Z_k| - |Z_{ku}|$ $(1 \leq k \leq h,\ i \in \{i : v_i \in V \text{ and } d_{in}^{Ex_{ku}}(v_i) \neq 0,$ $(2 \leq u \leq b_k)$, but $d_{in}^{Ex_k}(v_i)=0$ and $d_{in}^{Ex_{ks}}(v_i)=0$ $(s < u)\})$.

Moreover, the adjustment factor for the potential maximal overlapping length for each of the test segments $[a_k@DS]$ $(1 \leq k \leq h)$ initiating from $v_i \in V$ and $d_{in}^{Ex_k}(v_i)=0$, $d_{in}^{Ex_{ks}}(v_i)=0$ $(2 \leq s \leq b_k)$ should be its potential maximal overlapping length $|Z_k|$, i.e., $y_{ki} = |Z_k|$ ($1 \leq k \leq h$, $i \in \{i : v_i \in V \text{ and } d_{in}^{Ex_k}(v_i)=0, d_{in}^{Ex_{ks}}(v_i)=0 \ (2 \leq s \leq b_k)\}$).

In the best case, each test segment with $z_k \neq 0$ (i.e., each test segment [a_k@DS]) can overlap another test segment with its potential maximal overlapping length (i.e., $|a_k@D_{k1}|$), with the following exceptions: a) the first test segment in a test sequence initiating from the initial state cannot overlap any other test segment, b) test segments initiating from $v_i \in V$ and $d_{in}^{Ex_k}(v_i)=0$ ($1 \leq k \leq h$) cannot overlap any other test segment with their potential maximal overlapping length.

So, the maximal saving on test segments compared with $MRTS_{OTS\_D}$ can be represented by the difference between the sum of the potential maximal overlapping length z_j for each test segment except the first test segment in the test sequence initiating from the initial state and the sum of the adjustment factor for the potential maximal overlapping length y_{ji} for those test segments initiating from $v_i \in V$ and $d_{in}^{Ex_k}(v_i)=0$ ($1 \leq k \leq h$), i.e., $TSOP_D = (n \sum_{j=1}^{q} z_j - z_r) - \sum_{j=1}^{q} \sum_{i=1}^{n} y_{ji}$, where z_r is the potential maximal overlapping length for the first test segment initiating from the initial state.

It is clear that the first test segment should be one of those test segments with the minimum of z_j ($j=1,\ldots,q$), so that $TSOP_D$ can be maximized in the best case. So, $z_r = \min\{z_j : j=1,\ldots,q\}$, $1 \leq r \leq q$. When $h < q$, the first test segment in a test sequence initiating from the initial state should correspond to a transition other than an a_k-transition (i.e., $z_r=0$, $h+1 \leq r \leq q$).

Obviously, a test segment initiating from $v_i \in V$ with $z_j - y_{ji} \neq 0$ indicates that the test segment can overlap another test segment in the best case. Thus, due to the overlapping, no transfer sequences are required for these test segments [a_j@DS] ($1 \leq j \leq q$) initiating from $v_i \in V$ and $\xi^{E_C}(v_i) < 0$ with $z_j - y_{ji} \neq 0$ ($1 \leq i \leq n$). The total number of test segments initiating from $v_i \in V$ that can overlap other test segments is $|\{z_j - y_{ji} : z_j - y_{ji} \neq 0, j=1,\ldots,q\}|$. So, for each $v_i \in V$ and $\xi^{E_C}(v_i) < 0$, the total length of transfer sequences that may be saved compared with $MRTR_{OTS\_D}$, cannot be larger than the minimum of $|\xi^{E_C}(v_i)|$ and $|\{z_j - y_{ji} : z_j - y_{ji} \neq 0, j=1,\ldots,q\}|$. Therefore, a saving of length $t = \sum_{v_i \in V;\, \xi^{E_C}(v_i)<0} \min(|\xi^{E_C}(v_i)|, |\{z_j - y_{ji} : z_j - y_{ji} \neq 0, j=1,\ldots,q\}|)$ on transfer sequences can be obtained, compared with $MRTR_{OTS\_D}$, i.e., $TROP_D = t$.

Therefore, the length of a test sequence for the OTSO_D problem can not be below

$$(MRTS_{OTS\_D} - TSOP_D) + (MRTR_{OTS\_D} - TROP_D)$$

$$= [nq(1+L) + \tau - \beta] - [(n \sum_{j=1}^{q} z_j - z_r) - \sum_{j=1}^{q} \sum_{i=1}^{n} y_{ji}] - t.$$

◆

The lower bound given in Theorem 2 is LB_{OTSO_D}, in which $TROP_D = t$, representing the observations described in 2), and $TSOP_D = [(n \sum_{j=1}^{q} z_j - z_r) - \sum_{j=1}^{q} \sum_{i=1}^{n} y_{ji}]$, representing the observations described in 1). When a DS is in the format of (1) or (2), LB_{OTSO_D} can be simplified as the following corollaries.

Corollary 1: Given a strongly-connected digraph $G=(V, E)$ representing a deterministic, minimal, and completely-specified FSM $M=(S, I, O, \delta, \lambda)$, $|S|=n$, $|I|=q$, with a DS of length L, in the format (1). The length of a test sequence for the OTSO_D problem can not be below

$$[nq(1+L) + \tau - \beta] - nL - t$$

where

$\tau=\sum |\xi^{Ec}(v_i)|, v_i \in V, \xi^{Ec}(v_i) < 0;$

$\beta=1, \text{ if } \xi^{Ec}(v_1) < 0;$

$\beta=0, \text{ if } \xi^{Ec}(v_1) \geq 0;$

$t=|\{i : v_i \in V, \xi^{Ec}(v_i) < 0\}|.$

Corollary 2: Given a strongly-connected digraph $G=(V, E)$ representing a deterministic, minimal, and completely-specified FSM $M=(S, I, O, \delta, \lambda)$, $|S|=n$, $|I|=q$, with a DS of length L, in the format (2). The length of a test sequence for the OTSO_D problem can not be below

$$[nq(1+L) + \tau - \beta] - (n - p) - t$$

where

$\tau=\sum |\xi^{Ec}(v_i)|, v_i \in V, \xi^{Ec}(v_i) < 0;$

$\beta=1, \text{ if } \xi^{Ec}(v_1) < 0;$

$\beta=0, \text{ if } \xi^{Ec}(v_1) \geq 0;$

$p=|\{i : v_i \in V \text{ and } d_{in}^{Ex_1}(v_i)=0\}|;$

$t=|\{i : v_i \in V, \xi^{E_C}(v_i) < 0, \text{ and } d_{in}^{E_{X_1}}(v_i)\neq 0\}$.

and

$G[E_{X_1}]=(V, E_{X_1})$,

$E_{X_1}=\{(v_j,v_r;X_1(v_j)) : X_1$ is the intrinsically nonoverlapping prefix of the DS w.r.t. a_1@DS, $X_1\neq\emptyset$, $\text{Tail}(X_1(v_j))=v_r \in V\}$, and $|E_{X_1}|=|V|$;

Example:

A DS for the FSM M1 shown in Figure 1 is *aaa* or *aab*. We have L=3, n=6, q=2.

(1) DS=*aaa*

When DS=*aaa* is used, $G[E_C]$ of M1 is shown in Figure 2, where $\xi^{E_C}(v_1)=\xi^{E_C}(v_3)=2$, $\xi^{E_C}(v_2)=\xi^{E_C}(v_5)=0$, $\xi^{E_C}(v_4)=\xi^{E_C}(v_6)= -2$. Table 1 lists all the test segments, i.e., the edges in $G[E_C]$.

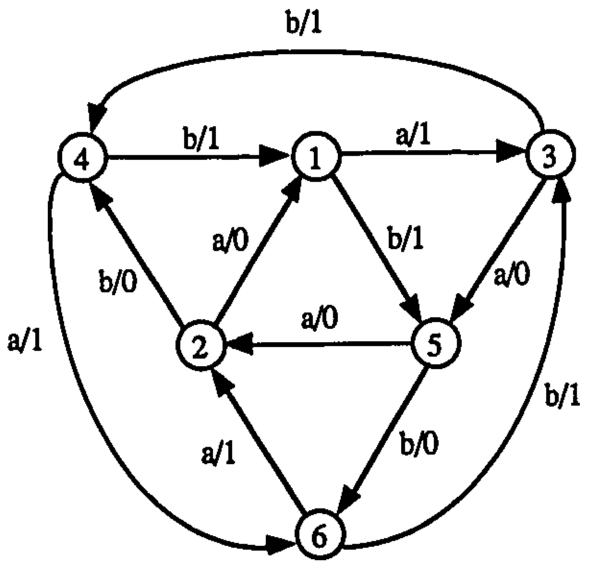


Figure 1 FSM M1

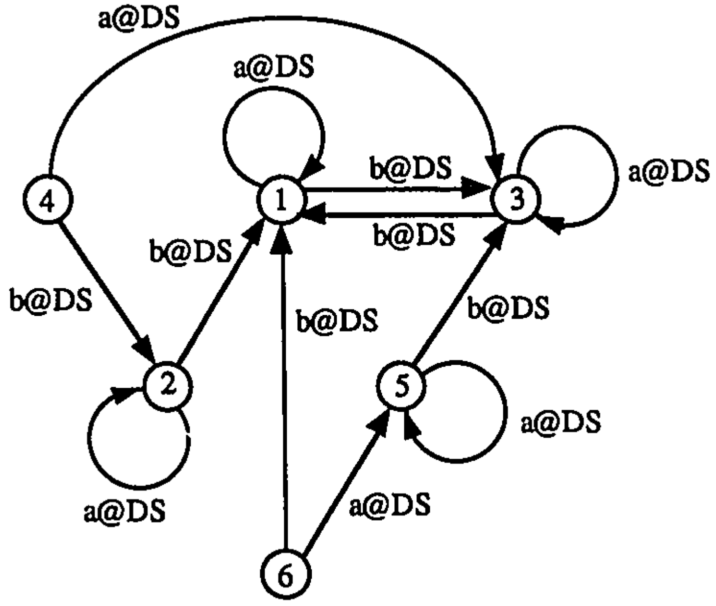


Figure 2 $G[E_C]$ for M1 when DS=*aaa*
(Only input symbols are shown)

S1 (1,3;a/1)(3,5;a/0)(5,2;a/0)(2,1;a/0) = (1,1;a/1@DS(3))

S2 (1,5;b/1)(5,2;a/0)(2,1;a/0)(1,3;a/1) = (1,3;b/1@DS(5))

S3 (2,1;a/0)(1,3;a/1)(3,5;a/0)(5,2;a/0) = (2,2;a/0@DS(1))

S4 (2,4;b/0)(4,6;a/1)(6,2;a/1)(2,1;a/0) = (2,1;b/0@DS(4))

S5 (3,5;a/0)(5,2;a/0)(2,1;a/0)(1,3;a/1) = (3,3;a/0@DS(5))

S6 (3,4;b/1)(4,6;a/1)(6,2;a/1)(2,1;a/0) = (3,1;b/1@DS(4))

S7 (4,6;a/1)(6,2;a/1)(2,1;a/0)(1,3;a/1) = (4,3;a/1@DS(6))

S8 (4,1;b/1)(1,3;a/1)(3,5;a/0)(5,2;a/0) = (4,2;b/1@DS(1))

S9 (5,2;a/0)(2,1;a/0)(1,3;a/1)(3,5;a/0) = (5,5;a/0@DS(2))

S10 (5,6;b/0)(6,2;a/1)(2,1;a/0)(1,3;a/1) = (5,3;b/0@DS(6))

S11 (6,2;a/1)(2,1;a/0)(1,3;a/1)(3,5;a/0) = (6,5;a/1@DS(2))

S12 (6,3;b/1)(3,5;a/0)(5,2;a/0)(2,1;a/0) = (6,1;b/1@DS(3))

Table 1 Test Segments for M1 when DS= *aaa*

From Theorem 1(b), we have $\tau=|\xi^{Ec}(v_4)|+|\xi^{Ec}(v_6)|=4$ since $\xi^{Ec}(v_4)=\xi^{Ec}(v_6)= -2 < 0$ and $\beta=0$ since $\xi^{Ec}(v_1)=2 > 0$. So, the lower bound of the length of a test sequence for the OTS_D problem $LB_{OTS\_D} = nq(1+L) + \tau - \beta = 52$.

Since DS=*aaa* is in format (1), from Corollary 1, we have $t=|\{i : v_i \in V, \xi^{Ec}(v_i) < 0\}|=2$ since $\xi^{Ec}(v_4)=\xi^{Ec}(v_6)= -2$ and $X=\emptyset$. Thus, the lower bound of the length of a test sequence for the OTSO_D problem $LB_{OTSO\_D} = LB_{OTS\_D} - TSOP_D - TROP_D = LB_{OTS\_D} - nL - t = 32$.

(2) DS=*aab*

When DS=*aab* is used, G[E_C] and G[E_X] of M1 is shown in Figure 3 and Figure 4, respectively. We have $\xi^{Ec}(v_1)=\xi^{Ec}(v_2)=\xi^{Ec}(v_3)= -2$, $\xi^{Ec}(v_4)=4$, $\xi^{Ec}(v_5)=2$, $\xi^{Ec}(v_6)=0$ in G[E_C], and $d_{in}^{Ex}(v_1)=d_{in}^{Ex}(v_2)=2$, $d_{in}^{Ex}(v_3)=d_{in}^{Ex}(v_5)=1$, $d_{in}^{Ex}(v_4)= d_{in}^{Ex}(v_6)=0$ in G[E_X]. Table 2 lists all the test segments, i.e., the edges in G[E_C].

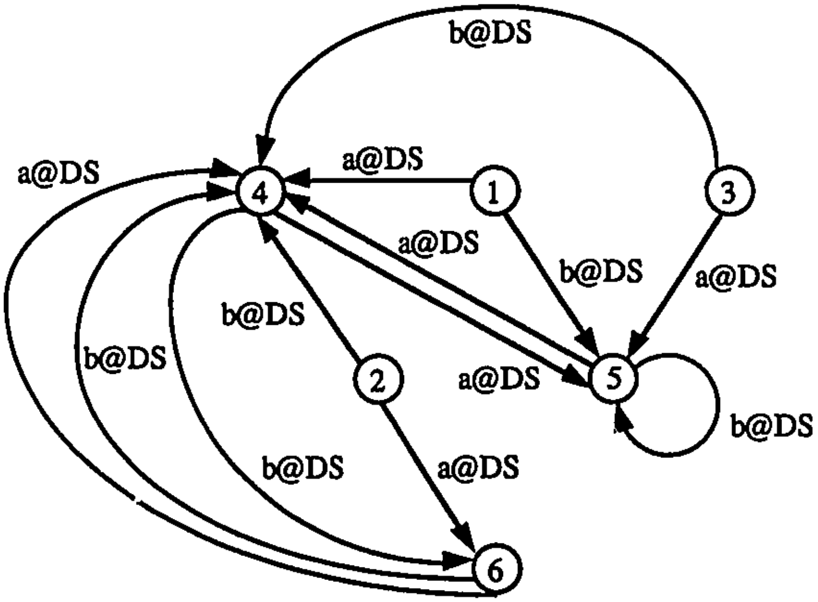


Figure 3 G[E_C] for M1 when DS=*aab*

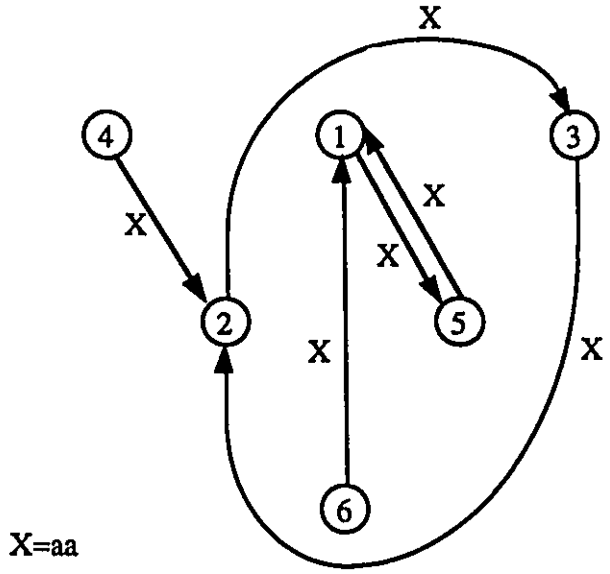


Figure 4 G[E_X] for M1 when DS=*aab*

S1	(1,3;a/1)(3,5;a/0)(5,2;a/0)(2,4;b/0) = (1,4;a/1@DS(3))
S2	(1,5;b/1)(5,2;a/0)(2,1;a/0)(1,5;b/1) = (1,5;b/1@DS(5))
S3	(2,1;a/0)(1,3;a/1)(3,5;a/0)(5,6;b/0) = (2,6;a/0@DS(1))
S4	(2,4;b/0)(4,6;a/1)(6,2;a/1)(2,4;b/0) = (2,4;b/0@DS(4))
S5	(3,5;a/0)(5,2;a/0)(2,1;a/0)(1,5;b/1) = (3,5; a/0@DS(5))
S6	(3,4;b/1)(4,6;a/1)(6,2;a/1)(2,4;b/0) = (3,4;b/1@DS(4))
S7	(4,6;a/1)(6,2;a/1)(2,1;a/0)(1,5;b/1) = (4,5;a/1@DS(6))
S8	(4,1;b/1)(1,3;a/1)(3,5;a/0)(5,6;b/0) = (4,6;b/1@DS(1))
S9	(5,2;a/0)(2,1;a/0)(1,3;a/1)(3,4;b/1) = (5,4;a/0@DS(2))
S10	(5,6;b/0)(6,2;a/1)(2,1;a/0)(1,5;b/1) = (5,5;b/0@DS(6))
S11	(6,2;a/1)(2,1;a/0)(1,3;a/1)(3,4;b/1) = (6,4;a/1@DS(2))
S12	(6,3;b/1)(3,5;a/0)(5,2;a/0)(2,4;b/0) = (6,4;b/1@DS(3))

Table 2 Test Segments for M1 when DS=*aab*

From Theorem 1(b), we have $\tau=|\xi^{Ec}(v_1)|+|\xi^{Ec}(v_2)|+|\xi^{Ec}(v_3)|=6$ since $\xi^{Ec}(v_1)=\xi^{Ec}(v_2)=\xi^{Ec}(v_3)= -2 < 0$, and $\beta=1$ since $\xi^{Ec}(v_1)= -2 < 0$. So, the lower bound of the length of a test sequence for the OTS_D problem $LB_{OTS\_D} = nq(1+L) + \tau - \beta = 53$.

Since DS=*aab* is in format (2), from Corollary 2, we have p=2 since $d_{in}^{Ex}(v_4)=d_{in}^{Ex}(v_6)=0$, and t=3 (since $\xi^{Ec}(v_1)=\xi^{Ec}(v_2)=\xi^{Ec}(v_3)= -2 < 0$ and $d_{in}^{Ex}(v_1)= d_{in}^{Ex}(v_2)=2\neq0$, $d_{in}^{Ex}(v_3)=1\neq0$). Thus, the lower bound of the length of a test sequence for the OTSO_D problem $LB_{OTSO\_D} = LB_{OTS\_D} - TSOP_D - TROP_D = LB_{OTS\_D} - (n - p) - t = 46$.

3.2 Algorithm-D1

As indicated in section 2.4.3, the OTSO problem can be solved as a maximum-cardinality minimum-cost matching problem in a bipartite graph. If the solution is a tour, then it is a valid and shortest test sequence. Otherwise, it is possible that only a near optimum test sequence can be obtained.

Algorithm-D1 combines this approach with the DS, which is described as follows.

Algorithm-D1:

1. Given a strongly connected digraph G=(V, E) representing a deterministic, minimal, and completely-specified FSM M which possesses a DS.

2. Construct a test segment for each transition by concatenating the transition with the DS.

   Denote each test segment by S_i, $i=1,2,\cdots\cdots,nq$.

3. Construct a bipartite graph $G_b=(V_b, E_b)$ where

$V_b=V_t \cup V_h$

$V_t=\{v_1\} \cup \{v_i : v_i=\text{Tail}(S_k) \text{ for } k=1,2,\cdots\cdots,nq\}$

$V_h=\{v_1\} \cup \{v_j : v_j=\text{Head}(S_k) \text{ for } k=1,2,\cdots\cdots,nq\}$

(Here, the inclusion of the initial state guarantees that the test sequence starts and ends at the initial state.)

$E_b=\{(v_i,v_j) : v_i \in V_t, v_j \in V_h; v_i \neq \text{Tail}(S_k), v_j \neq \text{Head}(S_k) \text{ for some } k\}$

(Note that each edge in E_b represents a connection from the tail of one test segment to the head of another test segment. Thus, for a test segment S_k, the edge from Tail(S_k) to Head(S_k) should not be included in E_b.)

Assign cost C_{ij} to edge $(v_i,v_j) \in E_b$ according to the following rules:

(a) Let (v_i,v_j)=(Tail(S_x), Head(S_y)). If S_y can overlap S_x, then a length equal to the negative of the overlapping length is assigned to edge (v_i,v_j) as its cost.

(b) Otherwise, the length of the shortest path from v_i to v_j in G is assigned to edge (v_i,v_j) as its cost.

(c) The length of the shortest path from v_1 to v_i and from v_i to v_1 in G is assigned to (v_1,v_i) and (v_i,v_1), respectively, as their costs.

4. Use a maximum-cardinality minimum-cost matching algorithm for graph G_b.

The solution to the above matching problem is an alternating sequence of test segments, overlapped test segment groups and transfer sequences connecting them. If the solution is a tour, it is a valid test sequence. Otherwise, we may have to connect the disconnected components using, for example, a greedy heuristic, which is very similar to the algorithm for constructing a directed spanning graph or arborescence [PAP82].

Algorithm-D1 described above applies overlapping approach for reducing the length of test sequences, which uses an idea similar to that in the algorithm proposed in [CHE90] which employs UIO sequences as SIV sequences. The time complexity of Algorithm-D1 is of order $|E|^2$ which is the same as that discussed in [CHE90].

3.3 An Example Using Algorithm-D1

For the machine M1 shown in Figure 1 and DS=*aaa*, Table 3 gives a solution obtained by the maximum-cardinality minimum-cost matching algorithm applied to G_b. In Table 3, row i contains the cost C_{ij} of the path from Tail(S_i) to Head(S_j) if segment S_i is followed immediately by segment S_j. So, if C_{ij} is circled in a solution, then S_i is followed immediately by S_j in the solution. Only one cost in each row or column of the table is circled to form a solution.

Based on the solution shown in Table 3 and the test segments listed in Table 1, the test sequence generated by Algorithm-D1 is shown in Table 4. For example, from the solution, we know that S_2 is followed by S_9 with a cost -3 between them. Accordingly, in the first two lines of Table 4, S_2 is followed by S_9 with an overlapping part (5,2;a/0)(2,1;a/0)(1,3;a/0). Also, from the solution, we know that S_9 is followed by S_{12} with a cost 1 between them. Thus, there is a transfer sequence (5,6;b/0) shown in the third line of Table 4, which is between S_9 and S_{12}.

S_i \ S_j	1	2	3	4	5	6	7	8	9	10	11	12
1	-	0	-1	2	◯-3	1	2	2	-2	1	2	2
2	-1	-	-2	2	0	0	1	1	◯-3	1	2	2
3	◯-3	1	-	0	-2	2	1	1	-1	2	2	2
4	0	◯0	-1	-	1	1	-3	2	1	1	-2	2
5	-1	2	-2	2	-	0	1	◯1	-3	1	2	2
6	0	0	-1	2	1	-	◯-3	2	1	1	-2	2
7	-1	2	-2	2	0	0	-	1	1	1	◯-3	2
8	-3	1	0	◯0	-2	2	1	-	-1	2	2	2
9	-2	2	-3	1	-1	2	2	2	-	0	1	◯1
10	-1	2	-2	2	0	◯0	1	1	1	-	-3	2
11	-2	2	◯-3	1	-1	2	2	2	0	0	-	1
12	0	0	-1	2	-3	1	2	2	-2	◯1	2	-

↑ C_{i1} between S_i and S_1 ◯ S_i --> S_j is chosen in the solution

Solution: S_2-->S_9-->S_{12}-->S_{10}-->S_6-->S_7-->S_{11}-->S_3-->S_1-->S_5-->S_8-->S_4-->S_2 (cost=33)

Table 3 A Solution Obtained by the Maximum-Cardinality Minimum-Cost Matching Algorithm for M1 when DS=*aaa*

The length of the test sequence is 33, which approaches the lower bound 32 except that one more transfer sequence of length 1 (i.e., the transfer sequence (1,5;b)) than the estimation for the lower bound is used.

Transition	Test Segment
(1,5;b/1)	(1,5;b/1)(5,2;a/0)(2,1;a/0)(1,3;a/1)
(5,2;a/0)	(5,2;a/0)(2,1;a/0)(1,3;a/1)(3,5;a/0)
	(5,6;b/0)
(6,3;b/1)	(6,3;b/1)(3,5;a/0)(5,2;a/0)(2,1;a/0)
	(1,5;b/1)
(5,6;b/0)	(5,6;b/0)(6,2;a/1)(2,1;a/0)(1,3;a/1)
(3,4;b/1)	(3,4;b/1)(4,6;a/1)(6,2;a/1)(2,1;a/0)
(4,6;a/1)	(4,6;a/1)(6,2;a/1)(2,1;a/0)(1,3;a/1)
(6,2;a/1)	(6,2;a/1)(2,1;a/0)(1,3;a/1)(3,5;a/0)
(2,1;a/0)	(2,1;a/0)(1,3;a/1)(3,5;a/0)(5,2;a/0)
(1,3;a/1)	(1,3;a/1)(3,5;a/0)(5,2;a/0)(2,1;a/0)
(3,5;a/0)	(3,5;a/0)(5,2;a/0)(2,1;a/0)(1,3;a/1)
	(3,4;b/1)
(4,1;b/1)	(4,1;b/1)(1,3;a/1)(3,5;a/0)(5,2;a/0)
(2,4;b/0)	(2,4;b/0)(4,6;a/1)(6,2;a/1)(2,1;a/0)

Test Sequence = b/1, a/0, a/0, a/1, a/0, **b/0**, b/1, a/0, a/0, a/0, **b/1**,
b/0, a/1, a/0, a/1, b/1, a/1, a/1, a/0, a/1, a/0, a/0,
a/0, a/1, **b/1**, b/1, a/1, a/0, a/0, b/0, a/1, a/1, a/0.
(Transfer sequences are bold-faced in the test sequence.)

Test Sequence Length = 33

Table 4 A Test Sequence Generated by Algorithm-D1 for M1 when DS=*aaa*

3.4 Algorithm-D2

One problem in Algorithm-D1 is that the solution of the maximum-cardinality minimum-cost matching algorithm may not be a tour even though there is such a solution that forms a tour. For example, both solutions in Table 3 and Table 5 have the same cost 33, but the latter does not form a tour. This arbitrariness resulting from the maximum-cardinality minimum-cost matching algorithm can be solved by an adjustment procedure so that the final solution will be a tour if there is one.

S_i \ S_j	1	2	3	4	5	6	7	8	9	10	11	12
1	-	0	-1	2	(-3)	1	2	2	-2	1	2	2
2	-1	-	-2	2	0	0	1	1	(-3)	1	2	2
3	-3	1	-	(0)	-2	2	1	1	-1	2	2	2
4	0	(0)	-1	-	1	1	-3	2	1	1	-2	2
5	-1	2	-2	2	-	0	1	(1)	-3	1	2	2
6	0	0	-1	2	1	-	(-3)	2	1	1	-2	2
7	-1	2	-2	2	0	(0)	-	1	1	1	-3	2
8	(-3)	1	0	0	-2	2	1	-	-1	2	2	2
9	-2	2	(-3)	1	-1	2	2	2	-	0	1	1
10	-1	2	-2	2	0	0	1	1	1	-	(-3)	2
11	-2	2	-3	1	-1	2	2	2	0	0	-	(1)
12	0	0	-1	2	-3	1	2	2	-2	(1)	2	-

C_{i1} between S_i and S_1

○ S_i --> S_j is chosen in the solution

Solution: S_2-->S_9-->S_3-->S_4-->S_2, S_{12}-->S_{10}-->S_{11}-->S_{12},
S_6-->S_7-->S_6, S_1-->S_5-->S_8-->S_1.
(cost=33)

Table 5 Another Solution Obtained by the Maximum-Cardinality Minimum-Cost Matching Algorithm for M1 when DS=*aaa*

Another problem in Algorithm-D1 is that if no tour can be found based on the solution of the maximum-cardinality minimum-cost matching algorithm, and the number of the disconnected components is considerably large, the generated test sequence will be far from the optimal one. If we assume that only the solution in Table 5 is available, then at least three more transitions will be added to connect those disconnected components, which represents an increase of almost 10% in the length of the test sequence.

To avoid these problems, a different approach should be used which reduces the OTSO problem to the RCPP and applies the Euler tour algorithm instead of the matching algorithm. For describing this approach, following terms are defined:

> A *maximum overlapping test segment group* (MOG) is a sequence of test segments which achieve the potential maximal overlapping length between consecutive ones. A *maximum-length MOG* (MMOG) is an MOG that can not be contained in any other MOG.
>
> If the first test segment of an MOG can overlap another test segment, we say that the MOG can *overlap* another test segment. Also, the potential maximal overlapping length (or part) of the first test segment of an MOG is said to be the *potential maximal overlapping length* (or *part*) of the MOG.

It is assumed that the given DS is in one of the following three formats: (1) $a_1a_1......a_1$ ($X_1=\emptyset$), (2) $X_1@a_1$, (3) $X_1@a_1@D_{11}=D_{11}D_{12}$ (X_1 is the intrinsically nonoverlapping prefix of the DS w.r.t. $a_1@DS$, $a_1@D_{11}$ is the only potential overlapping suffix of the DS w.r.t. $a_1@DS$). In all cases, only those test segments corresponding to an a_1-transitions can overlap other test segments.

Thus, in this approach, the OTSO_D problem is solved in two levels:

Level 1: for n test segments [$a_1@DS$] initiating from n states, find MOG's or MMOG's within these n test segments (called the *local optimization problem* of the OTSO_D problem);

Level 2: transform every MOG obtained in Level 1 into an MMOG by finding a test segment to be overlapped by the MOG with its potential maximal overlapping length; and combine all MMOG's and all the remaining test segments optimally (called the *global optimization problem* of the OTSO_D problem).

The main feature of this approach is that not all test segments are considered at once.

When $X_1=\emptyset$, the local optimization problem stated in Level 1 can be solved by constructing a minimum number of paths in G which cover every a_1-transition exactly once. This can be done by an efficient algorithm similar to that in [GON70]. Then, each of these paths corresponding to a_1-transitions can be transformed into an MOG by appending a path corresponding to the DS to the end of the path, since every test segment $[a_1@DS]$ except the first one in the extended path overlap its previous one with its potential maximal overlapping length $|DS|$.

When $X_1\neq\emptyset$, to solve the local optimization problem stated in Level 1, a new duplex graph $G_d=(V_d, E_d)$ is constructed from a strongly connected digraph $G=(V, E)$ representing a deterministic, minimal and completely-specified FSM M as follows:

$V_d = V \cup U'$ where $U'=\{u_i : v_i \in V\}$;

$E_d = E_{d1} \cup E_{d2}$ where

E_{d1} each edge in E_{d1} connects a vertex in V to a vertex in U' which corresponds to the a_1-transition of a test segment $[a_1@DS]$;

E_{d2} each edge in E_{d2} connects a vertex in U' to a vertex in V which corresponds to the X_1 of a test segment $[a_1@DS]$.

Obviously, traversing an edge in E_{d1} will be followed by traversing an edge in E_{d2}, and *vice versa*. Then, a minimum number of paths in G_d, which cover every edge in E_{d1} (i.e., a_1-transition) exactly once, can be constructed with the restriction that each path should start with an edge in E_{d1} and also end with an edge in E_{d1}. This can be done by an efficient algorithm similar to that in [GON70]. Moreover, each of these paths can be transformed into an MOG by appending a path corresponding to the DS to the end of the path, since each test segment [a_1@DS] except the first test segment in the extended path can overlap its previous one with its potential maximal overlapping length. Such an MOG may already be an MMOG if there is no path corresponding to X_1 which can reach the head of the MOG.

To solve the components of the global optimization problem stated in Level 2, i.e., how to transform every MOG obtained in Level 1 into an MMOG by finding a test segment corresponding to a transition other than an a_1-transition to be overlapped by the MOG with its potential maximal overlapping length, and how to combine all MMOG's and all remaining test segments optimally, a new digraph G'=(V', E') is constructed from a strongly connected digraph G=(V, E) representing a deterministic, minimal and completely-specified FSM M as follows:

$V' = V \cup U$ where

U each vertex in U corresponds to the head of an MOG obtained in Level 1;

$E' = E \cup E_U \cup E_g \cup E_C'$ where

$E_U = E_h \cup E_p \cup E_t$,

E_h - each edge in E_h connects a vertex in V to a vertex in U, which corresponds to the prefix of each test segment that can be overlapped by an MOG with its potential maximal overlapping length.. This prefix of the test segment is chosen in such a way that it can be appended in front of the MOG to transform the MOG into an MMOG;

E_p - each edge in E_p connects a vertex in U to a vertex in V, which corresponds to an MOG obtained in Level 1;

It can be noted that traversing a pair of edges from E_h and E_p, respectively, will force the first test segment [a_1@DS] in an MOG to overlap a test segment [a_j@DS] ($1 \leq j \leq q$) with its potential maximal overlapping length. In other words, this pair of edges becomes an MMOG.

E_t - each edge in E_t connects a vertex in U to a vertex in V, which corresponds to the suffix of each test segment that can be overlapped by an MOG with its potential maximal overlapping length. The suffix of the test segment corresponds to the potential maximal overlapping part of the MOG.

It can be noted that a test segment will be covered completely if a pair of edges from E_h and E_t, respectively, are traversed. One of these test segments which can be overlapped by the same MOG with its potential maximal overlapping length, is not considered in E_t, since one of them will be covered completely through traversing a pair of edges from E_h and E_p, respectively;

E_g each edge in E_g corresponds to an MMOG obtained in Level 1;

E_C' each edge in E_C' corresponds to one of the remaining test segments that are not covered by any edges in E_h, E_p, E_t, and E_g.

Obviously, traversing one edge in E_h will be followed by traversing one edge in E_p or E_t. We know that such a pair of edges corresponds to an MMOG which is generated from an MOG obtained in Level 1, or a test segment that can be overlapped by an MOG with its potential maximal overlapping length. Therefore, the components of the global optimization problem mentioned in Level 2 can be solved simultaneously by finding an RCPT of G' over $E_U \cup E_g \cup E_C'$. In other words, the problem of finding an optimal test sequence is reduced to finding an RCPT of G' over $E_U \cup E_g \cup E_C'$. As stated earlier, there are efficient algorithms to solve this problem if $G[E_U \cup E_g \cup E_C']$ is a weakly-connected spanning subgraph of G'.

Algorithm-D2 which uses the approach discussed above is given as follows:

Algorithm-D2

1. Given a strongly connected digraph G=(V, E) representing a deterministic, minimal, and completely-specified FSM M which possesses a DS in one of the following formats: (1) $a_1a_1......a_1$ (X_1=Ø), (2) $X_1@a_1$, (3) $X_1@a_1@D_{11}=D_{11}D_{12}$ (X_1 is the intrinsically nonoverlapping prefix of the DS w.r.t. a_1@DS, $a_1@D_{11}$ is the only potential overlapping suffix of the DS w.r.t. a_1@DS).

2. Construct a test segment for each transition by concatenating the transition with the DS.

3. Compute all MOG's or MMOG's.

   Case A: DS is in the format of $a_1a_1......a_1$.

   (i) Find the minimum number of paths which consist of only a_1-transitions and cover all a_1-transitions in G (denote this minimum number by δ).

   (ii) Append a path corresponding to the DS to the end of each path, which forms the MOG's required, which is denoted by P_{mi} ($1 \leq i \leq \delta$).

   Case B: DS is in the format of $X_1@a_1$ or $X_1@a_1@D_{11}=D_{11}D_{12}$.

   (i) Construct a new duplex graph $G_d=(V_d, E_d)$

where $V_d=V \cup U'$,

$U'=\{u_i : v_i \in V\}$

$E_d=E_{d1} \cup E_{d2}$

$E_{d1}=\{(v_j,u_k;a_1/o) : v_j \in V, u_k \in U', (v_j,v_k;a_1/o) \in E, v_k \in V\}$

$E_{d2}=\{(u_k,v_z;X_1(v_k)) : u_k \in U', v_k \in V, \mathrm{Tail}(X_1(v_k))=v_z \in V\}$

(ii) Find the minimum number of paths which cover all a_1-transitions, and start and end with an a_1-transition in G_d (denote this minimum number by δ).

(iii) Append a path corresponding to the DS to the end of each path, which forms the MOG's or MMOG's required, which is denoted by P_{mi} $(1 \le i \le \delta)$.

4. Construct a new graph $G'=(V', E')$, where

$V'=V \cup U$,

$U=\{u_j : \mathrm{Head}(P_{mi})=v_j \in V, P_{mi}$ is an MOG$\}$

$E'=E \cup E_U \cup E_g \cup E_C'$

$E_U=E_h \cup E_p \cup E_t$

$E_h=\{(v_x,u_y;i/o@X_1(v_k)) : v_x \in V, u_y \in U, (v_x,v_k;i/o) \in E, i \ne a_1, v_k \in V,$
$\mathrm{Tail}(X_1(v_k))=\mathrm{Head}(P_{mi})=v_y \in V, P_{mi}$ is an MOG$\}$

$E_p=\{(u_y,v_z;L_{mi}(v_y)) : u_y \in U, \mathrm{Head}(P_{mi})=v_y \in V, \mathrm{Tail}(P_{mi})=v_z \in V,$
P_{mi} is an MOG, $L_{mi}=\mathrm{Label}(P_{mi})\}$;

$E_t=\{(u_y,v_w; DS(v_k)-X_1(v_k)) : u_y \in U, \mathrm{Head}(P_{mi})=v_y \in V,$
P_{mi} is an MOG, $(v_x,u_y;i/o@X_1(v_k)) \in E_h - \{e_{xy}\}, e_{xy} \in E_h,$
and $\mathrm{Tail}(DS(v_k)-X_1(v_k))=v_w \in V\}$

$E_g=\{(v_j,v_k;L_{mi}(v_j)) : \mathrm{Head}(P_{mi})=v_j \in V, \mathrm{Tail}(P_{mi})=v_k \in V, P_{mi}$ is an MMOG,

$L_{mi}=\mathrm{Label}(P_{mi})\};$

$E_C'=\{(v_x, v_z; i/o@DS(v_k)) : (v_x, v_k; i/o) \in E, \mathrm{Tail}(DS(v_k))=v_z \in V, i\neq a_1,$

$\mathrm{Tail}(X_1(v_k))=v_y \in V, u_y \notin U\}$

The cost of each edge in G' is defined as the number of input (or output) symbols in its label.

5. Find an RCPT of G' over $E_U \cup E_g \cup E_C'$. It can be done by constructing an RSA G* of G', and then finding an Euler tour in G*.

It seems reasonable to obtain a minimum-cost test sequence by setting an optimization goal as achieving the maximal overlapping, since the saving by overlapping is the most significant part. This two-level optimization approach designed for the D-method is similar to that described implicitly in [YAN90] which is applied to the UIO method. However, taking advantage of the special features of a DS, Algorithm-D2 solves the local optimization problem as well as the global optimization problem efficiently.

The sufficiency condition for Algorithm-D2 to solve the problem efficiently can be summarized as follows.

Theorem 3: Given a strongly-connected digraph G=(V, E) representing a deterministic, minimal, and completely-specified FSM M=(S, I, O, δ, λ) with a DS and the reset feature, Algorithm-D2 can find a minimum-length test sequence in polynomial-time.

***Proof*:** It is obvious that steps 1-4 in Algorithm-D2 can be done in polynomial-time.

Since M has the reset feature, the test segment [ri@DS] starting from each state connects that state to Tail(DS(v_1)) if the test segment [ri@DS] is not used in overlapping, i.e., the test segment [ri@DS] $\in E_C'$. Otherwise, ri or ri@X_1 connects each state to the same $u_j \in U$, where X_1 is the intrinsically nonoverlapping prefix of the DS w.r.t. a_1@DS. So, $G[E_U \cup E_g \cup E_C']$ is a weakly-connected spanning subgraph of G'. Therefore, an RCPT of G' over $E_U \cup E_g \cup E_C'$ can be found in polynomial-time. ♦

The following theorem gives the sufficiency condition in which Algorithm-D2 obtains the optimal test sequence.

Theorem 4: Given a strongly-connected digraph G=(V, E) representing a deterministic, minimal, and completely-specified FSM M=(S, I, O, δ, λ) with a DS, Algorithm-D2 obtains the optimal test sequence if the potential maximal overlapping length for each test segment corresponding to an a_1-transition is achieved during constructing all MOG's and MMOG's except for those test segments initiating from $v_i \in V$ and $d_{in}^{E_{X_1}}(v_i)=0$ when $X_1 \neq \emptyset$, and the number of edges from G added to $G[E_U \cup E_g \cup E_C']$ in constructing an RSA G* is equal to

$$\tau - \beta - t$$

where

$\tau = \sum |\xi^{E_C}(v_i)|,\ v_i \in V,\ \xi^{E_C}(v_i) < 0;$

$\beta=1$, if $\xi^{E_C}(v_1) < 0;$

$\beta=0$, if $\xi^{E_C}(v_1) \geq 0;$

$t=|\{i : v_i \in V, \xi^{E_C}(v_i) < 0\}|$ if $X_1=\emptyset$;
$t=|\{i : v_i \in V, \xi^{E_C}(v_i) < 0, \text{ and } d_{in}^{E_{X_1}}(v_i)\neq 0\}|$ if $X_1\neq\emptyset$.

and

$G[E_{X_1}]=(V, E_{X_1})$,

$E_{X_1}=\{(v_j,v_r;X_1(v_j)) : X_1$ is the intrinsically nonoverlapping prefix of the DS w.r.t. a_1@DS, $X_1\neq\emptyset$, $Tail(X_1(v_j))=v_r \in V\}$, and $|E_{X_1}|=|V|$;

Proof: It follows from Theorem 2, Corollary 1 and Corollary 2. ◆

3.5 An Example Using Algorithm-D2

For the machine M1 shown in Figure 1, DS is *aaa* or *aab*. Note that output symbols will not be listed in the following for simplicity.

(1) DS=*aaa*

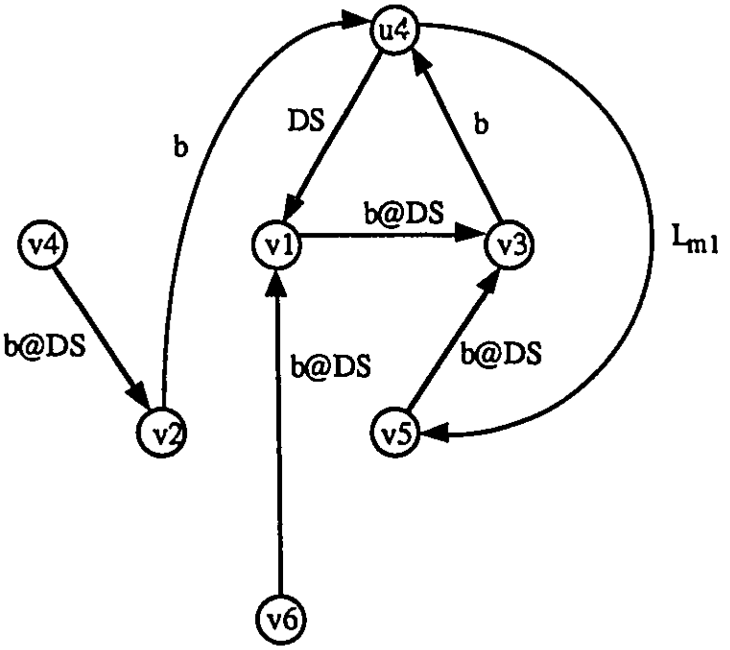
When DS=*aaa*, a path which consists of only *a*-transitions and covers all *a*-transitions is constructed as follows:

$(v_4,v_6;a)(v_6,v_2;a)(v_2,v_1;a)(v_1,v_3;a)(v_3,v_5;a)(v_5,v_2;a)$.

Then, a path corresponding to the DS is appended in the end of this path to form an MOG, i.e.,

$P_{m1}=(v_4,v_6;a)(v_6,v_2;a)(v_2,v_1;a)(v_1,v_3;a)(v_3,v_5;a)$
$(v_5,v_2;a)(v_2,v_1;a)(v_1,v_3;a)(v_3,v_5;a)$.

Based on Algorithm-D2, $G[E_U \cup E_g \cup E_C']$ is constructed as shown in Figure 5, where



L_{m1} = aaaaaa@DS

Figure 5 $G[E_U \cup E_g \cup E_d]$ for M1 when DS=*aaa*

L_{m1}=aaaaaa@DS

U={u_4}, since the only MOG P_{m1} starts from v_4 of G;

E_h={(v_2,u_4;b), (v_3,u_4;b)},

since the test segments corresponding to (v_2,v_4;b) and (v_3,u_4;b) can be overlapped by the MOG P_{m1} with the potential maximal overlapping part (v_4,v_6;a)(v_6,v_2;a)(v_2,v_1;a);

E_p={(u_4,v_5;L_{m1})}, since the MOG P_{m1} ends at v_5 of G;

E_t={(u_4,v_1;DS)},

since the test segments corresponding to (v_2,v_4;b) and (v_3,v_4;b) can be overlapped by the MOG P_{m1} with the potential maximal overlapping part (v_4,v_6;a)(v_6,v_2;a)(v_2,v_1;a), but only one of these two test segments will finally be overlapped by P_{m1}; in other words, the other one should be completed here;

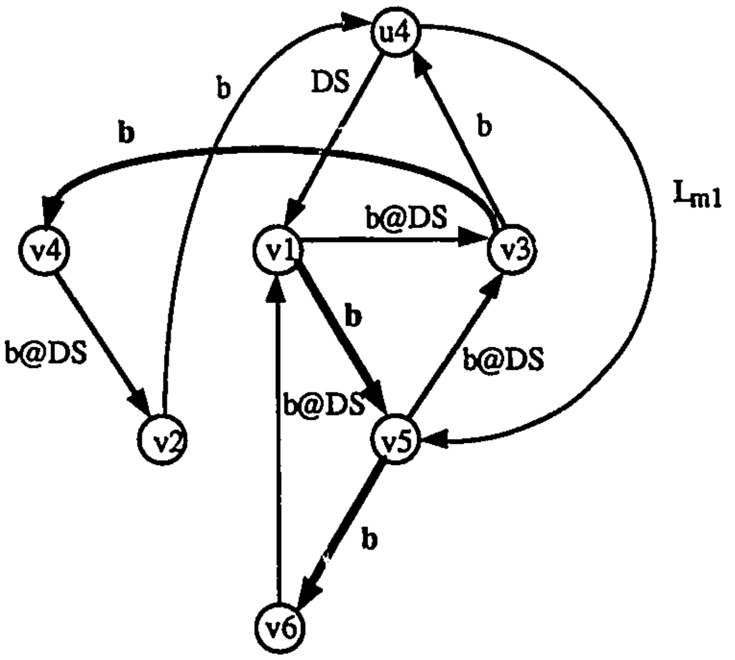
$E_U=E_h \cup E_p \cup E_t=\{(v_2,u_4;b), (v_3,u_4;b), (u_4,v_5;L_{m1}), (u_4,v_1;DS)\}$

$E_g=\phi$

$E_C'=\{(v_1,v_3;b@DS), (v_4,v_2;b@DS), (v_5,v_3;b@DS), (v_6,v_1;b@DS)\}$

Note that the edges in E_U cover those test segments corresponding to transitions $(v_2,v_4;b)$, $(v_3,v_4;b)$ and all *a*-transitions; while the edges in E_C' cover the test segments corresponding to transitions $(v_1,v_3;b)$, $(v_4,v_2;b)$, $(v_5,v_3;b)$, $(v_6,v_1;b)$. So, all test segments are covered by the edges in $E_U \cup E_g \cup E_C'$.

Then, the RSA G* of G' is constructed as shown in Figure 6, where $(v_1,v_5;b)$, $(v_5,v_6;b)$ and $(v_3,v_4;b)$ are three edges from G added to $G[E_U \cup E_g \cup E_C']$ (shown by bold lines).



L_{m1} = aaaaaa@DS

Figure 6 RSA G* of G' over $E_U \cup E_g \cup E_C'$ for M1 when DS=*aaa*

Finally, an RCPT of G' over $E_U \cup E_g \cup E_C'$ is found as follows:

$(v_1,v_3;b@DS)$, $(v_3,v_4;b)$, $(v_4,v_2;b@DS)$, $(v_2,u_4;b)$, $(u_4,v_5;L_{m1})$,
$(v_5,v_6;b)$, $(v_6,v_1;b@DS)$, $(v_1,v_5;b)$, $(v_5,v_3;b@DS)$, $(v_3,u_4;b)$, $(u_4,v_1;DS)$

The final test sequence corresponding to the RCPT is shown in Table 6, whose length is 33, the same as that of the test sequence generated by Algorithm-D1.

Transition	Test Segment
(1,5;b/1)	(1,5;b)(5,2;a)(2,1;a)(1,3;a)
	(3,4;b)
(4,1;b/1)	(4,1;b)(1,3;a)(3,5;a)(5,2;a)
(2,4;b/0)	(2,4;b)(4,6;a)(6,2;a)(2,1;a)
(4,6;a/1)	(4,6;a)(6,2;a)(2,1;a)(1,3;a)
(6,2;a/1)	(6,2;a)(2,1;a)(1,3;a)(3,5;a)
(2,1;a/0)	(2,1;a)(1,3;a)(3,5;a)(5,2;a)
(1,3;a/1)	(1,3;a)(3,5;a)(5,2;a)(2,1;a)
(3,5;a/0)	(3,5;a)(5,2;a)(2,1;a)(1,3;a)
(5,2;a/0)	(5,2;a)(2,1;a)(1,3;a)(3,5;a)
	(5,6;b)
(6,3;b/1)	(6,3;b)(3,5;a)(5,2;a)(2,1;a)
	(1,5;b)
(5,6;b/0)	(5,6;b)(6,2;a)(2,1;a)(1,3;a)
(3,4;b/1)	(3,4;b)(4,6;a)(6,2;a)(2,1;a)

Test Sequence = b/1, a/0, a/0, a/1, **b/1**, b/1, a/1, a/0, a/0, b/0, a/1,
a/1, a/0, a/1, a/0, a/0, a/0, a/1, a/0, **b/0**, b/1, a/0,
a/0, a/0, **b/1**, b/0, a/1, a/0, a/1, b/1, a/1, a/1, a/0.
(Transfer sequences are bold-faced in the test sequence.)

Test Sequence Length = 33

Table 6 A Test Sequence Generated by Algorithm-D2 for M1 when DS=*aaa*

(2) DS=*aab*

When DS=*aab*, X=aa. The duplex graph $G_d=(V_d, E_d)$ is shown in Figure 7, where

$V_d=\{v_1,v_2,v_3,v_4,v_5,v_6\}\cup\{u_1,u_2,u_3,u_4,u_5,u_6\}$

$E_{d1}=\{(v_1,u_5;b), (v_2,u_4;b), (v_3,u_4;b), (v_4,u_1;b), (v_5,u_6;b), (v_6,u_3;b)\}$

$E_{d2}=\{(u_1,v_5;X), (u_2,v_3;X), (u_3,v_2;X), (u_4,v_2;X), (u_5,v_1;X), (u_6,v_1;X)\}$

$E_d=E_{d1}\cup E_{d2}$

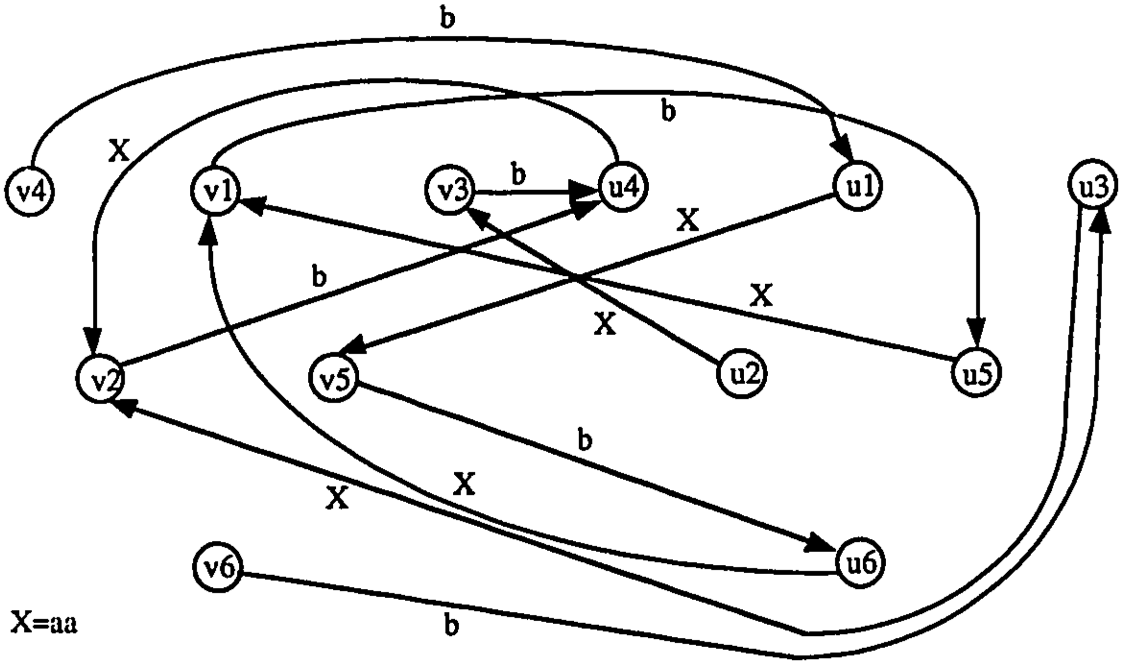


Figure 7 G_d for M1 when DS=*aab*

Three paths which cover all *b*-transitions, and start and end with a *b*-transition are constructed as follows (X=aa):

$(v_3,u_4;b)(u_4,v_2;X)(v_2,u_4;b)$,

$(v_4,u_1;b)(u_1,v_5;X)(v_5,u_6;b)(u_6,v_1;X)(v_1,u_5;b)$,

$(v_6,u_3;b)$.

Then, a path corresponding to the DS (=X@b) is appended to the end of each of these three paths. The first path forms an MOG, i.e.,

$P_{m1}=(v_3,u_4;b)(u_4,v_2;X)(v_2,u_4;b)(u_4,v_2;X)(v_2,u_4;b)$;

while the latter two paths already form MMOG's since $d_{in}^{Ex}(v_4)=d_{in}^{Ex}(v_6)=0$ (refer to Figure 4), i.e.,

$P_{m2}=(v_4,u_1;b)(u_1,v_5;X)(v_5,u_6;b)(u_6,v_1;X)(v_1,u_5;b)(u_5,v_1;X)(v_1,u_5;b)$

$P_{m3}=(v_6,u_3;b)(u_3,v_2;X)(v_2,u_4;b)$.

The graph $G[E_U \cup E_g \cup E_C']$ and the RSA G* of G' are constructed as shown in Figure 8 and Figure 9, respectively. In Figure 8:

L_{m1}=b@X@b@DS,	i.e., the label of the MOG P_{m1};
L_{m2}=b@X@b@X@b@DS	i.e., the label of the MMOG P_{m2};
L_{m3}=b@DS	i.e., the label of the MMOG P_{m3};

$U=\{u_3\}$, since the MOG P_{m1} starts from v_3 of G;

$E_h=\{(v_5,u_3;a@X), (v_6,u_3;a@X)\}$,

since the test segments corresponding to $(v_5,v_2;a)$ and $(v_6,v_2;a)$ can be overlapped by the MOG P_{m1} with the potential maximal overlapping part $(v_3,v_4;b)$;

$E_p=\{(u_3,v_4;L_{m1})\}$, since the MOG P_{m1} ends at v_4 of G;

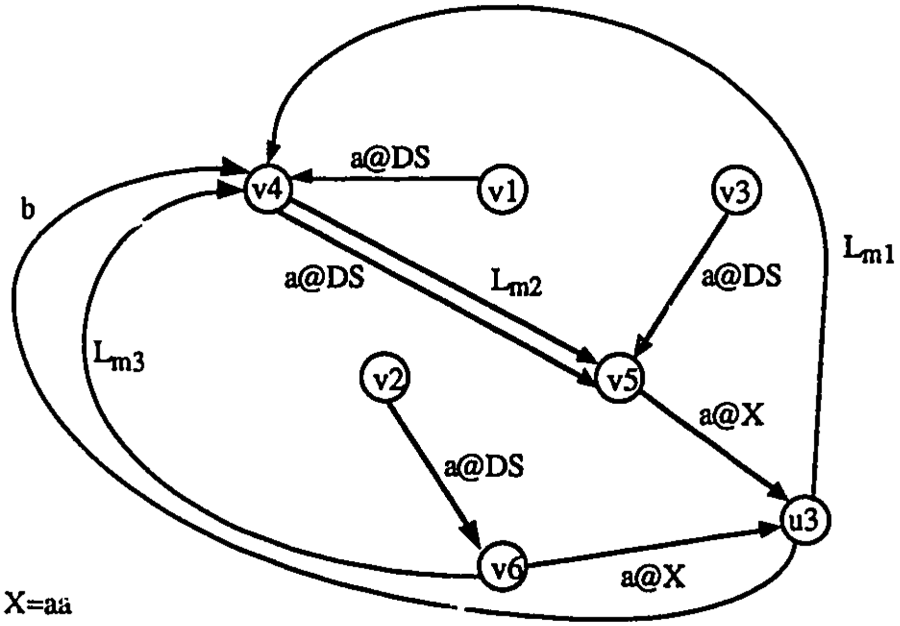
$E_t=\{(u_3,v_4;b)\}$,

since the test segments corresponding to $(v_5,v_2;a)$ and $(v_6,v_2;a)$ can be overlapped by the MOG P_{m1} with the potential maximal overlapping part $(v_3,v_4;b)$, but only one of these two test segments will finally be overlapped by P_{m1}; in other words, the other one should be completed here;

$E_U=E_h \cup E_p \cup E_t=\{(v_5,u_3;a@X), (v_6,u_3;a@X), (u_3,v_4;L_{m1}), (u_3,v_4;b)\}$

$E_g=\{(v_4,v_5;L_{m2}), (v_6,v_4;L_{m3})\}$, since P_{m2} and P_{m3} are MMOG's;

$E_C'=\{(v_1,v_4;a@DS), (v_2,v_6;a@DS), (v_3,v_5;a@DS), (v4,v_5;a@DS)\}$



X=aa

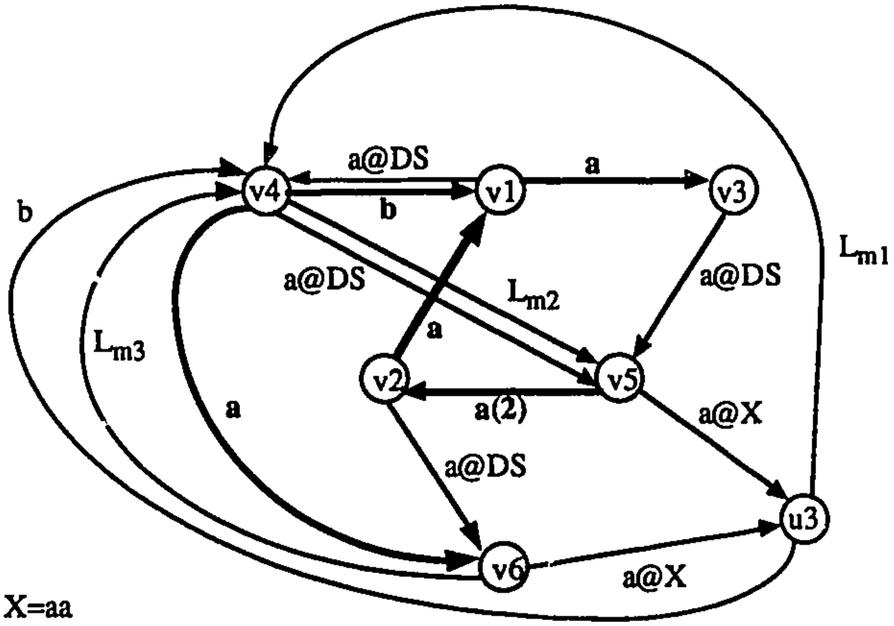
L_{m1} = b@X@b@DS
L_{m2} = b@X@b@X@b@DS
L_{m3} = b@DS

Figure 8 $G[E_U \cup E_g \cup E_C']$ for M1 when DS=*aab*

Note that the edges in E_U cover those test segments corresponding to transitions $(v_5,v_2;a)$, $(v_6,v_2;a)$ and all *b*-transitions; while the edges in E_C' cover the test segments corresponding to transitions $(v_1,v_4;a)$, $(v_2,v_6;a)$, $(v_3,v_5;a)$, $(v_4,v_5;a)$. So, all test segments are covered by the edges in $E_U \cup E_g \cup E_C'$.

In Figure 9, the edges from G added to $G[E_U \cup E_g \cup E_C']$ is shown in bold lines. An RCPT of G' over $E_U \cup E_g \cup E_C'$ is found as follows:

> $(v_1,v_4;a@DS)$, $(v_4,v_5;L_{m2})$, $(v_5,u_3;a@X)$, $(u_3,v_4;L_{m1})$, $(v_4,v_5;a@DS)$, $(v_5,v_2;a)$, $(v_2,v_6;a@DS)$, $(v_6,v_4;L_{m3})$, $(v_4,v_6;a)$, $(v_6,u_3;a@X)$, $(u_3,v_4;b)$, $(v_4,v_1;b)$, $(v_1,v_3;a)$, $(v_3,v_5;a@DS)$, $(v_5,v_2;a)$, $(v_2,v_1;a)$.



X=aa

$L_{m1} = b@X@b@DS$
$L_{m2} = b@X@b@X@b@DS$
$L_{m3} = b@DS$

Figure 9 RSA G* of G' over $E_U \cup E_g \cup E_C'$ for M1 when DS=*aab*

The corresponding test sequence is shown in Table 7, whose length is 50, while the lower bound is 46. The extra length over the lower bound is due to the transfer sequence (4,1;b)(1,3;a) and (5,2;a)(2,1;a) used in the test sequence beyond what is expected.

Transition	Test Segment
(1,3;a/1)	(1,3;a)(3,5;a)(5,2;a)(2,4;b)
(4,1;b/1)	(4,1;b)(1,3;a)(3,5;a)(5,6;b)
(5,6;b/0)	(5,6;b)(6,2;a)(2,1;a)(1,5;b)
(1,5;b/1)	(1,5;b)(5,2;a)(2,1;a)(1,5;b)
(5,2;a/0)	(5,2;a)(2,1;a)(1,3;a)(3,4;b)
(3,4;b/1)	(3,4;b)(4,6;a)(6,2;a)(2,4;b)
(2,4;b/0)	(2,4;b)(4,6;a)(6,2;a)(2,4;b)
(4,6;a/1)	(4,6;a)(6,2;a)(2,1;a)(1,5;b)
	(5,2;a)
(2,1;a/0)	(2,1;a)(1,3;a)(3,5;a)(5,6;b)
(6,3;b/1)	(6,3;b)(3,5;a)(5,2;a)(2,4;b)
	(4,6;a)
(6,2;a/1)	(6,2;a)(2,1;a)(1,3;a)(3,4;b)
	(4,1;b)
	(1,3;a)
(3,5;a/0)	(3,5;a)(5,2;a)(2,1;a)(1,5;b)
	(5,2;a)
	(2,1;a)

Test Sequence = a/1, a/0, a/0, b/0, b/1, a/1, a/0, b/0, a/1, a/0, b/1, a/0, a/0,
b/1, a/0, a/0, a/1, b/1, a/1, a/1, b/0, a/1, a/1, b/0, a/1, a/1,
a/0, b/1, **a/0**, a/0, a/1, a/0, b/0, b/1, a/0, a/0, b/0, **a/1**, a/1,
a/0, a/1, b/1, **b/1**, **a/1**, a/0, a/0, a/0, b/1, **a/0**, **a/0**.
(Transfer sequences are bold-faced in the test sequence.)

Test Sequence Length = 50

Table 7 Test Sequence Generated by Algorithm-D2 for M1 when DS=*aab*

For the machine M1 using a DS, the results can be summarized as follows:

DS	Lower Bound Without Overlapping	Algorithm-D1	Algorithm-D2	Lower Bound With Overlapping
aaa	52	33	33	32
aab	53	-	50	46

4. W-METHOD

In the W-method [CHO78, SAR84, SID89a, DAH90], a test segment for each transition consists of the transition and each component of the W-set that follows the tail of the transition. The test sequence for an FSM M is then constructed by combining test segments for all transitions of M. Some transfer sequences may be required for connecting two consequent test segments. If test segments are overlapped, the result of overlapping two test segments will not only eliminate the transfer sequence for connecting them, but also the overlapping part of two test segments will be used only once.

4.1 Lower Bounds on the Length of Test Sequences

The main goal of this section is to find a lower bound on the length of test sequences generated by any method employing a W-set with overlapping, i.e., a lower bound on the length of test sequences for the OTSO problem employing a W-set.

Similar to the definitions and notations used in section 3.1, the OTSO problem and the OTS problem employing a W-set are called the *OTSO_W problem* and the *OTS_W problem*, respectively. Also, the lower bounds on the length of test sequences for the OTSO_W problem and the OTS_W problem are denoted by LB_{OTSO_W} and LB_{OTS_W}, respectively.

As discussed in section 3.1, the OTSO_W problem is related to the OTS_W problem since overlapping of two test segments in the OTSO_W problem will not only eliminate the possible transfer sequence for connecting them which is required in the OTS_W problem, but also the overlapping part of two test segments will be used only once. Therefore, we reduce the problem of finding LB_{OTSO_W} to the problem of

(a) finding a lower bound on the length of test sequences for the OTS_W problem, i.e., LB_{OTS_W}.

(b) calculating the sum of the length of the possible overlapping parts between any two test segments (denoted by $TSOP_W$), and the sum of the length of the possible transfer sequences eliminated by overlapping (denoted by $TROP_W$).

Thus, $LB_{OTSO\_W} = LB_{OTS\_W} - TSOP_W - TROP_W$.

As discussed in section 3.1, it is not possible to find the greatest lower bound on the length of test sequences for the OTS_W problem in the general case. The following theorem establishes a lower bound on the length of test sequences for the OTS_W problem (i.e., LB_{OTS_W}) which provides an approximation to the greatest lower bound.

Theorem 5: Given a strongly-connected digraph G=(V, E) representing a deterministic, minimal, and completely-specified FSM $M=(S, I, O, \delta, \lambda)$, $|S|=n$, $|I|=q$, with a minimal W-set=$\{w_1,w_2,\ldots,w_\mu\}$, $|w_k|=L_k$, $k=1,2,\ldots,\mu$. The length of a test sequence for the OTS_W problem can not be below

$$\mu nq + nq \sum_{k=1}^{\mu} L_k + \tau - \beta$$

where

$E_C=\{(v_j, v_z; i/o@w_k(v_x)) : (v_j, v_x; i/o) \in E, i \in I, o \in O, w_k \in$ W-set, and $Tail(w_k(v_x))=v_z \in V\}$;

$\tau=\sum |\xi^{E_C}(v_i)|$, $v_i \in V$, $\xi^{E_C}(v_i) < 0$;

$\beta=1$, if $\xi^{E_C}(v_1) < 0$;
$\beta=0$, if $\xi^{E_C}(v_1) \geq 0$.

***Proof*:** nq is the number of transitions, and $\mu + \sum_{k=1}^{\mu} L_k$ is the length of the set of test segments for each transition. Without overlapping, the total length of the test sequence should be at least equal to the total length of all test segments

$$nq(\mu + \sum_{k=1}^{\mu} L_k) = \mu nq + nq \sum_{k=1}^{\mu} L_k$$

For $v_i \in V$, and $\xi^{Ec}(v_i) < 0$, to test all transitions initiating from v_i, $|\xi^{Ec}(v_i)|$ transfer sequences, each of which consists of at least one transition, will be employed to transfer M to v_i. Totally, at least an extra length $\sum |\xi^{Ec}(v_i)|$, $v_i \in V$, $\xi^{Ec}(v_i) < 0$ of transfer sequences have to be used for constructing a test sequence. If $\xi^{Ec}(v_1) < 0$, 1 should be deducted from the total added length of transfer sequences since no transfer sequence is required for the first test segment starting from v_1.

Therefore, the length of a test sequence for the OTS_W problem can not be below

$$\begin{array}{ll} \mu nq + nq \sum_{k=1}^{\mu} L_k + \tau - 1 & \text{if } \xi^{Ec}(v_1) < 0, \\ \mu nq + nq \sum_{k=1}^{\mu} L_k + \tau & \text{if } \xi^{Ec}(v_1) \geq 0, \end{array}$$

where $\tau = \sum |\xi^{Ec}(v_i)|$, $v_i \in V$, $\xi^{Ec}(v_i) < 0$. ♦

Henceforth, $\mu nq + nq \sum_{k=1}^{\mu} L_k$ and $(\tau - \beta)$ are denoted by $MRTS_{OTS\_W}$ and $MRTR_{OTS\_W}$, respectively. So, $LB_{OTS\_W} = MRTS_{OTS\_W} + MRTR_{OTS\_W} = \mu nq + nq \sum_{k=1}^{\mu} L_k + \tau - \beta$.

Since $LB_{OTSO\_W} = LB_{OTS\_W} - TSOP_W - TROP_W$, the next step is to find $TSOP_W$ and $TROP_W$ to establish LB_{OTSO_W}. Before we give a theorem to establish LB_{OTSO_W}, we will make the following observations highlighting the analysis required for the derivation of $TSOP_W$ and $TROP_W$:

1) If test segment S_b can overlap test segment S_a with a length at most l_{op}, then the length l_{op} should be added to $TSOP_W$. Due to the complexity of a W-set, l_{op} is determined by considering only the input portion of each test segment in an FSM. Then, the prefix of each test segment with which the test segment can overlap another test segment can be identified. Such a prefix of the maximum length for each test segment is called the *potential maximal overlapping part* of the test segment. The corresponding length is called the *potential maximal overlapping length*. So, the potential maximal overlapping length of a test segment is taken as the l_{op} of the test segment.

2) If test segment S_b can overlap test segment S_a, and S_b starts from $v_i \in V$, $\xi^{EC}(v_i) < 0$, then the length of the transfer sequence which is required for the OTS_W problem and counted in $MRTR_{OTS\_W}$ of LB_{OTS_W} should be added to $TROP_W$ since the transfer sequence will be eliminated due to the overlapping.

Let the set of input symbols I for an FSM be $\{a_1, a_2, \ldots, a_q\}$ and W-set for the FSM be $\{w_1, w_2, \ldots, w_\mu\}$. Then, there are only μq forms of input portions $a_j@w_k$ ($j=1,\ldots,q$, $k=1,\ldots,\mu$) among $n\mu q$ test segments for an FSM. Henceforth, a test segment with the input portion $a_j@w_k$ is called a *test segment with the form* $a_j@w_k$ (denoted by test segment $[a_j@w_k]$).

Obviously, n test segments with the same form $a_j@w_k$ ($1 \le j \le q$, $1 \le k \le \mu$) initiating from n states will have the same potential maximal overlapping length (denoted henceforth by z_{jk}). Thus, among all μnq test segments, we only need to compute the potential maximal overlapping length for μq forms of test segments $[a_j@w_k]$ ($j=1,\ldots,q$, $k=1,\ldots,\mu$).

It can be found that a W-set$=\{w_1, w_2,\ldots,w_\mu\}$, $|w_k|=L_k$ ($k=1,\ldots,\mu$), can only be in one of the following two formats:

(1) for each $w_k \in$ W-set, $w_k= a_k a_k \ldots a_k$, that is, w_k consists of L_k repeated input symbols (without loss of generality, it can be assumed that w_k ($k=1,\ldots,\mu$) consists of the first μ input symbols $a_1,\ldots,a_\mu$ in I, respectively);

(2) for some $w_k \in$ W-set ($1 \leq k \leq \mu$), $w_k \neq a_k a_k \ldots a_k$.

The following theorem gives a lower bound on the length of test sequences for the OTSO_W problem and formally presents the observations given above in detail.

Theorem 6: Given a strongly-connected digraph $G=(V, E)$ representing a deterministic, minimal, and completely-specified FSM $M=(S, I, O, \delta, \lambda)$, $|S|=n$, $|I|=q$, with a minimal W-set$=\{w_1,w_2,\ldots,w_\mu\}$, $|w_k|=L_k$, $k=1,2,\ldots,\mu$. The length of a test sequence for the OTSO_W problem can not be below

$$(\mu nq + nq \sum_{k=1}^{\mu} L_k + \tau - \beta) - (n \sum_{j=1}^{q} \sum_{k=1}^{\mu} z_{jk} - \varepsilon_{rs}) - t$$

where

$\tau=\sum |\xi^{E_C}(v_i)|$, $v_i \in V$, $\xi^{E_C}(v_i) < 0$;

$\beta=1$, if $\xi^{E_C}(v_1) < 0$,
$\beta=0$, if $\xi^{E_C}(v_1) \geq 0$;

z_{jk}, $1 \leq j \leq q$, $1 \leq k \leq \mu$, is the potential maximal overlapping length of each test segment $[a_j@w_k]$ ($1 \leq j \leq q$, $1 \leq k \leq \mu$), $a_j \in I$;

$z_{rs}=\min\{z_{jk} : j=1,\dots,q,\ k=1,\dots,\mu\},\ 1 \le r \le q,\ 1 \le s \le \mu;$

$$t= \sum_{v_i \in V;\ \xi^{E_C}(v_i)<0} \min(|\xi^{E_C}(v_i)|,\ |\{z_{jk} : z_{jk}\neq 0,\ j=1,\dots,q,\ k=1,\dots,\mu\}|).$$

***Proof*:** There are two possible cases for a given W-set.

Case 1: Each w_k of the W-set is in the format of $a_k a_k \cdots\cdots a_k$, that is, each w_k consists of L_k repeated input symbols, $1 \le k \le \mu$ (without loss of generality, it can be assumed that w_k $(k=1,\dots,\mu)$ consists of the first μ input symbols $a_1,\dots,a_\mu$ in I, respectively).

a_k, $k=1,\dots,\mu$, are different input symbols due to the feature of a minimal W-set. So, we have $\mu \le q$.

There are μq test segments $[a_j@w_k]$ $(a_j \in I,\ 1 \le j \le q,\ 1 \le k \le \mu)$ initiating from each state $v_i \in V$ $(1 \le i \le n)$. They can be listed as follows:

$$\begin{array}{ccccccc}
\multicolumn{4}{c}{\underline{\qquad\qquad B_1 \qquad\qquad}} & \multicolumn{3}{c}{\underline{\qquad\quad B_2 \qquad\quad}} \\
[a_1@w_1] & [a_2@w_1] & \cdots\cdots & [a_\mu@w_1] & [a_{\mu+1}@w_1] & \cdots\cdots & [a_q@w_1] \\
[a_1@w_2] & [a_2@w_2] & \cdots\cdots & [a_\mu@w_2] & [a_{\mu+1}@w_2] & \cdots\cdots & [a_q@w_2] \\
\vdots & \vdots & & \vdots & \vdots & & \vdots \\
[a_1@w_\mu] & [a_2@w_\mu] & \cdots\cdots & [a_\mu@w_\mu] & [a_{\mu+1}@w_\mu] & \cdots\cdots & [a_q@w_\mu]
\end{array}$$

The potential maximal overlapping length of each test segment $[a_k@w_k]$ $(1 \le k \le \mu)$ initiating from each state $v_i \in V$ $(1 \le i \le n)$ (i.e., the test segments in the diagonal entries of B_1) is L_k, i.e., $z_{kk}=L_k$ $(1 \le k \le \mu)$. Meanwhile, the potential maximal overlapping length of each test segment $[a_k@w_v]$ $(k\neq v,\ 1 \le k \le \mu,\ 1 \le v \le \mu)$ initiating from each state $v_i \in V$

$(1 \leq i \leq n)$ (i.e., the test segments in the non-diagonal entries of B_1) is 1, i.e., z_{kv}=1 ($k \neq v$, $1 \leq k \leq \mu$, $1 \leq v \leq \mu$). Finally, each test segment $[a_j@w_k]$ ($\mu+1 \leq j \leq q$, $1 \leq k \leq \mu$) initiating from each state $v_i \in V$ $(1 \leq i \leq n)$ (i.e., the test segments in B_2) cannot overlap any other test segment, i.e., z_{jk}=0 ($\mu+1 \leq j \leq q$, $1 \leq k \leq \mu$).

In the best case, each test segment $[a_k@w_k]$ $(1 \leq k \leq \mu)$ (i.e., the test segments in the diagonal entries of B_1) can overlap another test segment $[a_j@w_k]$ $(1 \leq j \leq q, 1 \leq k \leq \mu)$ with its potential maximal overlapping length L_k, and each test segment $[a_k@w_v]$ ($k \neq v$, $1 \leq k \leq \mu$, $1 \leq v \leq \mu$) (i.e., the test segments in the non-diagonal entries of B_1) can overlap another test segment $[a_j@w_k]$ $(1 \leq j \leq q, 1 \leq k \leq \mu)$ with its potential maximal overlapping length 1. Moreover, when $\mu < q$, the first test segment in a test sequence initiating from the initial state should correspond to a transition other than a_k-transition; and when μ=q, the first test segment in a test sequence initiating from the initial state should be one of test segments $[a_k@w_v]$ ($k \neq v$, $1 \leq k \leq \mu$, $1 \leq v \leq \mu$) (i.e., the test segments in the non-diagonal entries of B_1).

Since there are $n\mu$ test segments in the diagonal entries of B_1 (i.e., $[a_k@w_k]$ $(k=1,\ldots,\mu)$ initiating from n states), the total length of the potential maximal overlapping length for these test segments is $n \sum_{k=1}^{\mu} L_k$. Also, since there are $n\mu(\mu-1)$ test segments in the non-diagonal entries of B_1 (i.e., $[a_k@w_v]$ ($k \neq v$, $k,v=1,\ldots,\mu$) initiating from n states), the total length of the potential maximal overlapping length for these test segments is $n\mu(\mu-1)$. Therefore, if $\mu < q$, the maximal saving of length $n(\sum_{k=1}^{\mu} L_k + \mu(\mu-1))$ on test segments can be obtained, compared with $MRTS_{OTS\_W}$. Otherwise,

μ=q, and thus, the maximal saving of length $n(\sum_{k=1}^{\mu} L_k + \mu(\mu-1)) - 1$ on test segments can be obtained, compared with $MRTS_{OTS\_W}$, where the deduction 1 is due to the fact that the first test segment in the test sequence initiating from the initial state cannot overlap any other test segment with its potential maximal overlapping length 1. So, $TSOP_W=n(\sum_{k=1}^{\mu} L_k + \mu(\mu-1))$ when $\mu < q$, or $TSOP_W=n(\sum_{k=1}^{\mu} L_k + \mu(\mu-1)) - 1$ when μ=q.

Meanwhile, due to the overlapping, no transfer sequences are required for the test segments $[a_k@w_v]$ (k,v=1,...,μ) in B_1 initiating from $v_i \in V$, and $\xi^{E_C}(v_i) < 0$ (i=1,...,n). In the best case, the total number of test segments initiating from state v_i that can overlap other test segments is μ^2. So, for each $v_i \in V$, and $\xi^{E_C}(v_i) < 0$ (i=1,...,n), the total length of transfer sequences that may be saved cannot be larger than the minimum of $|\xi^{E_C}(v_i)|$ and μ^2. Therefore, a saving of length $t= \sum_{v_i \in V;\, \xi^{E_C}(v_i)<0} \min(|\xi^{E_C}(v_i)|, \mu^2)$ on transfer sequences can be obtained, compared with $MRTR_{OTS\_W}$, i.e., $TROP_W = t$.

So, the length of a test sequence for the OTSO_W problem can not be below $(MRTS_{OTS\_W} - TSOP_W) + (MRTR_{OTS\_W} - TROP_W)$ that is equal to

$$(\mu nq + nq \sum_{k=1}^{\mu} L_k + \tau - \beta) - n(\sum_{k=1}^{\mu} L_k + \mu(\mu-1)) - t \qquad \text{when } \mu < q;$$

$$\text{or} \quad (\mu nq + nq \sum_{k=1}^{\mu} L_k + \tau - \beta) - [n(\sum_{k=1}^{\mu} L_k + \mu(\mu-1)) - 1] - t \text{ when } \mu = q.$$

Since $z_{kk}=L_k$ (k=1,...,μ); $z_{kv}=1$ ($k \neq v$, k,v=1,...,μ); $z_{jk}=0$ (j=μ+1,...,q, k=1,...,μ), we get

$$z_{rs}=\min\{z_{jk} : j=1,\ldots,q,\ k=1,\ldots,\mu\}=0,\ (\mu+1 \leq r \leq q,\ 1 \leq s \leq \mu),$$

when $\mu < q$,

or $z_{rs}=\min\{z_{jk} : j=1,\ldots,q,\ k=1,\ldots,\mu\}=1,\ (r\neq s,\ 1 \leq r \leq q,\ 1 \leq s \leq \mu)$,

when $\mu = q$.

Consequently, we get

$$(n \sum_{j=1}^{q} \sum_{k=1}^{\mu} z_{jk} - z_{rs})=n(\sum_{k=1}^{\mu} L_k + \mu(\mu-1)) \qquad \text{when } \mu < q, \text{ or}$$

$$(n \sum_{j=1}^{q} \sum_{k=1}^{\mu} z_{jk} - z_{rs})=[n(\sum_{k=1}^{\mu} L_k + \mu(\mu-1)) - 1] \qquad \text{when } \mu = q.$$

Meanwhile, we get $|\{z_{jk} : z_{jk}\neq 0, j=1,\ldots,q, k=1,\ldots,\mu\}| = \mu^2$.

So, the lower bound is proved in this case.

Case 2: Some or all w_k's of the W-set is in the general format; that is, not all w_k's or none of them consists of L_k repeated input symbols, $k=1,2,\ldots,\mu$.

Similar to Case 1, there are μq test segments $[a_j@w_k]$ ($a_j \in I$, $1 \leq j \leq q$, $1 \leq k \leq \mu$) initiating from each state $v_i \in V$ ($1 \leq i \leq n$). It can be found that the potential maximal overlapping length of each test segment $[a_j@w_k]$ ($1 \leq j \leq q$, $1 \leq k \leq \mu$) is z_{jk}, with which it can overlap another test segment $[a_u@w_v]$ ($1 \leq u \leq q$, $1 \leq v \leq \mu$).

In the best case, each test segment with $z_{jk}\neq 0$ can overlap another test segment with its potential maximal overlapping length, except that the first test segment in a test sequence initiating from the initial state cannot overlap any other test segment. But, the first test segment should be one of those test segments with the minimum of z_{jk} ($j=1,\ldots,q$, $k=1,\ldots,\mu$), so that the total length of the potential maximal overlapping part (i.e., $TSOP_W$) is maximized in this best case.

Thus, the maximal saving on test segments compared with $MRTS_{OTS\_W}$ can be represented by the sum of the potential maximal overlapping length z_{jk} for each test segment except the first test segment in the test sequence initiating from the initial state, i.e., $TSOP_W = (n \sum_{j=1}^{q} \sum_{k=1}^{\mu} z_{jk} - z_{rs})$, where $z_{rs} = \min\{z_{jk} : j=1,\ldots,q,\ k=1,\ldots,\mu\}$ $(1 \le r \le q,\ 1 \le s \le \mu)$ represents the minimum of z_{jk} $(j=1,\ldots,q,\ k=1,\ldots,\mu)$ among all test segments initiating from the initial state.

Obviously, a test segment initiating from $v_i \in V$ with $z_{jk} \neq 0$ indicates that the test segment can overlap another test segment in the best case. Thus, due to the overlapping, no transfer sequences are required for these test segments $[a_j@w_k]$ $(1 \le j \le q,\ 1 \le k \le \mu)$ initiating from $v_i \in V$ and $\xi^{Ec}(v_i) < 0$ with $z_{jk} \neq 0$. The total number of test segments initiating from $v_i \in V$ that can overlap other test segments is $|\{z_{jk} : z_{jk} \neq 0, j=1,\ldots,q, k=1,\ldots,\mu\}|$. So, for each $v_i \in V$ and $\xi^{Ec}(v_i) < 0$, the total length of transfer sequences that may be saved compared with $MRTR_{OTS\_W}$, cannot be larger than the minimum of $|\xi^{Ec}(v_i)|$ and $|\{z_{jk} : z_{jk} \neq 0,\ j=1,\ldots,q,\ k=1,\ldots,\mu\}|$. Therefore, a saving of length

$$t = \sum_{v_i \in V;\ \xi^{Ec}(v_i) < 0} \min(|\xi^{Ec}(v_i)|,\ |\{z_{jk} : z_{jk} \neq 0,\ j=1,\ldots,q,\ k=1,\ldots,\mu\}|)$$

on transfer sequences can be obtained, compared with $MRTR_{OTS\_W}$, i.e., $TROP_W = t$.

So, the length of a test sequence for the OTSO_W problem can not be below $(MRTS_{OTS\_W} - TSOP_W) + (MRTR_{OTS\_W} - TROP_W)$ that is

$$(\mu n q + nq \sum_{k=1}^{\mu} L_k + \tau - \beta) - (n \sum_{j=1}^{q} \sum_{k=1}^{\mu} z_{jk} - z_{rs}) - t \qquad n$$

The lower bound given in Theorem 5 is LB_{OTSO_W}, in which $TROP_W=t$, representing the observations described in 2), and $TSOP_W=(n \sum_{j=1}^{q} \sum_{k=1}^{\mu} z_{jk} - z_{rs})$, representing the observations described in 1). When a W-set is in the format of (1), LB_{OTSO_W} can be simplified as the following corollary.

Corollary 3: Given a strongly-connected digraph G=(V, E) representing a deterministic, minimal, and completely-specified FSM $M=(S, I, O, \delta, \lambda)$, $|S|=n$, $|I|=q$, with a minimal W-set$=\{w_1,w_2,\ldots,w_\mu\}$, $w_k=a_k a_k \ldots\ldots a_k$ ($a_k \in I$), $|w_k|=L_k$, $k=1,2,\ldots,\mu$. The length of a test sequence for the OTSO_W problem can not be below

$$(\mu nq + nq \sum_{k=1}^{\mu} L_k + \tau - \beta) - [n(\sum_{k=1}^{\mu} L_k + \mu(\mu-1)) - \gamma] - t$$

where

$\tau=\sum |\xi^{E_C}(v_i)|$, $v_i \in V$, $\xi^{E_C}(v_i) < 0$;

$\beta=1$, if $\xi^{E_C}(v_1) < 0$,
$\beta=0$, if $\xi^{E_C}(v_1) \geq 0$;

$\gamma=0$, if $\mu < q$,
$\gamma=1$, if $\mu = q$;

$$t= \sum_{v_i \in V;\ \xi^{E_C}(v_i)<0} \min(|\xi^{E_C}(v_i)|, \mu^2).$$

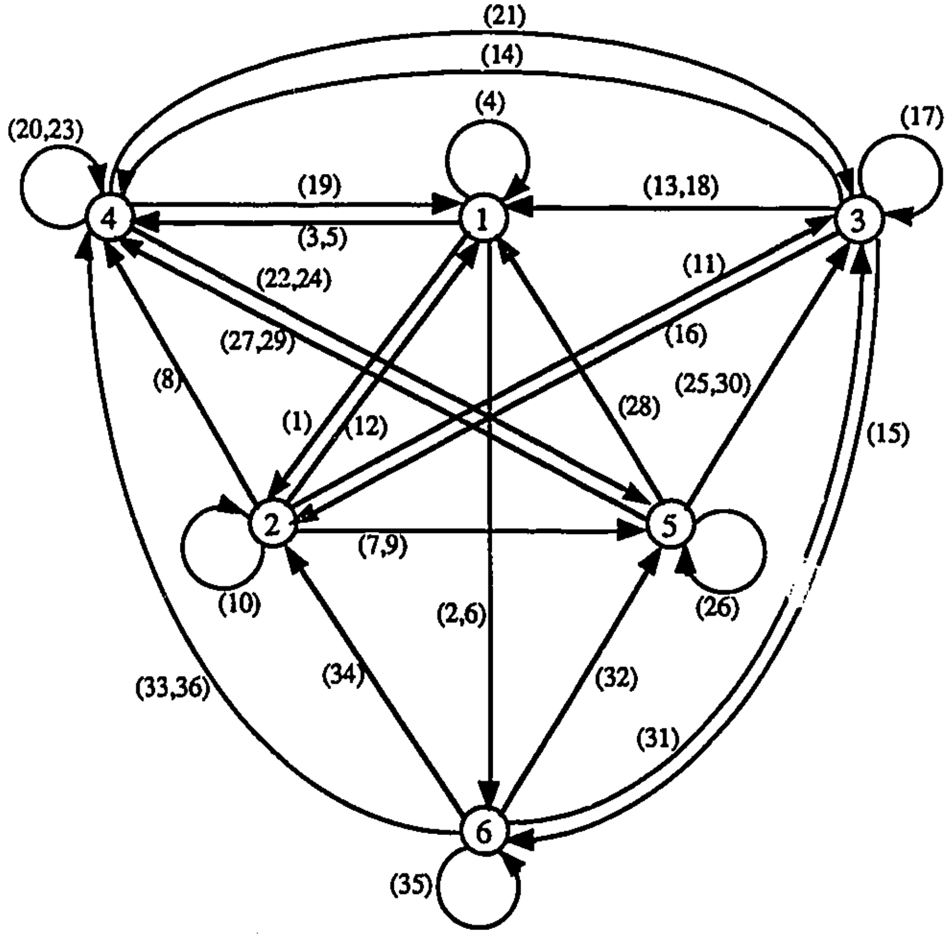
Example:

The W-set for the machine M1 shown in Figure 1 is {aa, ab, b} or {aa, bb}. We have n=6, q=2.

(1) W-set={aa, ab, b}

When W-set={aa, ab, b} is used, we know $\mu=3$, $L_1=2$, $L_2=2$, $L_3=1$. G[E_C] of M1 based on this W-set is shown in Figure 10, where $\xi^{E_C}(v_1)=\xi^{E_C}(v_3)=\xi^{E_C}(v_5)=0$, $\xi^{E_C}(v_2)=\xi^{E_C}(v_6)=-2$, $\xi(v_4)=4$. Table 8 lists all the test segments, i.e., the edges in G[E_C].

From Theorem 5, we have $\tau=|\xi^{E_C}(v_2)|+|\xi^{E_C}(v_6)|=4$, since $\xi^{E_C}(v_2)=\xi^{E_C}(v_6)=-2 < 0$, and $\beta=0$, since $\xi^{E_C}(v_1) \geq 0$. So, the lower bound of the length of a test sequence for the OTS_W problem = $\mu nq + nq \sum_{k=1}^{\mu} L_k + \tau - \beta = 100$.



(...) indicates the identification number of test segments

Figure 10 G[E_C] for M1 when W-set = {aa, ab, b}

S1 (1,3;a/1)(3,5;a/0)(5,2;a/0) = (1,2;a/1@w1(3))
S2 (1,3;a/1)(3,5;a/0)(5,6;b/0) = (1,6;a/1@w2(3))
S3 (1,3;a/1)(3,4;b/1) = (1,4;a/1@w3(3))
S4 (1,5;b/1)(5,2;a/0)(2,1;a/0) = (1,1;b/1@w1(5))
S5 (1,5;b/1)(5,2;a/0)(2,4;b/0) = (1,4;b/1@w2(5))
S6 (1,5;b/1)(5,6;b/0) = (1,6;b/1@w3(5))
S7 (2,1;a/0)(1,3;a/1)(3,5;a/0) = (2,5;a/0@w1(1))
S8 (2,1;a/0)(1,3;a/1)(3,4;b/1) = (2,4;a/0@w2(1))
S9 (2,1;a/0)(1,5;b/1) = (2,5;a/0@w3(1))
S10 (2,4;b/0)(4,6;a/1)(6,2;a/1) = (2,2;b/0@w1(4))
S11 (2,4;b/0)(4,6;a/1)(6,3;b/1) = (2,3;b/0@w2(4))
S12 (2,4;b/0)(4,1;b/1) = (2,1;b/0@w3(4))
S13 (3,5;a/0)(5,2;a/0)(2,1;a/0) = (3,1;a/0@w1(5))
S14 (3,5;a/0)(5,2;a/0)(2,4;b/0) = (3,4;a/0@w2(5))
S15 (3,5;a/0)(5,6;b/0) = (3,6;a/0@w3(5))
S16 (3,4;b/1)(4,6;a/1)(6,2;a/1) = (3,2;b/1@w1(4))
S17 (3,4;b/1)(4,6;a/1)(6,3;b/1) = (3,3;b/1@w2(4))
S18 (3,4;b/1)(4,1;b/1) = (3,1;b/1@w3(4))
S19 (4,6;a/1)(6,2;a/1)(2,1;a/0) = (4,1;a/1@w1(6))
S20 (4,6;a/1)(6,2;a/1)(2,4;b/0) = (4,4;a/1@w2(6))
S21 (4,6;a/1)(6,3;b/1) = (4,3;a/1@w3(6))
S22 (4,1;b/1)(1,3;a/1)(3,5;a/0) = (4,5;b/1@w1(1))
S23 (4,1;b/1)(1,3;a/1)(3,4;b/1) = (4,4;b/1@w2(1))
S24 (4,1;b/1)(1,5;b/1) = (4,5;b/1@w3(1))
S25 (5,2;a/0)(2,1;a/0)(1,3;a/1) = (5,3;a/0@w1(2))
S26 (5,2;a/0)(2,1;a/0)(1,5;b/1) = (5,5;a/0@w2(2))
S27 (5,2;a/0)(2,4;b/0) = (5,4;a/0@w3(2))
S28 (5,6;b/0)(6,2;a/1)(2,1;a/0) = (5,1;b/0@w1(6))
S29 (5,6;b/0)(6,2;a/1)(2,4;b/0) = (5,4;b/0@w2(6))
S30 (5,6;b/0)(6,3;b/1) = (5,3;b/0@w3(6))
S31 (6,2;a/1)(2,1;a/0)(1,3;a/1) = (6,3;a/1@w1(2))
S32 (6,2;a/1)(2,1;a/0)(1,5;b/1) = (6,5;a/1@w2(2))
S33 (6,2;a/1)(2,4;b/0) = (6,4;a/1@w3(2))
S34 (6,3;b/1)(3,5;a/0)(5,2;a/0) = (6,2;b/1@w1(3))
S35 (6,3;b/1)(3,5;a/0)(5,6;b/0) = (6,6;b/1@w2(3))
S36 (6,3;b/1)(3,4;b/1) = (6,4;b/1@w3(3))

Table 8 Test Segments for M1 when W-set = {aa, ab, b}

From Theorem 6, W-set={aa, ab, b} belongs to case 2. We can obtain z_{jk} corresponding to each test segment $[a_j@w_k]$ (j=1,2; k=1,2,3):

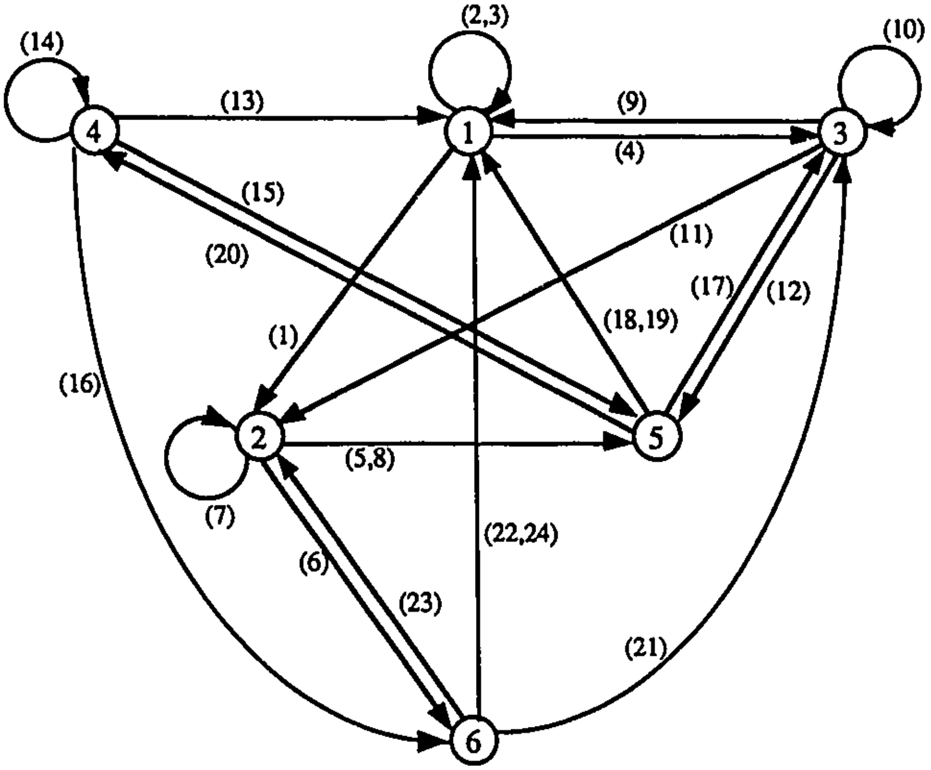
		k=1	k=2	k=3
		w_1=**aa**	w_2=**ab**	w_3=**b**
j=1	a_1=**a**	$z_{11}=2$ $\{w_1\}$	$z_{12}=2$ $\{w_1\}$	$z_{13}=2$ $\{w_2\}$
j=2	a_2=**b**	$z_{21}=1$ $\{w_2,w_3\}$	$z_{22}=1$ $\{w_2,w_3\}$	$z_{23}=1$ $\{w_2,w_3\}$

where { } indicates the w_v $(1 \le v \le 3)$, such that test segment $[a_j@w_k]$ $(1 \le j \le 2, 1 \le k \le 3)$ can overlap another test segment $[a_u@w_v]$ $(1 \le u \le 2, 1 \le v \le 3)$ with its potential maximal overlapping length. For example, the item in row 1 and column 2 indicates that test segment $[a_1@w_2]$ can overlap another test segment $[a_1@w_1]$ or $[a_2@w_1]$ with its potential maximal overlapping length. Thus, $t=|\xi^{Ec}(v_2)|+|\xi^{Ec}(v_6)|=4$, since $|\{z_{jk} : z_{jk}\neq 0, j=1,2, k=1,2,3\}|=6 > |\xi^{Ec}(v_2)|=|\xi^{Ec}(v_6)|=2$, and $z_{rs}=\min\{z_{jk} : j=1,2, k=1,2,3\}=1$. So, the lower bound of the length of a test sequence for the OTSO_W problem = $(\mu nq + nq \sum_{k=1}^{\mu} L_k + \tau - \beta) - (n \sum_{j=1}^{q} \sum_{k=1}^{\mu} z_{jk} - z_{rs}) - t = 100 - (54 - 1) - 4 = 43$.

(2) W-set={aa, bb}

When W-set={aa, bb}, we know $\mu=2$ and $L_1=L_2=2$. $G[E_C]$ of M1 based on this W-set is shown in Figure .11, where $\xi^{Ec}(v_1)=4$, $\xi^{Ec}(v_2)=\xi^{Ec}(v_3)=\xi^{Ec}(v_5)=0$, $\xi^{Ec}(v_4)=\xi^{Ec}(v_6)=-2$.

From Theorem 5, we have $\tau=|\xi^{Ec}(v_4)|+|\xi^{Ec}(v_6)|=4$, since $\xi^{Ec}(v_4)=\xi^{Ec}(v_6)=-2 < 0$, and $\beta=0$, since $\xi^{Ec}(v_1) > 0$. So, the lower bound of the length of a test sequence for the OTS_W problem = $\mu nq + nq \sum_{k=1}^{\mu} L_k + \tau - \beta = 76$.



(...) indicates the identification number of test segments

Figure 11 $G[E_C]$ for M1 when W-set = {aa, bb}

Since W-set={aa, bb} is in format (1), from Corollary 3, we know γ=1 since μ=q, and t=$|\xi^{E_C}(v_4)|+|\xi^{E_C}(v_6)|$=4, since $\mu^2=4 > |\xi^{E_C}(v_4)|=|\xi^{E_C}(v_6)|=2$. So, the lower bound of the length of a test sequence for the OTSO_W = $(\mu nq + nq \sum_{k=1}^{\mu} L_k + \tau - \beta) - [n(\sum_{k=1}^{\mu} L_k + \mu(\mu-1)) - \gamma] - t = 76 - (36 - 1) - 4 = 37$.

S1 (1,3;a/1)(3,5;a/0)(5,2;a/0) = (1,2;a/1@w1(3))
S2 (1,3;a/1)(3,4;b/1)(4,1;b/1) = (1,1;a/1@w2(3))
S3 (1,5;b/1)(5,2;a/0)(2,1;a/0) = (1,1;b/1@w1(5))
S4 (1,5;b/1)(5,6;b/0)(6,3;b/1) = (1,3;b/1@w2(5))
S5 (2,1;a/0)(1,3;a/1)(3,5;a/0) = (2,5;a/0@w1(1))
S6 (2,1;a/0)(1,5;b/1)(5,6;b/0) = (2,6;a/0@w2(1))
S7 (2,4;b/0)(4,6;a/1)(6,2;a/1) = (2,2;b/0@w1(4))
S8 (2,4;b/0)(4,1;b/1)(1,5;b/1) = (2,5;b/0@w2(4))
S9 (3,5;a/0)(5,2;a/0)(2,1;a/0) = (3,1;a/0@w1(5))
S10 (3,5;a/0)(5,6;b/0)(6,3;b/1) = (3,3;a/0@w2(5))
S11 (3,4;b/1)(4,6;a/1)(6,2;a/1) = (3,2;b/1@w1(4))
S12 (3,4;b/1)(4,1;b/1)(1,5;b/1) = (3,5;b/1@w2(4))
S13 (4,6;a/1)(6,2;a/1)(2,1;a/0) = (4,1;a/1@w1(6))
S14 (4,6;a/1)(6,3;b/1)(3,4;b/1) = (4,4;a/1@w2(6))
S15 (4,1;b/1)(1,3;a/1)(3,5;a/0) = (4,5;b/1@w1(1))
S16 (4,1;b/1)(1,5;b/1)(5,6;b/0) = (4,6;b/1@w2(1))
S17 (5,2;a/0)(2,1;a/0)(1,3;a/1) = (5,3;a/0@w1(2))
S18 (5,2;a/0)(2,4;b/0)(4,1;b/1) = (5,1;a/0@w2(2))
S19 (5,6;b/0)(6,2;a/1)(2,1;a/0) = (5,1;b/0@w1(6))
S20 (5,6;b/0)(6,3;b/1)(3,4;b/1) = (5,4;b/0@w2(6))
S21 (6,2;a/1)(2,1;a/0)(1,3;a/1) = (6,3;a/1@w1(2))
S22 (6,2;a/1)(2,4;b/0)(4,1;b/1) = (6,1;a/1@w2(2))
S23 (6,3;b/1)(3,5;a/0)(5,2;a/0) = (6,2;b/1@w1(3))
S24 (6,3;b/1)(3,4;b/1)(4,1;b/1) = (6,1;b/1@w2(3))

Table 9 Test Segments for M1 when W-set = {aa, bb}

4.2 Algorithm-W1

As in section 3.2, Algorithm-W1 combines the approach using a maximum-cardinality minimum-cost matching algorithm w[illegible] W-set, which is described as follows:

Algorithm-W1:

1. Given a strongly connected digraph G=(V, E) representing a deterministic, minimal, and completely-specified FSM M which possesses a minimal W-set $=\{w_1, w_2, \ldots, w_\mu\}$.

2. Construct a set of test segments for each transition by concatenating the transition with each w_k ($k=1,2,\ldots,\mu$) in the W-set.

   Denote each test segment by S_i, $i=1,2,\cdots\cdots,\mu nq$.

3. Construct bipartite graph $G_b=(V_b, E_b)$ where

   $V_b=V_t \cup V_h$

   $V_t=\{v_1\} \cup \{v_i : v_i=\mathrm{Tail}(S_k) \text{ for } k=1,2,\cdots\cdots,\mu nq\}$
   $V_h=\{v_1\} \cup \{v_j : v_j=\mathrm{Head}(S_k) \text{ for } k=1,2,\cdots\cdots,\mu nq\}$
   (Here, the inclusion of the initial state guarantees that the test sequence starts and ends at the initial state.)

   $E_b=\{(v_i,v_j) : v_i \in V_t, v_j \in V_h; v_i \neq \mathrm{Tail}(S_k), v_j \neq \mathrm{Head}(S_k) \text{ for some } k\}$
   (Note: for a test segment S_k, the edge from $\mathrm{Tail}(S_k)$ to $\mathrm{Head}(S_k)$ should not be included in E_b.)

   Assign cost C_{ij} to edge $(v_i,v_j) \in E_b$ according to the following rules:

   (a) Let $(v_i,v_j)=(\mathrm{Tail}(S_x), \mathrm{Head}(S_y))$. If S_y can overlap S_x, then a length equal to the negative of the overlapping length is assigned to edge (v_i,v_j) as its cost.

   (b) Otherwise, the length of the shortest path from v_i to v_j in G is assigned to edge (v_i,v_j) as its cost.

(c) The length of the shortest path from v_1 to v_i and from v_i to v_1 in G is assigned to (v_1,v_i) and (v_i,v_1), respectively, as their costs.

4. Use a maximum-cardinality minimum-cost matching algorithm for graph G_b.

The solution to the above matching problem is an alternating sequence of test segments, overlapped test segment groups and transfer sequences connecting them. If the solution is a tour, it is a valid test sequence. Otherwise, we may have to connect the disconnected components using, for example, a greedy heuristic, which is very similar to the algorithm for constructing a directed spanning graph or arborescence [PAP82].

Similar to the analysis for Algorithm-D1, the time complexity of Algorithm-W1 is of order $\mu^2|E|^2$.

4.3 An Example Using Algorithm-W1

(1) W-set={aa, ab, b}

For the machine M1 shown in Figure 1, if W-set={aa, ab, b} is used, the test sequence generated by Algorithm-W1 is shown in Table 10. The length of the test sequence is 46. It is interesting to note that this length is equal to the lower bound of the length of the test sequence for M1 (i.e., 43), plus the total length of the part in the test sequence which is expected to be an overlapping part, but can not be (i.e., the length 1 of (4,1;b) in the test segment (4,1;b)(1,3;a)(3,5;a)), plus the total length of the transfer sequence needed to terminate the test sequence at state 1 (i.e., the length 2 of the transfer sequence (5,2;a)(2,1;a)).

Transition	Test Segment
(1,5;b/1)	(1b5)(5a2)(2a1)
(5,2;a/0)	(5a2)(2a1)(1a3)
(2,1;a/0)	(2a1)(1a3)(3a5)
(1,3;a/1)	(1a3)(3a5)(5a2)
(3,5;a/0)	(3a5)(5a2)(2a1)
(5,2;a/0)	(5a2)(2a1)(1b5)
(2,1;a/0)	(2a1)(1b5)
(1,5;b/1)	(1b5)(5a2)(2b4)
(5,2;a/0)	(5a2)(2b4)
(2,4;b/0)	(2b4)(4a6)(6a2)
(4,6;a/1)	(4a6)(6a2)(2a1)
(6,2;a/1)	(6a2)(2a1)(1a3)
(2,1;a/0)	(2a1)(1a3)(3b4)
(1,3;a/1)	(1a3)(3b4)
(3,4;b/1)	(3b4)(4a6)(6a2)
(4,6;a/1)	(4a6)(6a2)(2b4)
(6,2;a/1)	(6a2)(2b4)
(2,4;b/0)	(2b4)(4a6)(6b3)
(4,6;a/1)	(4a6)(6b3)
(6,3;b/1)	(6b3)(3a5)(5a2)
(3,5;a/0)	(3a5)(5a2)(2b4)
(2,4;b/0)	(2b4)(4b1)
(4,1;b/1)	(4b1)(1b5)
(1,5;b/1)	(1b5)(5b6)
(5,6;b/0)	(5b6)(6b3)
(6,3;b/1)	(6b3)(3b4)
(3,4;b/1)	(3b4)(4b1)
(4,1;b/1)	(4b1)(1a3)(3b4)
(3,4;b/1)	(3b4)(4a6)(6b3)
(6,3;b/1)	(6b3)(3a5)(5b6)
(5,6;b/0)	(5b6)(6a2)(2b4)
(4,1;b/1)	**(4b1)**(1a3)(3a5)
(1,3;a/1)	(1a3)(3a5)(5b6)
(3,5;a/0)	(3a5)(5b6)
(5,6;b/0)	(5b6)(6a2)(2a1)
(6,2;a/1)	(6a2)(2a1)(1b5)
	(5a2)
	(2a1)

(Note: each transition is denoted by (s i s') due to the space limitation.
Underline is used to show the overlapping part between two test segments which can not be shown by indentation due to the space limitation.
Bold is used to show the part of a test segment which is expected to be the overlapping part, but can not be.)

Test Sequence = b/1, a/0, a/0, a/1, a/0, a/0, a/0, b/1, a/0, b/0, a/1, a/1,
a/0, a/1, b/1, a/1, a/1, b/0, a/1, b/1, a/0, a/0, b/0, b/1,
b/1, b/0, b/1, b/1, b/1, a/1, b/1, a/1, b/1, a/0, b/0, a/1,
b/0, b/1, a/1, a/0, b/0, a/1, a/0, b/1, **a/0, a/0.**
(Transfer sequences are bold-faced in the test sequence.)

Test Sequence Length = 46

Table 10 A Test Sequence Generated by Algorithm-W1 for M1 when W-set={aa, ab, b}

(2) W-set={aa, bb}

If W-set={aa, bb} is used, the test sequence generated by Algorithm-W1 is shown in Table 11. The length of the test sequence is 42, which is equal to the lower bound of the length of the test sequence for M1 (i.e., 37), plus the total length (i.e., 2) of the part in the test sequence which is expected to be the overlapping part, but can not be (i.e., (2,4;b) in the test segment (2,4;b)(4,6;a)(6,2;a) and (4,6;a) in the test segment (4,6;a)(6,3;b)(3,4;b)), plus the total length (i.e., 3) of the transfer sequences (1,5;b)(5,2;a) and (3,4;b).

4.4 Algorithm-W2

As discussed in section 3.4 for any method employing a DS, the OTSO problem can be reduced to the RCPP and the Euler tour algorithm can be applied instead of the matching algorithm. In the following, such an algorithm, Algorithm-W2, is proposed, in which a W-set for a given FSM M=(S, I, O, δ, λ) is in the following format: every w_k ($k=1,\ldots,\mu$) in the W-set consists of repeated input symbols $a_k \in I$.

Transition	Test Segment
(1,3;a/1)	(1a3)(3b4)(4b1)
(3,4;b/1)	(3b4)(4b1)(1b5)
(4,1;b/1)	(4b1)(1b5)(5b6)
(5,6;b/0)	(5b6)(6a2)(2a1)
	(1b5)
	(5a2)
(2,4;b/0)	**(2b4)**(4a6)(6a2)
(4,6;a/1)	(4a6)(6a2)(2a1)
(6,2;a/1)	(6a2)(2a1)(1a3)
(2,1;a/0)	(2a1)(1a3)(3a5)
(1,3;a/1)	(1a3)(3a5)(5a2)
(3,5;a/0)	(3a5)(5a2)(2a1)
(5,2;a/0)	(5a2)(2a1)(1a3)
	(3b4)
(4,6;a/1)	**(4a6)**(6b3)(3b4)
(6,3;b/1)	(6b3)(3b4)(4b1)
(4,1;b/1)	(4b1)(1a3)(3a5)
(3,5;a/0)	(3a5)(5b6)(6b3)
(5,6;b/0)	(5b6)(6b3)(3b4)
(3,4;b/1)	(3b4)(4a6)(6a2)
(6,2;a/1)	(6a2)(2b4)(4b1)
(2,4;b/0)	(2b4)(4b1)(1b5)
(1,5;b/1)	(1b5)(5a2)(2a1)
(2,1;a/0)	(2a1)(1b5)(5b6)
(1,5;b/1)	(1b5)(5b6)(6b3)
(6,3;b/1)	(6b3)(3a5)(5a2)
(5,2;a/0)	(5a2)(2b4)(4b1)

(Note: refer to Table 10 for the notation.)

Test Sequence = a/1, b/1, b/1, b/1, b/0, a/1, a/0, **b/1, a/0**, b/0, a/1, a/1, a/0, a/1,
a/0, a/0, a/0, a/1, **b/1**, a/1, b/1, b/1, b/1, a/1, a/0, b/0, b/1, b/1,
a/1, a/1, b/0, b/1, b/1, a/0, a/0, b/1, b/0, b/1, a/0, a/0, b/0, b/1.

(Transfer sequences are bold-faced in the test sequence.)

Test Sequence Length = 42

Table 11 A Test Sequence Generated by Algorithm-W1 for M1 when W-set={aa, bb}

The main idea of Algorithm-W2 is to solve the OTSO_W problem in three levels:

Level 1: for every n test segments $[a_k@w_k]$ $(1 \leq k \leq \mu)$ initiating from n states, find MOG's within these n test segments;

Level 2: find all MMOG's by considering the MOG's obtained in Level 1 and the test segments $[a_j@w_k]$ $(j=1,2,\ldots,q,\ k=1,2,\ldots,\mu,\ j \neq k)$;

Level 3: combine all MMOG's optimally.

The problems stated in Level 1 and Level 2 are called the *local optimization problems* of the OTSO_W problem; while the problem stated in Level 3 is called the *global optimization problem* of the OTSO_W problem.

The local optimization problem stated in Level 1 can be solved by constructing a minimum number of paths in G which cover every a_k-transition $(1 \leq k \leq \mu)$ exactly once. This can be done by an efficient algorithm similar to that in [GON70]. Then, each of these paths corresponding to a_k-transitions $(1 \leq k \leq \mu)$ can be transformed into an MOG by appending a path corresponding to the w_k to the end of the path, since every test segment $[a_k@w_k]$ except the first one in the extended path overlap its previous one with its potential maximal overlapping length $|w_k|$.

To solve the local optimization problem stated in Level 2, a new digraph $G_m=(V_m, E_m)$ is constructed from a strongly connected digraph $G=(V, E)$ representing a deterministic, minimal and completely-specified FSM M as follows:

$V_m = V \cup U \cup V_1 \cup V_2 \cup \ldots \cup V_q$ where

V_j $(j=1,2,\ldots,q)$ the vertex set V is duplicated into q vertex sets V_j $(j=1,2,\ldots,q)$;

U each vertex in U corresponds to the head of an MOG obtained in Level 1;

$E_m = E_U \cup E_V$

E_U each edge in E_U is connected with a vertex in U and V_j $(1 \leq j \leq q)$, respectively. E_U consists of E_{U1}, E_{U2} and E_{U3}:

E_{U1} - each edge in E_{U1} connects a vertex in V_j $(1 \leq j \leq q)$ to a vertex in U and corresponds to the a_j-transition in a test segment $[a_j@w_k]$ $(1 \leq j \leq q, 1 \leq k \leq \mu, j \neq k)$, where the tail of the a_j-transition is the head of an MOG for test segments $[a_k@w_k]$;

E_{U2} - each edge in E_{U2} connects a vertex in U to a vertex in V_k $(1 \leq k \leq \mu)$ and corresponds to an MOG for test segments $[a_k@w_k]$ $(1 \leq k \leq \mu)$ without the last a_k-transition.

It can be noted that traversing a pair of edges from E_{U1} and E_{U2}, respectively, will force the first test segment of an MOG for test segments $[a_k@w_k]$ $(1 \leq k \leq \mu)$ to overlap a test segment $[a_j@w_k]$ $(1 \leq j \leq q, 1 \leq k \leq \mu, j \neq k)$ with its potential maximal overlapping length.

E_{U3} - each edge in E_{U3} connects a vertex in U to a vertex in V_k $(1 \leq k \leq \mu)$ and corresponds to the w_k in a test segment $[a_j@w_k]$ $(1 \leq j \leq q, 1 \leq k \leq \mu, j \neq k)$ without the last a_k-transition, where the tail of the a_j-transition is the head of an MOG for test segments $[a_k@w_k]$.

It can be noted that traversing a pair of edges from E_{U1} and E_{U3}, respectively, will cover one of the test segments $[a_j@w_k]$ $(1 \leq j \leq q, 1 \leq k \leq \mu, j \neq k)$ except the last a_k-transition. One less than the number of the test segments $[a_j@w_k]$ need to be considered here, since one of them will be covered through traversing a pair of edges, one from E_{U1} and the other from E_{U2}.

E_V each edge in E_V connects a vertex in V to a vertex in V_j $(1 \leq j \leq q)$ or vice versa. E_V consists of E_{V1} and E_{V2}:

E_{V1} - each edge in E_{V1} connects a vertex in V_j $(1 \leq j \leq q)$ to a vertex in V and corresponds to the a_j-transition in a test segment $[a_j@w_k]$ $(1 \leq j \leq q, 1 \leq k \leq \mu, j \neq k)$, where the tail of the a_j-transition is not the head of any MOG for test segments $[a_k@w_k]$;

E_{V2} - each edge in E_{V2} connects a vertex in V to a vertex in V_k $(1 \leq k \leq \mu)$ and corresponds to the w_k in a test segment $[a_j@w_k]$ $(1 \leq j \leq q, 1 \leq k \leq \mu, j \neq k)$ without the last a_k-transition, where the tail of the a_j-transition is not the head of any MOG for test segments $[a_k@w_k]$.

It can be noted that traversing a pair of edges from E_{V1} and E_{V2}, respectively, will cover one of the test segments $[a_j@w_k]$ $(1 \leq j \leq q, 1 \leq k \leq \mu, j \neq k)$ except the last a_k-transition.

Obviously, there is no edge

a) from a vertex in V to a vertex in U or vice versa;

b) from a vertex in V_j $(j=1,\ldots,q)$ to a vertex in $V_{j'}$ $(j'=1,\ldots,q)$;

c) from a vertex in U or V to a vertex in V_j $(j=\mu+1,\ldots,q)$.

It is important to note that traversing an edge from V_j $(1 \leq j \leq q)$ to U (or V) should be followed by traversing an edge from U (or V) to V_k $(1 \leq k \leq \mu, k \neq j)$ so as to guarantee a test segment $[a_j@w_k]$ $(1 \leq j \leq q, 1 \leq k \leq \mu, j \neq k)$ be covered instead of a test segment $[a_k@w_k]$ which has already been covered in Level 1. This can be guranteed by constructing a minimum number of paths in G_m which cover every edge in G_m exactly once with the restriction that an edge from V_k $(1 \leq k \leq \mu)$ to U (or V) cannot be followed by an edge from U (or V) to V_k $(1 \leq k \leq \mu)$, and each of the paths in G_m should start from a vertex in V_j $(1 \leq j \leq q)$ and end at a vertex in V_k $(1 \leq k \leq \mu)$. The minimum number of such paths can be constructed by an efficient algorithm similar to that in [GON70]. Furthermore, each of these paths can be transformed into an MMOG by appending an a_k-transition to the end of the path, since each test segment $[a_k@w_v]$ $(1 \leq k \leq \mu,\ 1 \leq v \leq \mu,\ k \neq v)$ except the first test segment in the extended path can overlap its previous one with its potential maximal overlapping length 1 and each test segment $[a_k@w_k]$ $(1 \leq k \leq \mu)$ in the extended path can overlap its previous one with its potential maximal overlapping length $|w_k|$.

Finally, to solve the global optimization problem stated in Level 3, a new digraph $G'=(V', E')$ is constructed such that $V' = V$ and $E' = E \cup E_g$ where E_g is the set of edges corresponding to all MMOG's. Thus, this problem can be solved by finding an RCPT of G' over E_g. As stated earlier, there are efficient algorithms to find an RCPT of G' over E_g if $G[E_g]$ is a weakly-connected spanning subgraph of G'.

Algorithm-W2 which uses the approach discussed above is given as follows:

Algorithm-W2:

1. Given a strongly connected digraph $G=(V, E)$ representing a deterministic, minimal, and completely-specified FSM M which possesses a minimal W-set=$\{w_1,w_2,\ldots,w_\mu\}$, in which every w_k $(k=1,2,\ldots,\mu)$ consists of repeated input symbols $a_k \in I$.

2. Construct a set of test segments for each transition by concatenating the transition with each w_k ($k=1,2,\ldots,\mu$) in the W-set.

3. Compute all MMOG's.

   3.1 For each $w_k=a_k a_k \ldots a_k$ ($k=1,2,\ldots,\mu$).

   (i) Find the minimal number of paths which consists of a_k-transitions only and covers every a_k-transition in G exactly once (denote this minimum number by ρ_k).

   (ii) Append a path corresponding to the w_k to the end of each path. The resulting path which corresponds to an MOG is denoted by P_{r,a_k} $(1 \leq r \leq \rho_k)$.

   3.2 (i) Construct a new graph $G_m=(V_m, E_m)$

   where

   $V_m=V \cup U \cup V_1 \cup V_2 \cup \ldots \cup V_q$

   $V_j=\{v_{ji} : v_i \in V\}$, $j=1,2,\ldots,q$

   $U=\{u_i : Head(P_{r,a_k})=v_i \in V, P_{r,a_k}$ corresponds to an MOG$\}$

   $E_m=E_U \cup E_V$

   $E_U=E_{U1} \cup E_{U2} \cup E_{U3}$

   $E_{U1}=\{(v_{jx}, u_y; a_j/o) : v_{jx} \in V_j, u_y \in U,$
   $(v_x, v_y; a_j/o) \in E, j \neq k,$
   $Head(P_{r,a_k})=v_y \in V, k=1,\ldots,\mu\}$

   $E_{U2}=\{(u_y, v_{kz}; L'_{r,a_k}) : u_y \in U, v_{kz} \in V_k,$
   $Head(P'_{r,a_k})=v_y \in V, Tail(P'_{r,a_k})=v_z \in V,$

$P'_{r,a_k} = P_{r,a_k}$ - last a_k-transition in P_{r,a_k},

$L'_{r,a_k}=\text{Label}(P'_{r,a_k}),\ k=1,\ldots,\mu\}$

$E_{U3}=\{(u_y,v_{kz};w_k'(v_y)) : u_y \in U,\ \text{Head}(P'_{r,a_k})=v_y \in V,$

$P'_{r,a_k} = P_{r,a_k}$ - last a_k-transition in P_{r,a_k},

$v_{kz} \in V_k,\ \text{Tail}(w_k'(v_y))=v_z \in V,$

$w_k'=w_k- a_k,\ k=1,\ldots,\mu,\ j\neq k$

$(v_{jx},u_y;\ a_j/o) \in E_{U1}-\{e_{xy}\},\ e_{xy} \in E_{U1}\}$

$E_V=E_{V1}\cup E_{V2}$

$E_{V1}=\{(v_{jx},v_y;\ a_j/o) : v_{jx} \in V_j,\ v_y \in V,\ a_j \in I,$

$(v_x,\ v_y;\ a_j/o) \in E,$

$u_y \notin U,\ \ j=1,2,\ldots,q\}$

$E_{V2}=\{(v_y,v_{kz};\ w_k'(v_y)) : v_y \in V,\ (v_{jx},v_y;\ a_j/o) \in E_{V1},$

$v_{kz} \in V_k,\ k\neq j,\ \text{Tail}(w_k'(v_y))=v_z \in V,$

$w_k'=w_k - a_k\ ,\ \ k=1,2,\ldots,\mu\}$

(ii) Find the minimum number of paths which cover every edge in G_m exactly once (denote this minimum number by δ).

(iii) Append an a_k-transition to the end of each path in G_m. Each resulting path which terminates at a vertex in V_k $(1 \leq k \leq \mu)$ to form an MMOG is denoted by P_{mi} $(1 \leq i \leq \delta)$.

4. Construct a new graph $G'=(V', E')$, where

$V'=V$

$E'=E\cup E_g$

$E_g=\{(v_x,v_y;L_{mi}(v_x)) : Head(P_{mi})=v_x \in V,\ Tail(P_{mi})=v_y \in V,$

P_{mi} is an MMOG, $L_{mi}=Label(P_{mi})\}$

The cost of each edge in G' is defined as the number of input (or output) symbols in its label.

5. Find an RCPT of G' over E_g. It can be done by first constructing an RSA G* of G', and then finding an Euler tour in G*.

As Algorithm-D2, Algorithm-W2 also obtains an minimum-cost test sequence by setting an optimization goal as achieving the maximal overlapping, since the saving by overlapping is the most significant part. However, due to the complexity of the approach using a W-set, a three-level optimization approach is used here instead of the two-level optimization approach that is used in the approach using a DS. It can be found that the local optimization problems in the first two levels can be solved efficiently by step 3.1 and 3.2 of Algorithm-W2. However, the global optimization problem in the third level can be solved efficiently in step 4 of Algorithm-W2 if $G[E_g]$ is a weakly-connected spanning subgraph of G' [AHO88].

When $G[E_g]$ is weakly-connected but not a spanning subgraph of G', we can construct yet another graph G"=(V", E"), where

$V''=\{v_x : v_x=Head(P_{mi}) \in V\}\cup\{v_y : v_y=Tail(P_{mi}) \in V\}$ where P_{mi} is an MMOG;

$E''=E_p\cup E_g$

$E_p=\{(v_x,v_y;\ SP_{xy}(v_x)) : v_x,\ v_y \in V'',\ SP_{xy}$ is the label of the shortest path from v_x to v_y in G'$\}$

$E_g=\{(v_x,v_y;L_{mi}(v_x)) : Head(P_{mi})=v_x \in V,\ Tail(P_{mi})=v_y \in V,\ L_{mi}=Label(P_{mi})\}$

It can be noted that the way for constructing the new graph G" guarantees that G[E_g] is a spanning subgraph of G". Thus, G[E_g] becomes a weakly-connected spanning subgraph of G". Therefore, finding an RCPT of G" over E_g can be solved in polynomial-time.

The following lemma indicates that an RCPT of G" over E_g is also an RCPT of G' over E_g.

Lemma 1: If G[E_g] is weakly connected, then an RCPT of G"=(V", E") over E_g is also an RCPT of G'=(V', E') over E_g.

***Proof*:** Since G[E_g] is weakly connected and is a spanning subgraph of G", an RCPT of G" over E_g can be found in polynomial-time.

Such an RCPT of G" over E_g can be found in two steps: first, an RSA G"*=(V"*, E"*) of G" over E_g can be constructed; then, an Euler tour is found on G"*. The E"* consists of all edges in E_g and the minimum cost of edges from E" which are required to make G[E_g] symmetric.

Based on G"*, we can construct a rural symmetric augmentation of G' over E_g simply by extending these edges that are in E"* and belong to E"-E_g into paths in G', since every edge in E"-E_g corresponds to a shortest path between two vertices of G[E_g] in G'. This rural symmetric augmentation of G' over E_g must be an RSA G'* of G' over E_g. If not, it implies that there is another rural symmetric augmentation (denoted by G'+=(V'+, E'+)) of G' over E_g which costs less than G'*. Also, if there is such one, we can always find one such that all edges in E'+-E_g (which are also belong to E') forms several shortest paths between vertices in G[E_g]. Otherwise, we can replace those paths between vertices in G[E_g] with the shortest paths between them to get a rural symmetric augmentation with less cost.

Based on G'^{+}, we can also construct a rural symmetric augmentation (denoted by G''^{+}) of G" over E_g simply by combining those shortest paths between vertices in $G[E_g]$ that consists of edges in $E'^{+}-E_g$ into edges in G", since all edges in G" corresponds to a shortest paths between vertices in $G[E_g]$ that consists of edges in G'. So, we get another rural symmetric augmentation with less cost than the RSA G''^{*} of G" over E_g. This is contrary to the given condition.

Therefore, an RSA of G" over E_g corresponds to an RSA of G' over E_g. Consequently, the RCPT of G" over E_g that is found by finding an Euler tour on the RSA is also an RCPT of G' over E_g. ♦

The following theorem gives the sufficiency condition in which Algorithm-W2 obtains the optimal test sequence.

Theorem 7: Given a strongly-connected digraph G=(V, E) representing a deterministic, minimal, and completely-specified FSM $M=(S, I, O, \delta, \lambda)$ with a minimal W-set=$\{w_1,w_2,\ldots,w_\mu\}$, $w_k=a_ka_k\ldots\ldots a_k$ $(a_k \in I)$, $|w_k|=L_k$, $k=1,2,\ldots,\mu$, Algorithm-W2 obtains the optimal test sequence if the potential maximal overlapping length for each test segment is achieved during constructing all MMOG's, and the number of edges from G added to $G[E_g]$ in constructing an RSA G^* is equal to

$$\tau - \beta - t$$

where

$\tau=\sum |\xi^{Ec}(v_i)|,\ v_i \in V,\ \xi^{Ec}(v_i) < 0;$

$\beta=1$, if $\xi^{Ec}(v_1) < 0$,

$\beta=0$, if $\xi^{Ec}(v_1) \geq 0$;

$$t= \sum_{v_i \in V;\ \xi^{Ec}(v_i)<0} \min(|\xi^{Ec}(v_i)|, \mu^2).$$

Proof: It follows from Theorem 6 and Corollary 3. ◆

4.5 An Example Using Algorithm-W2

For the machine M1 shown in Figure 1, we consider W-set={aa,bb}. Note that output symbols will not be listed in the following for simplicity.

In step 3.1, P_{1a} and P_{1b} are obtained as paths consisting of all *a*-transitions followed by w_1 and all *b*-transitions followed by w_2, respectively:

P_{1a}=$(v_4,v_6;a)(v_6,v_2;a)(v_2,v_1;a)(v_1,v_3;a)$
$(v_3,v_5;a)(v_5,v_2;a)(v_2,v_1;a)(v_1,v_3;a)$,
P_{1b}=$(v_2,v_4;b)(v_4,v_1;b)(v_1,v_5;b)(v_5,v_6;b)$
$(v_6,v_3;b)(v_3,v_4;b)(v_4,v_1;b)(v_1,v_5;b)$.

In step 3.2.(i), graph G_m is obtained as shown in Figure 12:

$V_m=V \cup U \cup V_1 \cup V_2$
$V_1=\{v_{11}, v_{12}, v_{13}, v_{14}, v_{15}, v_{16}\}$
$V_2=\{v_{21}, v_{22}, v_{23}, v_{24}, v_{25}, v_{26}\}$
$U=\{u_2, u_4\}$, since P_{1b} and P_{1a} starts from v_2 and v_4 of G, respectively;
$E_m=E_U \cup E_V$
$E_U=E_{U1} \cup E_{U2} \cup E_{U3}$
P_{1a}'=$(v_4,v_6;a)(v_6,v_2;a)(v_2,v_1;a)(v_1,v_3;a)$
$(v_3,v_5;a)(v_5,v_2;a)(v_2,v_1;a)$, i.e., P_{1a} without the last transition;

P_{1b}'=(v_2,v_4;b)(v_4,v_1;b)(v_1,v_5;b)(v_5,v_6;b)

(v_6,v_3;b)(v_3,v_4;b)(v_4,v_1;b), i.e., P_{1b} without the last transition;

L_{1a}'=aaaaaaa i.e., the label of P_{1a}';

L_{1b}'=bbbbbbb i.e., the label of P_{1b}';

w_1'=a i.e., w_1 without the last input symbol;

w_2'=b i.e., w_2 without the last input symbol;

E_{U1}={(v_{16},u_2;a), (v_{15},u_2;a), (v_{23},u_4;b), (v_{22},u_4;b)},

since the test segments (v_6,v_2;a)@w_2 and (v_5,v_2;a)@w_2 can be overlapped by P_{1b} with the potential maximal overlapping part (v_2,v_4;b)(v_4,v_1;b); and also the test segments (v_3,v_4;b)@w_1 and (v_2,v_4;b)@w_1 can be overlapped by P_{1a} with the potential maximal overlapping part (v_4,v_6;a)(v_6,v_2;a).

E_{U2}={(u_2,v_{21};L_{1b}'), (u_4,v_{11};L_{1a}')},

since P_{1b}' starts from v_2 and ends at v_1; and P_{1a}' starts from v_4 and ends at v_1;

E_{U3}={(u_2,v_{24};w_2'), (u_4,v_{16};w_1')},

since the test segments (v_6,v_2;a)@w_2 and (v_5,v_2;a)@w_2 can be overlapped by P_{1b} with the potential maximal overlapping part (v_2,v_4;b)(v_4,v_1;b), but only one of them will in fact be overlapped by P_{1b}, only one of these two test segments should be completed here; and also the test segments (v_3,v_4;b)@w_1 and (v_2,v_4;b)@w_1 can be overlapped by P_{1a} with the potential maximal overlapping part (v_4,v_6;a)(v_6,v_2;a), but only one of them will in fact be overlapped by P_{1a}, only one of these two test segments should be completed here.

E_{V1}={(v_{11},v_3;a), (v_{12},v_1;a), (v_{13},v_5;a), (v_{14},v_6;a),

(v_{21},v_5;b), (v_{24},v_1;b), (v_{25},v_6;b), (v_{26},v_3;b)},

since the test segments (v_1,v_3;a)@w_2, (v_2,v_1;a)@w_2, (v_3,v_5;a)@w_2, and (v_4,v_6;a)@w_2 cannot be overlapped by P_{1b}; and the test segments

$(v_1,v_5;b)@w_1$, $(v_4,v_1;b)@w_1$, $(v_5,v_6;b)@w_1$, and $(v_6,v_3;b)@w_1$ cannot be overlapped by P_{1a};

$E_{V2}=\{(v_3,v_{24};w_2'), (v_1,v_{25};w_2'), (v_5,v_{26};w_2'), (v_6,v_{23};w_2'),$
$(v_5,v_{12};w_1'), (v_1,v_{13};w_1'), (v_6,v_{12};w_1'), (v_3,v_{15};w_1')\}$,

since these test segments which cannot be overlapped by P_{1b} or P_{1a} should be completed here;

After step 3.2, we find the following paths that cover all edges in G_m, i.e., cover all test segments:

$P_{m1} = (v_{22},u_4;b)(u_4,v_{11};L_{1a}')(v_{11},v_{24};a@w_2')(v_{24},v_{13};b@w_1')(v_{13},v_{26};a@w_2')$
$(v_{26},v_{15};b@w_1')(v_{15},u_2;a)(u_2,v_{21};L_{1b}')(v_{21},v_{12};b@w_1')(v_{12},v_{25};a@w_2')$
$(v_{25},v_{12};b@w_1')(v_{12},v_1;a)$

$P_{m2} = (v_{14},v_{23};a@w_2')(v_{23},u_4;b)(u_4,v_{16};w_1')(v_{16},u_2;a)(u_2,v_{24};w_2')(v_{24},v_1;b)$

In step 4, G' is constructed. However, $G[E_g]$ is not a spanning subgraph of G' since P_{m1} and P_{m2} connect with v_1, v_2 and v_4 only. Thus, G" is obtained as shown in Figure 13, where

$V"=\{v_1, v_2, v_4\}$, since P_{m1} and P_{m2} connect with v_1, v_2 and v_4 only;

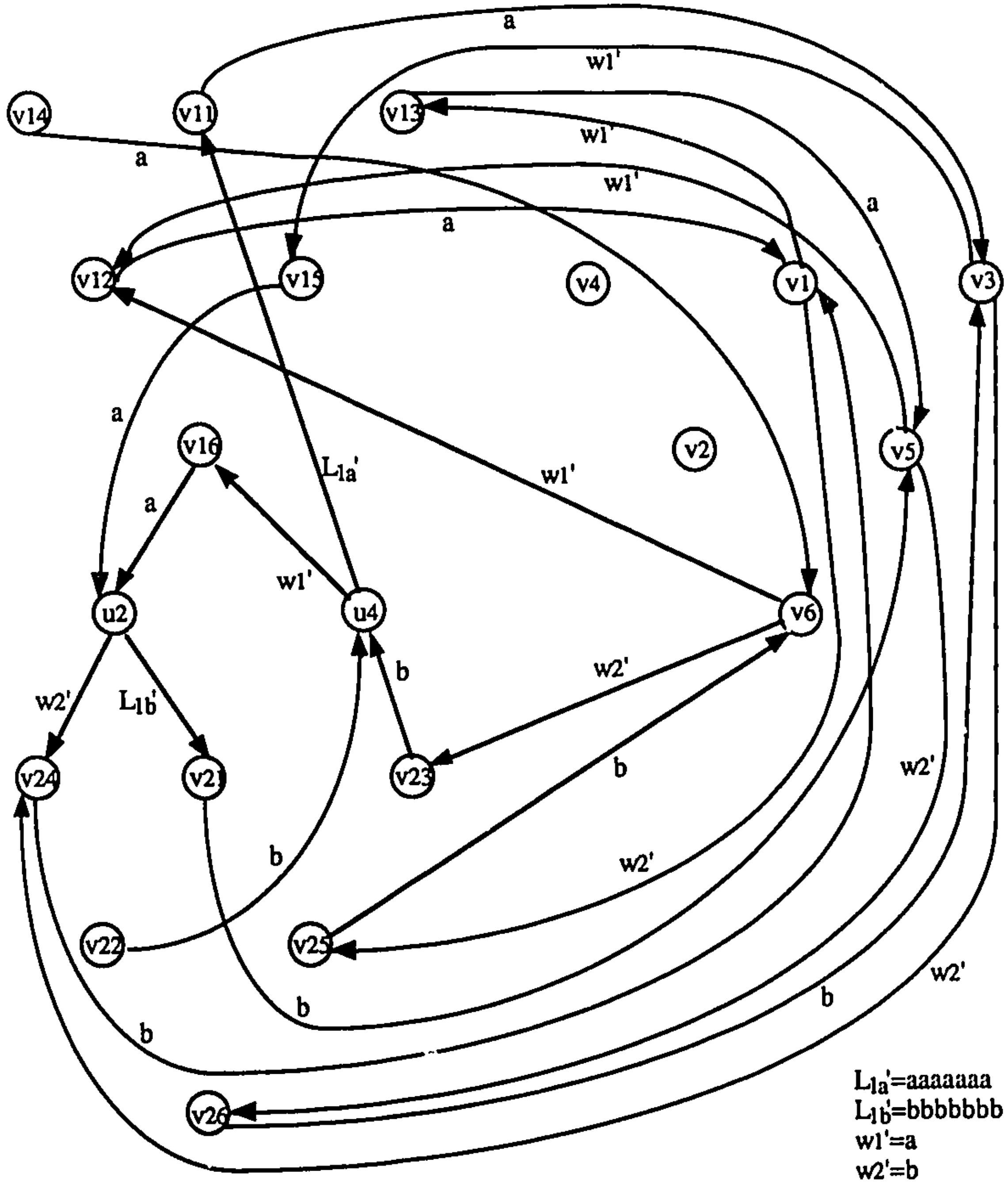
$E_p=\{(v_1,v_2;ba), (v_2,v_1;a), (v_1,v_4;ab), (v_4,v_1;b), (v_2,v_4;b), (v_4,v_2;aa)\}$,

since the input portions of the labels correspond to the shortest paths from v_1 to v_2, v_2 to v_1, v_1 to v_4, v_4 to v_1, v_2 to v_4 and v_4 to v_2 are ba, a, ab, b, b and aa, respectively;

$L_{m1}= b@L_{1a}'@a@w_2'@b@w_1'@a@w_2'@b@w_1'@a@L_{1b}'@$
$b@w_1'@a@w_2'@b@w_1'@a$, i.e., the label of P_{m1};

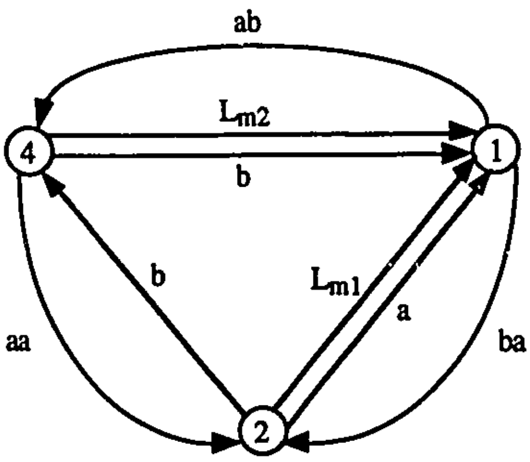
$L_{m2}= a@w_2'@b@w_1'@a@w_2'@b$, i.e., the label of P_{m2};

$E_g=\{(v_2,v_1;L_{m1}), (v_4,v_1;L_{m2})\}$.



L_{1a}'=aaaaaaa
L_{1b}'=bbbbbbb
w1'=a
w2'=b

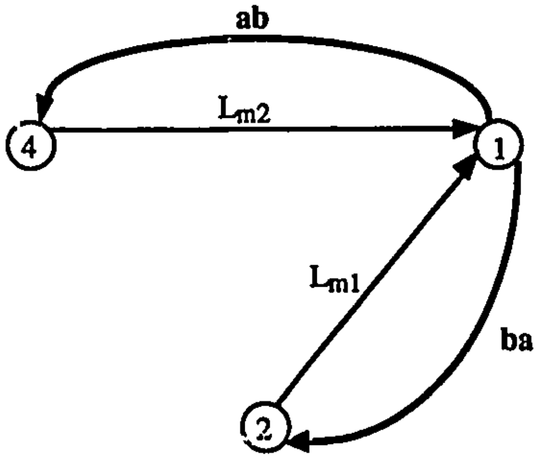
Figure 12 G_m for M1 when W-set={aa,bb}



L_{m1}=b@L1a'@a@w2'@b@w1'@a@w2'@b@w1'@a@L1b'@b@w1'@a@w2'@b@w1'@a
L_{m2}=a@w2'@b@w1'@a@w2'@b

Figure 13 G" for M1 when W-set={aa,bb}

Finally, G[E_g] and the RSA G* of G" over E_g is constructed as shown in Figure 14, where the edges from E" added to G[E_g] are shown in bold lines.



L_{m1} =b@L1a'@a@w2'@b@w1'@a@w2'@b@w1'@a@L1b'@b@w1'@a@w2'@b@w1'@a
L_{m2} =a@w2'@b@w1'@a@w2'@b

Figure 14 G[Eg] and the RSA G* of G" over Eg for M1 when W-set={aa,bb}

An RCPT of G" over E_g is found as follows:

$(v_1,v_2;ba), (v_2,v_1;L_{m1}), (v_1,v_4;ab), (v_4,v_1;L_{m2})$

The final test sequence is shown in Table 12, whose length = $|L_{m1}|+|L_{m2}|+4 = 42$.

For the machine M1 using a W-set, the results can be summarized as follows:

W-set	Lower Bound Without Overlapping	Algorithm-W1	Algorithm-W2	Lower Bound With Overlapping
{aa, ab, b}	100	46	-	43
{aa, bb}	76	42	42	37

Transition	Test Segment
	(1b5)
	(5a2)
(2,4;b/0)	**(2b4)**(4a6)(6a2)
(4,6;a/1)	(4a6)(6a2)(2a1)
(6,2;a/1)	(6a2)(2a1)(1a3)
(2,1;a/0)	(2a1)(1a3)(3a5)
(1,3;a/1)	(1a3)(3a5)(5a2)
(3,5;a/0)	(3a5)(5a2)(2a1)
(5,2;a/0)	(5a2)(2a1)(1a3)
(1,3;a/1)	(1a3)(3b4)(4b1)
(4,1;b/1)	(4b1)(1a3)(3a5)
(3,5;a/0)	(3a5)(5b6)(6b3)
(6,3;b/1)	(6b3)(3a5)(5a2)
(5,2;a/0)	(5a2)(2b4)(4b1)
(2,4;b/0)	(2b4)(4b1)(1b5)
(4,1;b/1)	(4b1)(1b5)(5b6)
(1,5;b/1)	(1b5)(5b6)(6b3)
(5,6;b/0)	(5b6)(6b3)(3b4)
(6,3;b/1)	(6b3)(3b4)(4b1)
(3,4;b/1)	(3b4)(4b1)(1b5)
(1,5;b/1)	(1b5)(5a2)(2a1)
(2,1;a/0)	(2a1)(1b5)(5b6)
(5,6;b/0)	(5b6)(6a2)(2a1)
	(1a3)
	(3b4)
(4,6;a/1)	**(4a6)**(6b3)(3b4)
(3,4;b/1)	(3b4)(4a6)(6a2)
(6,2;a/1)	(6a2)(2b4)(4b1)

(Note: refer to Table 10 for the notation.)

Test Sequence = **b/1, a/0,** b/0, a/1, a/1, a/0, a/1, a/0, a/0, a/0, a/1, b/1, b/1, a/1,
a/0, b/0, b/1, a/0, a/0, b/0, b/1, b/1, b/0, b/1, b/1, b/1, b/1, a/0,
a/0, b/1, b/0, a/1, a/0, **a/1, b/1,** a/1, b/1, b/1, a/1, a/1, b/0, b/1.

(Transfer sequences are bold-faced in the test sequence.)

Test Sequence Length = 42

Table 12 A Test Sequence Generated by Algorithm-W2 for M1 when W-set={aa, bb}

5. CONCLUSIONS

We have studied the problem of the optimization of the length of test sequences for FSM-based protocol conformance testing. This study focuses on finding the extent to which the length of a test sequence for an FSM can be optimized, and obtaining algorithms to optimize the length of a test sequence for an FSM.

For a special case (i.e., for a class of FSM's), we give the greatest lower bound (GLB) on the length of test sequences generated by any method employing a DS (D-method) without overlapping test segments. For the general case, we give a lower bound (i.e., an approximation to the GLB) on the length of test sequences generated by any D-method without overlapping test segments. Based on this lower bound for the general case, we establish a lower bound (i.e., an approximation to the GLB) on the length of test sequences generated by any D-method that overlaps test segments.

It is observed that the reduction in the length of test sequences generated by any D-method due to overlapping is significant. Thus. we propose the first type of algorithm (i.e., Algorithm-D1) for a method employing a DS that overlaps test segments, which utilizes the maximum-cardinality minimum-cost matching algorithm. The time complexity of Algorithm-D1 is polynomial.

However, some deficiencies in using the maximum-cardinality minimum-cost matching algorithm in optimizing test sequences are identified. To overcome these deficiencies, we propose the second type of algorithm (i.e., Algorithm-D2) for a method employing a DS that overlaps test segments, which utilizes the rural Chinese postman tour algorithm. We also discuss the sufficiency conditions for Algorithm-D2 to find a minimum-length test sequence in polynomial-time, and the sufficiency conditions for Algorithm-D2 to achieve the optimal test sequence for an FSM.

Considering any method employing a W-set (W-method), we give a lower bound (i.e., an approximation to the GLB) on the length of test sequences generated by any W-method without overlapping test segments in the general case. Based on this lower bound, we establish a lower bound (i.e., an approximation to the GLB) on the length of test sequences generated by any W-method that overlaps test segments.

Also, it is observed that the reduction in the length of test sequences generated by any W-method due to overlapping is significant. Thus, we propose the first type of algorithm (i.e., Algorithm-W1) for a method employing a W-set that overlaps test segments, which utilizes the maximum-cardinality minimum-cost matching algorithm. The time complexity of Algorithm-W1 is polynomial.

The same deficiencies exist in Algorithm-W1 as that in Algorithm-D1. To overcome these deficiencies, we propose the second type of algorithm (i.e., Algorithm-W2) for a method employing a W-set that overlaps test segments, which utilizes the rural Chinese postman tour algorithm. We then discuss the sufficiency conditions for Algorithm-W2 to find a minimum-length test sequence in polynomial-time, and the sufficiency conditions for Algorithm-W2 to achieve the optimal test sequence for an FSM.

Algorithm-D2 and -W2 can easily be incorporated into TSG (which is a FSM-based test sequence generator developed in the University of Ottawa) [URA91a]. It must be noted that these algorithms which utilize the rural Chinese postman tour algorithm (i.e., Algorithm-D2 and -W2), are discussed only for limited formats of DS's or W-sets. Further research is needed to extend these algorithms for the general format of a DS or a W-set. Also, more research is required to investigate the possible usage of the LB's in comparing different algorithms.

REFERENCES

[AHO88] A.V. Aho, A.T. Dahbura, D. Lee, and M.U. Uyar, "An optimization technique for protocol conformance test sequence generation based on UIO sequences and rural Chinese postman tours," in *Protocol Specification, Testing, and Verification, XIII*, S. Aggarwal and K. Sabnani, eds., Amsterdam: North-Holland, pp.75-86, 1988.

[BOC82] G.v. Bochmann and C.A. Sunshine, "A survey of formal methods," in *Computer Networks and Protocols*, P.E. Green, ed., New York: Plenum Press, pp.561-578, 1982.

[BOY91] S.C. Boyd and H. Ural, "On the complexity of generating optimal test sequences," *IEEE Trans. on Software Engineering*, vol.SE-17, pp.976-978, Sep. 1991.

[CHA89] W.Y.L. Chan, S.T. Vuong and M.R. Ito, "An improved protocol test generation procedure based on UIOs," in *Proc. ACM SIGCOMM'89 Symposium: Communication Architectures and Protocols*, pp.283-292, 1989.

[CHE90] M.S. Chen, Y. Choi, and A. Kershenbaum, "Approaches utilizing segment overlap to minimize test sequences," in *Protocol Specification, Testing and Verification, X*, L. Logrippo, R.L. Probert and H. Ural, eds., Amsterdam: North-Holland, pp.85-98, 1990.

[CHO78] T. Chow, "Testing software design modeled by finite state machines," *IEEE Trans. on Software Engineering*, vol.SE-4, pp.178-187, March 1978.

[DAH90] A.T. Dahbura, K.K. Sabnani, and M.U. Uyar, "Formal methods for generating protocol conformance test sequences," *Proceedings of the IEEE*, vol.78, no.8, pp.1317-1325, August 1990.

[EDM73] J. Edmonds and E.L. Johnson, "Matching, Euler tours and the Chinese postman," *Mathematical Programming*, vol.5, pp.88-124, 1973.

[FUJ91] S. Fujiwara, G.v. Bochmann, F. Khendek, M. Amalou, and A. Ghedamsi, "Test selection based on finite state models," *IEEE Trans. on Software Engineering*, vol.SE-17, pp.591-603, June 1991.

[GIL62] A. Gill, *Introduction to the Theory of Finite-State Machines*, New York: McGraw-Hill, 1962.

[GON70] G. Gonenc, "A method for the design of fault detection experiments," *IEEE trans. on Computer*, vol. C-19, pp.551-558, June 1970.

[KOH78] Z. Kohavi, *Switching and Finite Automata Theory*, New York: McGraw-Hill, 1978.

[KUA62] M.K. Kuan, "Graphic programming using odd or even points," *Chinese Math.*, vol.1, pp.273-277, 1962.

[LEN76] J.K. Lenstra and A.H.G. Rinnooy Kan, "On General Routing Problems," *Networks*, vol.6, pp.273-280, 1976.

[MIL90] R.E. Miller and S. Paul, "Generating minimal length test sequences for conformance testing of communication protocols," submmitted for publication, 1990.

[NAI81] S. Naito and M. Tsunoyama, "Fault detection for sequential machines by transition tours," in *Proc. 11th IEEE Fault Tolerant Comput. Symp.*, IEEE Computer Soc. Press, pp.238-243, 1981.

[PAP82] C. Papadimitriou and K. Steiglitz, *Combinatorial Optimization Algorithms and Complexity*, Englewood Cliffs, N.J.: Prentice-Hall, 1982.

[RAY87] D. Rayner, "OSI conformance testing," *Computer Networks and ISDN Systems*, vol.14, no.1, pp.79-98, 1987.

[SAB85] K.K. Sabnani and A.T. Dahbura, "A new technique for generating protocol tests," in *Proc. 9th Data Comm. Symp.*, pp.36-43, September 1985.

[SAB88] K.K. Sabnani and A.T. Dahbura, "A protocol test generation procedure," *Computer Networks and ISDN Systems*, vol.15, no.4, pp.285-297, 1988.

[SAR84] B. Sarikaya and G.v. Bochmann, "Synchronization and specification issues in protocol testing," *IEEE Trans. on Communication*, vol.COM-32, pp.389-395, April 1984.

[SAR87] B. Sarikaya, G.v. Bochmann, and E. Cerny, "A test design methodology for protocol testing," *IEEE trans. on Software Engineering*, vol.SE-13, pp.518-531, May 1991.

[SHE89] Y.N. Shen, F. Lombardi, and A.T. Dahbura, "Protocol conformance testing using multiple UIO sequences," in *Protocol Specification, Testing* *and Verification, IX*, E. Brinksma, G. Scollo, and C.A. Vissers, eds., Amsterdam: North-Holland, pp.131-143, 1989.

[SID89a] D.P. Sidhu and T.K. Leung, "Formal methods for protocol testing: a detailed study," *IEEE Trans. on Software Engineering*, vol.SE-15, no.4, pp.413-426, 1989.

[SID89b] D.P. Sidhu and R. Vallurupalli, "On arbitrariness in protocol conformance test generation," submmitted for publication, 1989.

[SID90a] D.P. Sidhu, "Protocol testing: the first ten years, the next ten years," in *Protocol Specification, Testing and Verification, X*, L. Logrippo, R.L. Probert and H. Ural, eds., Amsterdam: North-Holland, pp.47-68, 1990.

[SID90b] D.P. Sidhu and A. Chung, "On sufficient conditions for an efficient protocol conformance test generation technique based on Rural Chinese Postman Problem," submmitted for publication, 1990.

[URA90] H. Ural and Y. Lu, "An improved method for test sequence generation," Technical Report, TR-90-12, Department of Computer Science, University of Ottawa, March 1990.

[URA91a] H. Ural, "Formal methods for test sequence generation," to appear in *Computer Communications*, 1991.

[URA91b] H. Ural and B. Yang, "A test sequence selection method for protocol testing," *IEEE trans. on Communications*, vol.COM-39, pp.514-523, April 1991.

[UYA86] M.U. Uyar and A.T. Dahbura, "Optimal test sequence generation for protocols: the Chinese postman algorithm applied to Q.931," in *Proc. 1986 IEEE Global Telecommunications Conf.*, pp.68-72, 1986.

[YAN90] B. Yang and H. Ural, "Protocol conformance test generation using multiple UIO sequences with overlapping," in *Proc. ACM SIGCOMM'90*, pp.118-125, 1990.