INFORMATION TO USERS

This manuscript has been reproduced from the microfilm master. UMI films the text directly from the original or copy submitted. Thus, some thesis and dissertation copies are in typewriter face, while others may be from any type of computer printer.

The quality of this reproduction is dependent upon the quality of the copy submitted. Broken or indistinct print, colored or poor quality illustrations and photographs, print bleedthrough, substandard margins, and improper alignment can adversely affect reproduction.

In the unlikely event that the author did not send UMI a complete manuscript and there are missing pages, these will be noted. Also, if unauthorized copyright material had to be removed, a note will indicate the deletion.

Oversize materials (e.g., maps, drawings, charts) are reproduced by sectioning the original, beginning at the upper left-hand corner and continuing from left to right in equal sections with small overlaps.

Photographs included in the original manuscript have been reproduced xerographically in this copy. Higher quality 6" x 9" black and white photographic prints are available for any photographs or illustrations appearing in this copy for an additional charge. Contact UMI directly to order.

Bell & Howell Information and Learning
300 North Zeeb Road, Ann Arbor, MI 48106-1346 USA
800-521-0600

UMI®

Université d'Ottawa • University of Ottawa

Solutions to Controllability and Observability Problems in Distributed Testing

David Whittier

A 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 Masters in Computer Science.**

School of Information Technology and Engineering
University of Ottawa
Ottawa, Ontario

---

* The Masters program in Computer Science is a joint program with Carleton University, administered by the Ottawa-Carleton Institute for Computer Science

Abstract

The objective of protocol testing is to confirm that a protocol implementation under test conforms to its specification. However, in protocol test architectures that utilize multiple remote testers in a distributed environment, this objective can be complicated by the fact that testers may encounter problems relating to controllability and observability during the application of a test sequence. Solutions in literature to these problems usually involve first generating a test sequence from the specification of an implementation under test, then inserting coordination messages or appending selected test subsequences that will allow the testers to solve the controllability and observability problems.

This thesis proposes a method that uses a set of transformation rules to construct a directed graph from a given specification. A transition tour of this directed graph based on a rural Chinese postman tour will result in a test sequence with no potential for controllability or observability problems, and where the use of coordination messages is either minimized or, if possible, avoided altogether.

Acknowledgements

I would like to acknowledge my supervisor, Dr. Hasan Ural, for introducing me to the field of protocol testing and, in particular, the challenge of controllability and observability problems. My thanks to him for his guidance, patience and support.

My studies would have never been possible had it not been for the sponsorship and support of the Department of National Defence and the Subsidized University Education Program.

To Maya, Taro and Shoji

Thanks for being there.

Table of Contents

List of Figures

List of Tables

Chapter One

Introduction

1.1 Background

The objective of protocol testing is to confirm that a protocol Implementation Under Test (IUT) conforms to its specification. [II95] outlines a number of Abstract Test Methods, referred to in this thesis as Abstract Test Architectures, in which an IUT can be tested, and an abstract test case can be specified. Open Distributed Processing (ODP) [IS95] specifies a generic framework for building open distributed systems, and considerable effort has been spent with a view to defining test architectures for ODP implementations.

The majority of the literature published on the topic of testing refers to an IUT that has only one point of control and observation, otherwise known as a "port," which is accessed by one tester. However, the introduction of multiple ports and multiple testers within what is known as a distributed test architecture introduces the possibility of coordination problems among testers during the application of a test sequence, known as problems of controllability or observability. A controllability problem, also known as a synchronization problem, occurs if a tester cannot determine when to apply a particular input to an IUT. The literature deals with this problem by proposing various methods of constructing a synchronizable test sequence, i.e., a test sequence free of any synchronization problems. An observability problem occurs if a tester cannot determine whether a particular output from an IUT was generated in response to the related input.

Solutions to the observability problem vary depending on whether the distributed test architecture supports indirect communication, i.e., when the testers communicate with each other through their interaction with the IUT [II95], or direct communication, when the testers communicate with each other using external coordination messages over a multicast channel [IS95]. In this thesis we propose a method that will solve the controllability and observability problems within a distributed test architecture with multiple testers.

1.2 Motivation and Objectives of the Thesis

The problem that will be studied is, in its most general form, how to detect and eliminate potential controllability and observability problems that may be encountered while testing an implementation of a given Finite State Machine (FSM) that represents the specification of a communication protocol within the ISO/ODP distributed test architecture. Many researchers have studied the problems of controllability and observability, and a review of these studies is included later in this thesis. Among these studies, the observability problem has received relatively less attention than the controllability problem. Those researchers who have studied the observability problem have proposed solutions where first a synchronizable test sequence is generated by applying some test construction method to a given FSM. The synchronizable test sequence is then examined for instances of observability problems and corrective action is taken, depending on whether the test architecture being used supports direct communication between testers through the exchange of external coordination messages, or indirect communication between testers through their interaction with the IUT.

Our review of these solutions has led us to study a method that identifies controllability and observability problems before constructing a synchronizable test sequence, in an effort to minimize the amount of corrective action needed for the elimination of these problems. Thus, the objective of this thesis is to propose a method of generating a synchronizable test sequence whereby the length of the test sequence and the number of corrective actions needed to eliminate controllability and observability problems are minimized, and that can be applied in a distributed test architecture regardless of whether the test architecture supports direct or indirect communication between testers.

1.3 Contributions of the Thesis

We propose a method that generates a minimum-length test sequence with no possibility of potential controllability or observability problems when applied to a 2-port FSM in a distributed test architecture. This method can be used to generate test sequences that can be applied to distributed test architectures where testers are able to communicate amongst themselves directly, via external coordination messages and, with a minor augmentation, can be used to generate test sequences that can be applied in distributed test architectures where testers communicate amongst themselves indirectly, via their interactions with the IUT.

We prove that the proposed method ensures the absence of potential controllability and observability problems.

We show the proposed method to perform at least as well as other related methods proposed in literature, and show it to perform better than each one under certain conditions.

Finally, we extend the proposed method so that it can be applied to an n-port FSM in a distributed test architecture, $n > 2$.

1.4 Organization of the Thesis

Chapter 2 outlines the preliminaries needed to describe the proposed method, including the FSM model and the basics of protocol conformance testing. Chapter 3 reviews the existing methods in literature that have been proposed to solve controllability and observability problems. Chapter 4 describes our proposed method and compares it with the existing methods, and Chapter 5 presents our conclusions, with a summary of contributions and directions for future research.

Chapter Two

Preliminaries

2.1 FSM Model and its Graphical Representation

An ***n*-port Finite State Machine (*np*-FSM)** $M = (S, \Sigma, \Gamma, \delta, \lambda, s_0)$ where

- S is a finite set of states of M,
- $s_0 \in S$ is the initial state of M,
- Σ is an n-tuple $(\Sigma_1, \Sigma_2, \ldots, \Sigma_n)$, where Σ_k is the input alphabet of port k, and $\Sigma_i \cap \Sigma_j = \varnothing$ for $i \neq j$, $i, j = 1, 2, \ldots, n$. Let $I = \Sigma_1 \cup \Sigma_2 \cup \ldots \cup \Sigma_n \cup \{-\}$, where - means *null* input,
- Γ is an n-tuple $(\Gamma_1, \Gamma_2, \ldots, \Gamma_n)$, where Γ_k is the output alphabet of port k, and $\Gamma_i \cap \Gamma_j = \varnothing$ for $i \neq j$, $i, j = 1, 2, \ldots, n$. Let $O = \{<a_1, a_2, \ldots, a_n> \mid \exists\, a_i \in \Gamma_i \cup \{-\}, 1 \leq i \leq n\}$, where - means *null* output,
- δ is the transition function that maps $S \times I$ to S, i.e., $\delta: S \times I \rightarrow S$, and
- λ is the output function that maps $S \times I$ to O, i.e., $\lambda: S \times I \rightarrow O$.

An FSM M is **deterministic** if, for each input $x \in I$, there is at most one transition defined at each state of M. An FSM M is said to be **minimal** if none of its states are equivalent (i.e., $\forall\, s_i, s_j \in S$, $s_i \neq s_j$, $\exists$ an input sequence $x \in I^*$ such that $\lambda(s_i, x) \neq \lambda(s_j, x)$). An FSM M is said to be **completely specified** if, for each input $x \in I$, 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 represents the set S of states of M, and a set of directed edges E represents all specified transitions of M. A **transition** of an *np*-FSM M is a 3-tuple $t_{jk} = (s_j, s_k; x/y)$, where $s_j, s_k \in S$, $x \in I$, and $y \in O$ such that $\delta(s_j, x) = s_k$, $\lambda(s_j, x) = y$, and x/y is known as an

input/output pair. Each edge $e_{jk} = (v_j, v_k; x/y) \in E$ represents a state transition from state v_j to state v_k with input x and output y where the input/output pair x/y is the **label** of e_{jk}, denoted by $label(e_{jk})$, v_j is called the **head** of e_{jk}, denoted by $head(e_{jk})$, and v_k is called the **tail** of e_{jk}, denoted by $tail(e_{jk})$.

A **path** $P = (v_1, v_2; x_1/y_1)(v_2, v_3; x_2/y_2)\ldots(v_{k-1}, v_k; x_{k-1}/y_{k-1})$, $k>1$, in $G = (V, E)$ is a finite sequence of adjacent (but not necessarily distinct) edges in G, where v_1 and v_k are $head(P)$ and $tail(P)$, and $x_1/y_1, x_2/y_2, \ldots, x_{k-1}/y_{k-1}$ is $label(P)$. The **cost** or **length** of each edge of G is equal to the number of input/output pairs in its label. The cost of a path (or length of a path) P in G is the sum of the costs (or lengths) of edges included in P.

A digraph $G = (V, E)$ is **strongly connected** if, for any pair of vertices v_j and v_k, there exists a path from v_j to v_k. It is **weakly connected** if its underlying undirected graph is connected. A **tour** of G is a path in G that starts and ends at the same vertex of G. An **Euler tour** of G is a tour that contains every edge of E exactly once. A **postman tour (PT)** of G is a tour that contains every edge in E at least once. A **rural postman tour (RPT)** of G over a set $E_C \subseteq E$ is a tour traversing every edge in E_C at least once. A **Chinese postman tour (CPT)** is a minimum-cost tour that contains every edge in E at least once. A **rural Chinese postman tour (RCPT)** of G over a set $E_C \subseteq E$ is a minimum-cost tour that traverses every edge in E_C at least once. The **edge-induced subgraph** of $G[E_C]$ is the subgraph of G whose vertex set is the set of ends of edges in E_C and whose edge set is E_C. $G[E_C]$ is an edge-induced **spanning** subgraph of G if its vertex set is V.

Given a vertex $v \in V$, the **in-degree** of v, $d_i(v)$, is defined as $|\{(u, v): (u, v) \in E\}|$ and the **out-degree** of v, $d_o(v)$, is defined as $|\{(v, w): (v, w) \in E\}|$. A digraph $G = (V, E)$ is

symmetric if $d_i(v) = d_o(v)$, $\forall\ v \in V$. Given a postman tour P of G, let $\chi(v_j, v_k; x/y) \geq 1$ be the number of times edge $(v_j, v_k; x/y)$ is contained in P. If edge $(v_j, v_k; x/y)$ is replicated $\chi(v_j, v_k; x/y)$ times, a symmetric graph $G^\wedge$, called a **symmetric augmentation** of G, is obtained and P is an Euler tour of $G^\wedge$. A **min-cost symmetric augmentation** of G is achieved by replicating edges of G such that $G^\wedge$ is symmetric and the number of replicated edges are minimized.

2.2 Protocol Conformance Testing

The aim of protocol conformance testing is to determine, by means of testing, if a protocol implementation under test (IUT) conforms to its specification [IS94]. Typically, protocol conformance testing involves constructing a test sequence from the protocol specification, applying the test sequence to the IUT, and analyzing the result of the application of the test sequence to determine whether the IUT conforms to its specification. A test sequence is either a sequence of **test cases** (pairs of test input and expected output) or a collection of subsequences of test cases, in which case it is referred to as a **test suite**. The application of a test sequence is carried out in a test architecture that consists of testers and an IUT. ISO and ODP define a collection of abstract test architectures.

An **Abstract Test Architecture** (ATA), called an **Abstract Test Method** (ATM) in the context of the OSI model, is a description of how an IUT which is in a System Under Test (SUT) is to be tested, given at an appropriate level of abstraction to make the description independent of any particular realization. It must, however, include enough detail to enable abstract test cases to be specified for this test architecture [II95]. An

ATA in the context of a $2p$-FSM is defined in terms of the relationship between the IUT and the Upper and Lower Testers, through:

Points of Control and Observation (PCO), or points within a testing environment where the occurrence of test inputs and actual outputs is to be controlled and observed, as defined in an ATA [II95],

Abstract Service Primitives (ASPs), or implementation independent descriptions of an interaction between a service user and a service provider at a service boundary, as defined in an OSI service definition [II95], and

Protocol Data Units (PDUs), or information that is delivered as a unit among peer entities of a network and that may contain control information, address information, or data.

The rules for cooperation between Upper and Lower Testers are defined in the applicable **Test Coordination Procedures** (TCPs) for each architecture. These ATAs have analogous models in other architectures such as ODP [IS95], [FK95], [RA95]. Within an ATA, an **abstract test suite** (ATS) is defined as a test suite composed of **abstract test cases**, the complete and independent specifications of the actions of the testers required to achieve a specific test purpose.

The **Upper Tester (UT)** is defined as the representation of the means of providing, during test execution, control and observation of the upper service boundary of the IUT, as defined by the chosen ATA. The **Lower Tester (LT)** is defined as the representation of the means of providing, during test execution, indirect control and observation of the lower service boundary of the IUT via the underlying service provider. The ISO ATAs are defined as follows [II95]:

- **Local** – an ATA in which both the Upper and Lower Testers are located within the Test System and there is a PCO at the upper service boundary of the IUT (Figure 1). The local test architecture requires the upper service boundary of the IUT to be a standardized hardware interface. In the ODP model, this ATA is called **centralized**, which refers to one tester.

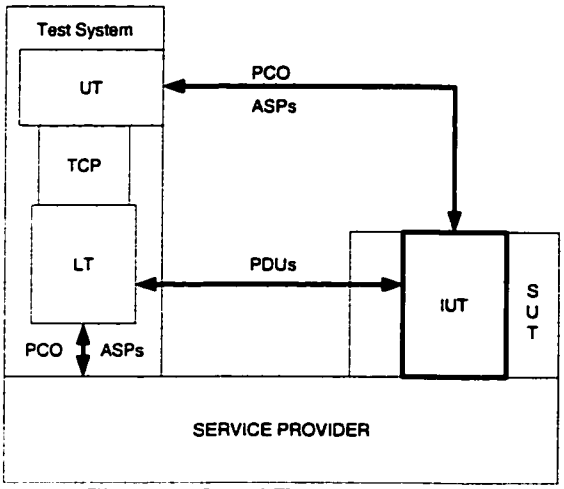

Figure 1. Local Test Architecture

- **Coordinated** – an ATA in which the Upper Tester is within the SUT and for which a standardized Test Management Protocol is defined for the Test Coordination Procedures, enabling the control and observation to be specified solely in terms of the Lower Tester Activity. This includes the control and management of Test Management PDUs (Figure 2).

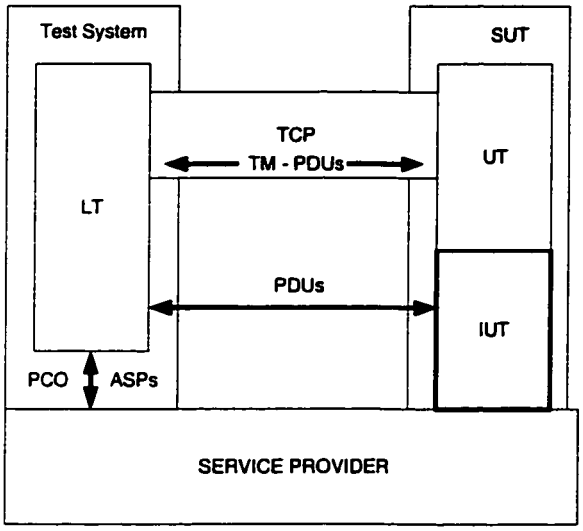

Figure 2. Coordinated Test Architecture

Distributed – an ATA in which the Upper Tester is within the SUT and there is a PCO at the upper service boundary of the IUT (Figure 3). The upper service boundary of the IUT may be either human user interface or standardized programming language interface. While requirements for TCP are specified, procedures are not.

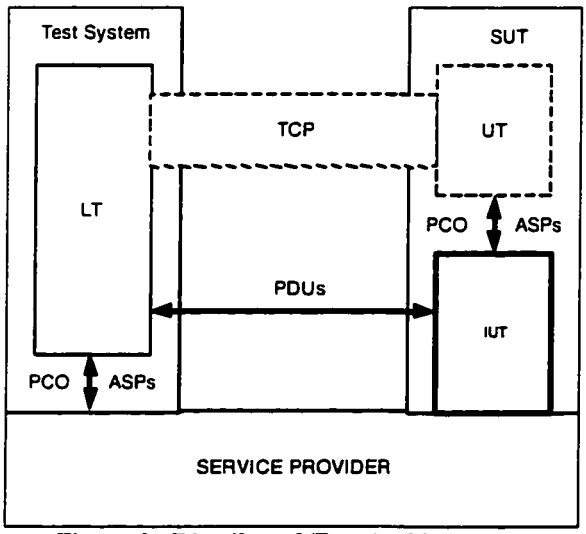

Figure 3. Distributed Test Architecture

Remote – an ATA in which the control and observation of test inputs and outputs is specified solely in terms of Lower Tester activity (Figure 4). Some requirements for Test

Control Procedures may be implied or informally expressed in the Abstract Test Suite, but no assumption is made regarding their feasibility or realization. There is no Upper Tester as such, but some Upper Tester functions may be performed by the SUT.

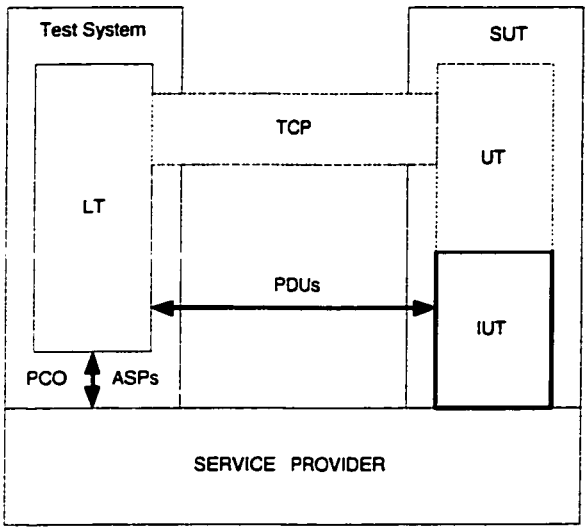

Figure 4. Remote Test Architecture

Among the ATAs listed above, an IUT has the highest observability and controllability in the local test architecture, and the lowest one in the remote test architecture. If a set of detectable faults (S_r, S_d, S_c, S_l) are associated with each test architecture (remote, distributed, coordinated, and local, respectively) the fault detectability in these architectures is shown to be the following [LB94]:

$S_r \subseteq S_d \subseteq S_c \subseteq S_l$

In this thesis, we will consider a sequence of test cases, i.e., a test sequence, rather than a collection of subsequences of test cases, i.e., a test suite, in order to compare our method with the existing methods.

A **global test sequence** ω of an *np*-FSM M is the label of a path of the digraph $G = (V, E)$ representing M, which is of the form $x_1/y_1, x_2/y_2, \ldots, x_m/y_m$, where $x_i \in I$ and

$y_i \in O$, $1 \le i \le m$. Given a global test sequence ω, a **local test sequence** ω_k for port k of an *np*-FSM M is of the form $\alpha_1, \alpha_2, \ldots, \alpha_m$, where α_i $(1 \le i \le m)$ is either:

- x_i/y_i, $x_i \in \Sigma_k \cup \{-\}$, $y_i = <a_1, a_2, \ldots, a_n>$, where $a_k \in \Gamma_k \cup \{-\}$ and $a_j = -$, for $j \ne k$, $1 \le j, k \le n$, or
- an external coordination message reception or transmission to ensure observability or controllability.

Note that $a_k \in \Gamma_k \cup \{-\}$ together with $x_i \in \Sigma_k \cup \{-\}$ allows -/- transitions within a local test sequence ω_k for port k of an *np*-FSM. In an ATA involving multiple testers, each tester executes a local test sequence constructed from the global test sequence of a given FSM M. In the local test sequence, tester k knows only that there are transitions involving itself or others, and doesn't know the inputs or outputs of the other testers.

A **fault model** is defined as a set of models of non-conforming implementations of a given FSM M [IS94]. For a particular specification, two types of fault models may be recognized – fault models with:

- **transfer faults**, i.e., the final state of a tested transition of an implementation may be different from the final state specified by the specification, and
- **output faults**, i.e., the observed output of an implementation after a specified input may not be the one specified by the specification.

A **hybrid fault** is defined as a combination of the above.

Controllability refers to the ease of affecting the specified outputs. **Observability** refers to the ease of determining if specified inputs affect the outputs. **Detectability** refers to the ease of detecting a fault.

As mentioned earlier, some ATAs may cause controllability or observability problems, depending on the ability of the testers to coordinate the application of their respective local test sequences. It is said that a **controllability (synchronization) problem** exists when a tester is required to send an input to the IUT in the current transition, and because it is not **involved** in the previous transition, i.e., it didn't send the input or receive the output in the previous transition, it does not know when to send the input. An **observability problem** exists when a tester is expecting to receive an output from the IUT and, because it did not send the input in the current transition nor did it receive an output in the previous transition, it does not know when to start or stop waiting for the reception of the output of the IUT.

Given an *np*-FSM M and a global test sequence $\omega = x_1/y_1, x_2/y_2, \ldots, x_m/y_m$ of M, where $x_i \in I$ and $y_i \in O$, $1 \leq i \leq m$, a **synchronization problem** occurs when, in the labels x_j/y_j and x_{j+1}/y_{j+1} of any two consecutive transitions, there exists a tester k that sends x_{j+1} that is neither the one sending x_j nor one of those receiving an output belonging to y_j, $1 \leq j \leq m\text{-}1$.

Given an *np*-FSM M and a global test sequence $\omega = x_1/y_1, x_2/y_2, \ldots, x_m/y_m$ of M, where $x_i \in I$ and $y_i \in O$, $1 \leq i \leq m$, any two consecutive transitions t_{ij} and t_{jk} whose labels are x_j/y_j and x_{j+1}/y_{j+1} form a **synchronizable pair** of transitions if t_{jk} can follow t_{ij} without causing a synchronization problem. For a transition $t_{ij} = (v_i, v_j; x_j/y_j)$, each transition $t_{jk} = (v_j, v_k; x_k/y_k)$ that forms a synchronizable pair of transitions with t_{ij} is called an **eligible successor** of t_{ij}. Any (sub)sequence of transitions in which every pair of transitions is synchronizable is called a **synchronizable transition (sub)sequence**. A global test sequence is said to be **synchronizable** if it is the label of a synchronizable transition sequence.

Given an *np*-FSM M and a global test sequence $\omega = x_1/y_1, x_2/y_2, \ldots, x_m/y_m$ of M, where $x_i \in I$ and $y_i \in O$, $1 \leq i \leq m$, a **1-shift output fault** in an implementation N of M exists when, in the labels x_j/y_j and x_{j+1}/y_{j+1} of any two consecutive transitions, there exists one a_k in $y_j = \langle a_1, a_2, \ldots, a_n \rangle$ of M which occurs in $y_{j+1} = \langle a_1, a_2, \ldots, a_n \rangle$ in N (and not in y_j in N) or there exists one a_k in $y_{j+1} = \langle a_1, a_2, \ldots, a_n \rangle$ of M which occurs in $y_j = \langle a_1, a_2, \ldots, a_n \rangle$ in N (and not in y_{j+1} in N), $1 \leq j \leq m\text{-}1$, $1 \leq k \leq m$. An instance of the observability problem manifests itself as an **undetectable 1-shift output fault** if there is a 1-shift output fault related to $a_k \in \Gamma_k$ in any two consecutive transitions whose labels are x_j/y_j and x_{j+1}/y_{j+1}, such that tester k satisfies the condition (a_k is in y_j XOR a_k is in y_{j+1}) AND $x_{j+1} \notin \Sigma_k$. In this case, we say that tester k is **involved** in the shift, and would not be able to detect it.

Chapter Three

Review of Test Coordination Methods

3.1 Synchronizable Test Sequence Generation

Earlier work on test coordination has focused mainly on defining methods for the generation of synchronizable global test sequences. In [SB84], Sarikaya and Bochman were the first team of researchers to formally address the synchronization problem and to propose a method for constructing synchronizable global test sequences from a given FSM. In general terms, each transition or test subsequence that is added to the global test sequence in the course of following a transition tour of the digraph of an FSM is examined in order to see if it is synchronizable with its predecessor. Non-synchronizable transitions or subsequences are discarded, and a different one from the present state is considered. This process is continued until all the transitions of the FSM are covered, or it is decided, through backtracking, that there is no synchronizable global test sequence for the given FSM. It is accepted in [SB84] that this method may result in a synchronizable global test sequence that is not of minimum length. Sarikaya and Bochman also discuss the concept of protocol specifications that are intrinsically nonsynchronizable. A protocol specification is **intrinsically nonsynchronizable** if it contains a transition that does not have an eligible successor. They consider the X.25 DTE protocol as an example of an intrinsically nonsynchronizable protocol, and propose a procedure that would enhance such a protocol for the purpose of testing. This is done

by modifying the state table of the protocol so that a synchronizable global test sequence can be found.

In [BU91], Boyd and Ural give an efficient algorithm to determine the existence of a synchronizable global test sequence for a given FSM. They also analyze the computational complexity involved in constructing a minimum length synchronizable global test sequence from a given $2p$-FSM, and conclude that it would be unlikely that a polynomial time algorithm will be found. Specifically, they define two problems that are based on the following definitions: a digraph $G = (V, E)$ is said to be **order-specified** if for each edge $e_{ij} \in E$, a subset of the edges leaving vertex v_j is specified as eligible successors of e_{ij}, and a path in G is said to be **correctly ordered** if for every consecutive pair of edges $e_{jk} = (v_j, v_k; x_j/y_j)$ and $e_{kl} = (v_k, v_l; x_k/y_k)$, e_{kl} is an eligible successor of e_{jk}. Then, given an order specified digraph $G = (V, E)$ and a specified vertex $v \in V$, the **Ordered Euler Tour Problem** (OETP) is defined as determining whether there exists a correctly ordered Euler tour of G that starts at v. Given an order specified digraph $G = (V, E)$, a specified vertex $v \in V$ and a cost function defined on E, the **Ordered Chinese Postman Problem** is defined as finding a minimum cost correctly ordered postman tour of G that starts at v. Boyd and Ural prove that the OETP and the OCPP are both NP-complete problems. Their work provides justification for heuristic techniques for generating synchronizable global test sequences that are not necessarily of minimum length.

In [UW93], [GU95], and [CU95], the authors present methods for constructing synchronizable global test sequences. In general, these methods involve the modification of a digraph of an FSM such that a tour of the modified graph is a synchronizable global

test sequence. In [UW93], if a given $2p$-FSM M can be represented by an order specified digraph $G = (V, E)$, then a synchronizable global test sequence may be generated from G by constructing a digraph $G' = (V', E')$, identifying test subsequences, adding these test subsequences to G' to form $G^{\wedge} = (V^{\wedge}, E^{\wedge})$ and finding an RPT in $G^{\wedge}$ over the set of edges representing the test subsequences. [GU95] uses the concept of a correctly ordered digraph and a given set of test subsequences to construct a synchronizable global test sequence. This sequence can then be minimized by the elimination of redundant subsequences and judicious choice of transfer sequences and test subsequences. [CU95] also constructs a digraph $G' = (V', E')$ by adding the set E_C of all test subsequences in a given FSM M to $G = (V, E)$, and finding an RCPT of G' over E_C. Chen and Ural show that if the edge induced spanning subgraph $G[E_C]$ of G' is weakly connected, then finding an RCPT of G' over E_C can be achieved in polynomial time. This is justified as follows: in [AA88], Aho et al utilize the **Rural Chinese Postman Problem** to find a rural postman tour of a digraph with minimum cost. Although computing such a tour is known to be NP-complete, the authors present a class of graphs for which a minimum cost global test sequence can be constructed in polynomial time. Specifically, they prove that if an edge-induced subgraph $G[E']$ of a digraph $G = (V, E)$, $E' \subset E$, is a weakly connected spanning subgraph of G, then any rural symmetric augmentation graph $G^{\wedge}$ of G is strongly connected and therefore has an Euler tour. When $G[E']$ is weakly connected, an RCPT over E' can therefore be found in polynomial time.

[BU91], [UW93], and [GU95] point out that not all FSMs will yield synchronizable global test sequences using their methods, as they make the assumption that the testers

may only communicate with each other through the IUT. As well, it is recognized that some FSMs are intrinsically nonsynchronizable. [CU95] supports the use of an additional communication channel that allows the exchange of external coordination messages amongst the testers, and includes the cost of those messages in their minimization algorithm. The use of external coordination messages between testers hence can be gainfully used to construct a synchronizable global test sequence from any FSM, including the ones that are intrinsically nonsynchronizable.

In [TY98], Tai and Young introduce a definition of synchronizable that is slightly different than the one presented in this thesis and the other earlier work in the literature, a definition that depends on the current state and eligible transitions of an IUT as well as the transitions in the local test sequences. Specifically, a global test sequence for an FSM M is said by Tai and Young to be synchronizable if any execution of M and the testers according to the global test sequence is deterministic (i.e., the current state of M has at most one eligible transition). The synchronization problem as it is defined in this thesis is referred to as the **pair-wise synchronization problem** by Tai and Young. [TY98] considers the testing strategy of port-based testing, which allows testers to communicate with each other only indirectly via their interactions with the IUT. Tai and Young also expand on the idea of port-based testing by introducing the concept of group-based testing, which divides the ports of an IUT into groups and allows the testers for ports in the same group to communicate with each other directly using external coordination messages.

3.2 Observability Problems in Global Test Sequences

Although some research has been done into the issue of observability problems in distributed testing, it has received much less attention than the problem of synchronization. The methods proposed by [CR99] and [LB94], described in detail in the next two sections, involve generating a global test sequence from the original digraph of an FSM, and inserting corrective actions into the global test sequence to ensure that it is free from controllability and observability problems. The most significant difference between the two methods is in their treatment of the interaction between testers. In [LB94], direct communication between the testers is not supported, and testers must communicate indirectly through the IUT. In [CR99], direct communication is allowed through the use of a multi-cast channel connecting the testers to each other, which allows for the exchange of external coordination messages specifically related either to observability or controllability.

In [YT98], Young and Tai describe three types of implementation faults that cause incorrect test observations. Given an FSM M, an implementation N of M, and a global test sequence ω, N is said to have an **input exchange fault** with respect to ω if, for two transitions t_{ij} and t_{kl} in ω, N is modified from M by exchanging the input symbols of t_{ij} and t_{kl}. N is said to have a **forward output shifting fault** with respect to ω if N is modified from M by removing an output symbol from t_{ij} and adding it to t_{kl}, provided the input symbol of t_{kl} is associated with a different port than the port of the output being shifted. N is said to have a **backward output shifting fault** with respect to ω if N is modified from M by removing an output symbol from t_{kl} and adding it to t_{ij}, provided the input symbol of t_{kl} is associated with a different port than the port of the output being shifted. They

propose a strategy for solving the observability problem as follows: given a global test sequence consisting of transitions $t1$, $t2$, $t3$, ..., tn for a given FSM M and the implementation N of M, first test N by using $t1$. If this test is successful, test N using $t1$, $t2$, then $t1$, $t2$, $t3$, and so on. The intent of this strategy is to validate the transitions in the global test sequence one at a time in the given order, and assumes a test architecture whereby the testers communicate amongst themselves indirectly via their interactions with the IUT.

In [HR00], Hieron proposes a method of augmenting the digraph G of an FSM M by adding edges that represent test subsequences, then finding a minimal length tour of the augmented digraph that contains each test subsequence. Hieron also discusses the concept of forward output-shifting faults and backward output-shifting faults, and proposes two types of observability-related coordination messages that can be used to detect these faults. Given two transitions t_{ij} and t_{jk}, a **post-transition framing message** is sent to the tester sending the input to the IUT in t_{jk} from each tester receiving an output from the IUT in t_{ij}, in order to preclude any possibility of a forward output shifting fault from t_{ij} to t_{jk}. If a tester is receiving an output in t_{ij} and sending an input in t_{jk}, no post-transition framing message for that tester is required. Given two transitions t_{ij} and t_{jk}, a **pre-transition framing message** is sent from the tester sending the input to the IUT in t_{jk} to each tester receiving an output from the IUT in t_{jk}, in order to preclude any possibility of a backward output-shifting fault from t_{jk} to t_{ij}. Hieron's method assumes a test architecture whereby testers may communicate directly amongst themselves using external coordination messages.

3.3 General Distributed Test Architecture

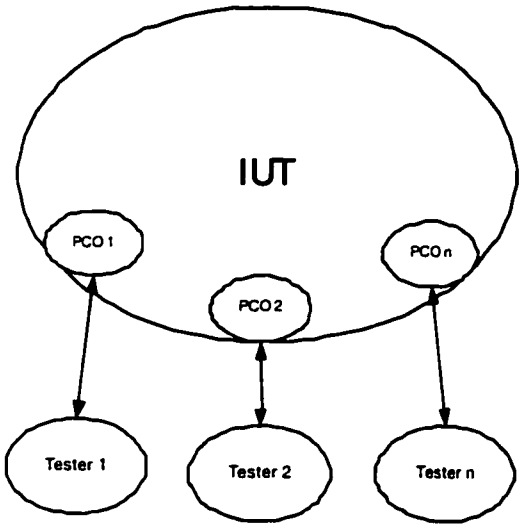

Figure 5. General Distributed Test Architecture

In the general distributed test architecture depicted in Figure 5, there are several testers located at different sites within the test system. However, testers communicate and synchronize with one another indirectly, via their interactions with the IUT. When $n = 2$, this model is reduced to the ISO Distributed Test Architecture.

Because no external coordination messages are possible in this architecture, [LB94] solves the problem of controllability by constructing a synchronizable global test sequence for an FSM as follows:

- for each transition t to be added to the global test sequence constructed so far, check to see if there is a synchronization problem with the previous transition of the sequence.
- if no, add t to the global test sequence and carry on.
- if yes, consider a different transition from the current state.
- if problem still exists, backtrack to the previous state and carry on.

Note that, given a transition, there may not exist any synchronizable sequence that can force the machine to traverse this transition. If this is the case, stop.

Consider the $2p$-FSM $M1$ in Figure 6. By applying the above steps, we can construct a synchronizable global test sequence:

Π = *label*($t1$, $t3$, $t2$, $t4$), i.e., 0/<b, a>, 1/<-, c>, 1/<b, ->, 0/<b, ->, which solves the controllability problem.

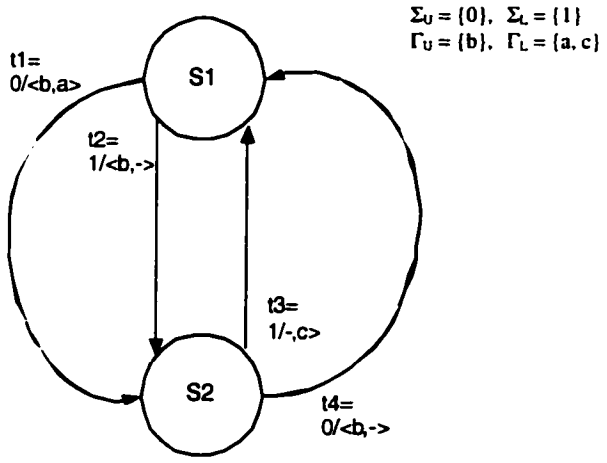

Figure 6. Digraph $G = (V, E)$ of $2p$-FSM $M1$

The observability problem is defined by [LB94] as follows: for two adjacent transitions t_{jk} and t_{kl} in a given specification M, a faulty implementation N can be obtained from M by removing an output from one of these two transitions and adding the output to the other transition. [LB94] calls this class of output faults **output shifting faults** (referred to in this thesis as 1-shift output faults), and proposes an approach that adds specific transitions to the synchronizable global test sequence, with the intent of detecting all 1-shift output faults during the application of the sequence in the distributed test architecture. [LB94] proposes the following algorithm to accomplish this:

- generate a synchronizable global test sequence Π by using one of the test generation methods for FSMs,

- find a set Ω of all consecutive transition pairs along the path caused by applying Π, where each pair may have a potential undetectable 1-shift output fault,
- if Ω is empty, stop. Otherwise, add a set of additional synchronizable test subsequences to Π such that Π can ensure the absence of potential undetectable 1-shift output faults in the transition pairs of Π.

Figure 7 shows a faulty implementation of the 2*p*-FSM *M*1 in Figure 6, where a 1-shift output fault has caused the output *b* intended for tester 1 in transition *t*1 to shift to transition *t*3. This fault is not detectable using the synchronizable global test sequence Π = 0/<*b*, *a*>, 1/<-, *c*>, 1/<*b*, ->, 0/<*b*, ->. This is because the Upper Tester only knows that its second interaction with the IUT after sending the input "0" is the receipt of the output "*b*", and has no way of knowing whether that output was generated by transition *t*1 (as is required by the specification) or transition *t*3 (as occurs in the faulty implementation). In either case, the Upper Tester will not detect the fault. As the Lower Tester is not able to observe the output "*b*", it will also not detect the fault

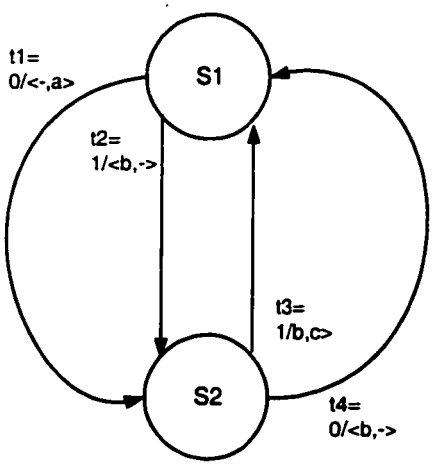

Figure 7. Faulty Implementation of 2*p*-FSM *M*1

To detect the fault in the example shown in Figure 7, and any others, the synchronizable global test sequence Π is examined for instances of potential undetectable

1-shift output faults. In this example there are two such faults: the potential shift of output *b* from *t*1 to *t*3, and the potential shift of output *b* from *t*2 to *t*3. In each instance, additional synchronizable test subsequences are appended to Π:

- adding 0/<*b*, *a*>, 0/<*b*, -> will detect a 1-shift output fault from *t*1 to *t*3. This is because the Upper Tester will not send the input "0" in transition 0/<*b*, -> until it receives the output "*b*" in transition 0/<*b*, *a*>. If that output has shifted, the Upper Tester will therefore detect the shift.
- adding 0/<*b*, *a*>,1/<-, *c*> will detect a 1-shift output fault from *t*2 to *t*3. This is because the Upper Tester is not expecting the second output "b" in transition 1/<-, *c*> from the IUT after it sends the input "0". If that output has shifted, and given that 1/<-, *c*> is the last transition in the appended synchronizable global test sequence, the Upper Tester will therefore detect the shift.

The addition of the synchronizable test subsequences results in a new synchronizable global test sequence of:

0/<*b*, *a*>, 1/<-, *c*>, 1/<*b*, ->, 0/<*b*, ->, 0/<*b*, *a*>, 0/<*b*, ->, 0/<*b*, *a*>, 1/<-, *c*>,

in which the potential undetectable 1-shift output faults in the original synchronizable global test sequence are rendered detectable. Note that appending additional test subsequences has resulted in three additional potential undetectable 1-shift output faults. Specifically, there is the potential for the output "*a*" to shift from 0/<*b*, *a*> to 0/<*b*, -> undetected, as shown below:

0/<*b*, *a*>, 1/<-, *c*>, 1/<*b*, ->, 0/<*b*, ->, 0/<*b*, *a*>, 0/<*b*, ->, 0/<*b*, *a*>, 1/<-, *c*>.

However, the appended transitions have been selected in such a manner that these additional potential undetectable 1-shift output faults are rendered detectable by the

transition pair 0/<*b*, *a*>, 1/<-, *c*> in the original synchronizable global test sequence. This situation will be examined in more detail in Section 4.4.

3.4 ODP Distributed Test Architecture

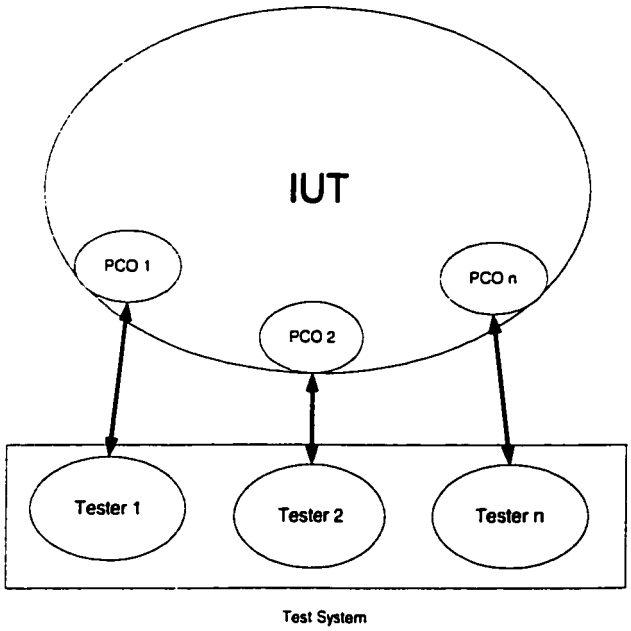

Figure 8. ODP Distributed Test Architecture

The ODP distributed test architecture, depicted in Figure 8, involves several testers located at different sites within the test system. Direct communication between testers is supported by a multicast channel.

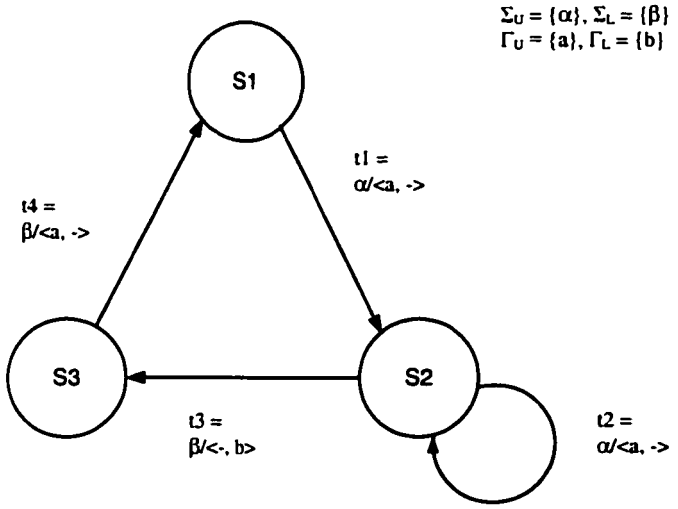

Figure 9. Digraph $G = (V, E)$ of 2*p*-FSM *M*2

In [CR99], the controllability (synchronization) problem is demonstrated in the $2p$-FSM $M2$ of Figure 9. In this example, the label ($\alpha/<a, ->$, $\alpha/<a, ->$, $\beta/<-, b>$, $\beta/<a, ->$) of the transition tour TT = $t1$, $t2$, $t3$, $t4$ is a global test sequence of $M2$. However, a synchronization problem exists between transitions $t2$ and $t3$. Specifically, in transition $t3$, the Lower Tester is required to send an input β to the IUT, but is not involved in transition $t2$, and therefore does not know when to send its input. It is important to note that since $t3$ causes a synchronization problem whether it follows $t1$ or $t2$, there is no synchronizable global test sequence for this FSM without external coordination between testers.

Recall from Section 2.2 that, given an np-FSM M and a global test sequence $\omega = x_1/y_1$, x_2/y_2, ..., x_m/y_m of M, where $x_i \in I$ and $y_i \in O$, $1 \leq i \leq m$, a **synchronization problem** occurs when in the labels x_j/y_j and x_{j+1}/y_{j+1} of any two consecutive transitions, there exists a tester k that sends x_{j+1} that is neither the one sending x_j nor one of those receiving an output belonging to y_j, $1 \leq j \leq m$-1. In the algorithm proposed by [CR99] to eliminate controllability problems, tester h can be the tester that receives an output from the IUT in y_j or, in the case where there exists no output from the IUT in y_j, tester h is the tester that sends an input to the IUT in x_j. For the purposes of this explanation we will assume tester h is the tester sending an input to the IUT in x_j.

The solution proposed by [CR99] to the synchronization problem, one that is reflected at least in some degree in the solutions of [BM99] and [HR00], is to insert the external coordination message exchange relating to controllability "$<-C_k, +C_h>$" between x_j/y_j and x_{j+1}/y_{j+1} in the global test sequence, which will result in the insertion of:

- "$-C_k$" in the local test sequence of tester h just after sending the input x_j to the IUT to indicate that it is to send an external coordination message to tester k, and
- "$+C_h$" in the local test sequence of tester k just before sending the input x_{j+1} to the IUT, to indicate that it is to receive an external coordination message from tester h telling tester k it is time to send its input to the IUT.

The synchronization problem that occurs between transitions $\alpha/\langle a, ->$ and $\beta/\langle -, b\rangle$ can be solved by inserting an external coordination message exchange $\langle -C_L, +C_U\rangle$ after $\alpha/\langle a, ->$ and before $\beta/\langle -, b\rangle$ in the global test sequence. Thus, the global test sequence ω of $M2$, which is:

$\alpha/\langle a, ->, \alpha/\langle a, ->, \beta/\langle -, b\rangle, \beta/\langle a, ->$, becomes

$\alpha/\langle a, ->, \alpha/\langle a, ->, \langle -C_L, +C_U\rangle, \beta/\langle -, b\rangle, \beta/\langle a, ->,$

which can be used to generate the following two local test sequences for the Upper and Lower Tester respectively:

- $\omega_U = \alpha/\langle a, ->, \alpha/\langle a, ->, -C_L, -/\langle -, ->, -/\langle a, ->$
- $\omega_L = -/\langle -, ->, -/\langle -, ->, +C_U, \beta/\langle -, b\rangle, \beta/\langle -, ->$

The problem of observability is demonstrated as follows. Given the digraph $G = (V, E)$ of the $2p$-FSM $M2$ shown in Figure 9, consider a faulty implementation of $M2$ shown in Figure 10. In this faulty implementation, note that the output message "a" intended for the Upper Tester in transition $t4$ has now been shifted to transition $t3$. This 1-shift output fault is not detectable by the application of the two local test sequences derived above, since the Upper Tester receives the proper output from the IUT but has no way of knowing that this output has been generated by the wrong transition.

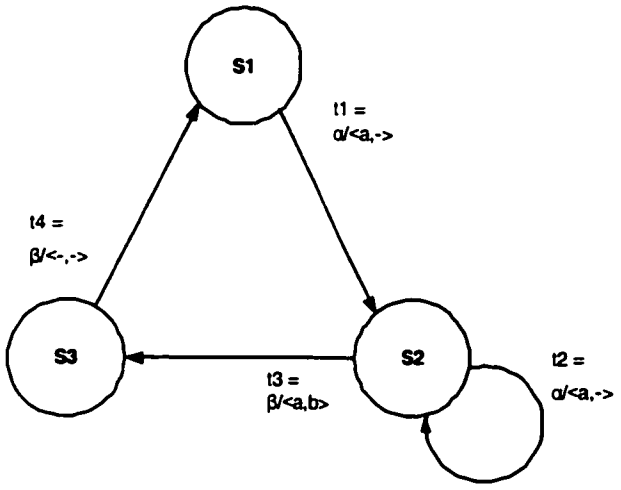

Figure 10. Faulty Implementation of 2*p*-FSM *M2*

In order to ensure that 1-shift output faults are detected, [CR99] proposes a solution to the observability problem that includes the use of the external coordination message "*O*".

Recall from Section 2.2 that, given an *np*-FSM *M* and a global test sequence $\omega = x_1/y_1, x_2/y_2, \ldots, x_m/y_m$ of *M*, where $x_i \in I$ and $y_i \in O$, $1 \leq i \leq m$, a **1-shift output fault** in an implementation *N* of *M* exists when, in the labels x_j/y_j and x_{j+1}/y_{j+1} of any two consecutive transitions, there exists one a_k in $y_j = \langle a_1, a_2, \ldots, a_n \rangle$ of *M* which occurs in $y_{j+1} = \langle a_1, a_2, \ldots, a_n \rangle$ in *N* (and not in y_j in *N*), or there exists one a_k in $y_{j+1} = \langle a_1, a_2, \ldots, a_n \rangle$ of *M* which occurs in $y_j = \langle a_1, a_2, \ldots, a_n \rangle$ in *N* (and not in y_{j+1} in *N*), $1 \leq j \leq m\text{-}1$, $1 \leq k \leq m$. An instance of the observability problem manifests itself as an **undetectable 1-shift output fault** if there is a 1-shift output fault related to $a_k \in \Gamma_k$ in any two consecutive transitions whose labels are x_j/y_j and x_{j+1}/y_{j+1}, such that tester *k* satisfies the condition (a_k is in y_j XOR a_k is in y_{j+1}) AND $x_{j+1} \notin \Sigma_k$. In this case, we say that tester *k* is **involved** in the shift, and would not be able to detect it.

To eliminate the observability problem (i.e., to render the 1-shift output fault detectable), [CR99] proposes the insertion of an external coordination message exchange

relating to observability "<-O_k, +O_h>" between x_j/y_j and x_{j+1}/y_{j+1} in the global test sequence which will result in the insertion of (In the following cases, tester k refers to the tester involved in the shift, and tester h refers to the tester sending the input to the IUT in x_{j+1}):

- Case 1: ($a_k \in y_{j+1}$) an external coordination message "+O_h" in the local test sequence for tester k just before receiving the output a_k from the IUT, sent by tester h. Tester h will have a message "-O_k" inserted in its local test sequence just before sending the input x_{j+1} to the IUT. This message exchange has the effect of tester h telling tester k: "prepare to receive an output from the IUT."
- Case 2: ($a_k \in y_j$) an external coordination message "+O_h" in the local test sequence for tester k just after receiving the output a_k from the IUT, sent by tester h. Tester h will have a message "-O_k" inserted in its local test sequence just before sending the input x_{j+1} to the IUT. This message exchange has the effect of tester h telling tester k: "you should have received an output from the IUT by now."

The observability problem demonstrated in Figure 10 can be detected by inserting the external coordination message exchange relating to observability <-O_U, +O_L> between β/<-, b> and β/<a, -> in the global test sequence to detect the potential shift of the output "a". Similarly, an external coordination message exchange relating to observability <-O_U, +O_L> is inserted between α/<a, -> and β/<-, b> in the global test sequence to detect the potential shift of output "a". Thus, the synchronizable global test sequence ω of $M2$, which is:

α/<a, ->, α/<a, ->, <-C_L, +C_U>, β/<-, b>, β/<a, ->, with the insertion of the external coordination message exchange relating to observability, becomes:

$\alpha/\langle a, ->, \alpha/\langle a, ->, \langle -C_L, +C_U \rangle, \langle -O_U, +O_L \rangle, \beta/\langle -, b \rangle, \langle -O_U, +O_L \rangle, \beta/\langle a, ->$, which can then be used to generate the two local test sequences for the Upper and Lower testers, respectively:

- $\omega_U = \alpha/\langle a, ->, \alpha/\langle a, ->, -C_L, +O_L, -/\langle -, ->, +O_L, -/\langle a, ->$
- $\omega_L = -/\langle -, ->, -/\langle -, ->, +C_U, -O_U, \beta/\langle -, b \rangle, -O_U, \beta/\langle -, ->$

Chapter Four

The Proposed Method

4.1 Motivation

The methods described in [CR99], [LB94], and [YT98] examine a given global test sequence for instances of potential controllability and observability problems and take corrective measures in an attempt to eliminate them in the given sequence. As a result, the synchronizable global test sequence generated may or may not be minimal, either in terms of the total length of the test sequence or in the number of external coordination messages that are used.

We propose a method that takes into consideration both external coordination and input/output costs and consists of a set of transformation rules that construct a modified digraph from the specification of a given *np*-FSM M, allowing for an RCPT of this graph to correspond to a synchronizable global test sequence of M. This synchronizable global test sequence will ensure also that no 1-shift output fault remain undetected, and is minimal both in terms of the total number of external coordination message exchanges and the total length of the sequence.

Formally, the proposed method is a solution of the following problem:

Consider a minimal and deterministic *np*-FSM $M = (S, \Sigma, \Gamma, \delta, \lambda, s_0)$ which is represented by a strongly connected digraph $G = (V, E)$. As in the case of [LB94], [CR99], [YT98] and [HR00], let $\Phi(M) = \{M' = (S, \Sigma, \Gamma, \delta', \lambda, s_0)\}$. That is, $\Phi(M)$ is the set of all those implementations of M, each of which has no transfer fault and has the

same sets of states, inputs, outputs and the initial state as M. Suppose that the test architecture to be used for testing implementations of M in $\Phi(M)$ is the same as that of [CR99], i.e., it consists of n testers and supports direct communication between testers through the use of a multicast channel and exchange of external coordination messages. Suppose also that the cost of executing an external coordination message is higher than that of a transition of M, and thus should be avoided when possible. Then, given M and $\Phi(M)$, construct a synchronizable global test sequence such that it:

- distinguishes M from any faulty implementation of M in $\Phi(M)$,
- will not cause any controllability and observability problems when applied in a distributed test architecture, and
- is minimal, both in terms of the number of transitions of M and the number of external coordination message exchanges.

Below, we will present the proposed method as the solution of the above stated problem for an np-FSM where $n = 2$ and the ports are labelled U (Upper) and L (Lower). In Section 4.6, we will extend the solution for $n > 2$.

4.2 Description of the Proposed Method

In the first phase of the proposed method, the digraphs $G' = (V', E')$ and $G'' = (V'', E'')$ are constructed from the digraph $G = (V, E)$ of the given $2p$-FSM M for the purpose of identifying the eligible successors of each transition of M and inserting external coordination message exchanges relating to controllability to eliminate potential controllability problems.

Given $G = (V, E)$, perform the following steps:

- Step 1. For each vertex $v_i \in V$:

- create a pair of vertices i^U and i^L in V', and
- create a pair of dashed edges $(i^U, i^L; <-C_L, +C_U>)$ and $(i^L, i^U; <-C_U, +C_L>)$ in E', which indicate the external coordination message exchanges relating to controllability.

- Step 2. For each edge $e_{jk} = (v_j, v_k; x/y) \in E$, create the following edges in E':
  - $(j^U, k^U; x/y)$, if $x \in \Sigma_U$, and $\nexists$ an a_L in y,
  - $(j^U, k^U; x/y)$ and $(j^U, k^L; x/y)$, if $x \in \Sigma_U$ and $\exists$ an a_L in y,
  - $(j^L, k^L; x/y)$, if $x \in \Sigma_L$, and $\nexists$ an a_U in y,
  - $(j^L, k^L; x/y)$ and $(j^L, k^U; x/y)$, if $x \in \Sigma_L$ and $\exists$ an a_U in y.
- Step 3. For each vertex $v \in V'$ where only dashed edges are arriving and leaving, remove from E' dashed edges arriving and leaving v and then remove v from V'. After this step is complete, the resulting digraph will be known as $G' = (V', E')$.
- Step 4. Initially, let $V'' = V'$ and $E'' = E'$.

  For each vertex v_i in V':
  - for each pair of edges $(v_i, v_j^U; x/y)$ and $(v_i, v_j^L; x/y)$:
    - create a null vertex v_i^x in V'',
    - create a solid bold edge $(v_i, v_i^x, -/-)$ in E'',
    - create two edges $(v_i^x, v_j^U; x/y)$ and $(v_i^x, v_j^L; x/y)$ in E'' and eliminate $(v_i, v_j^U; x/y)$ and $(v_i, v_j^L; x/y)$ from E''.
  - any remaining solid edges leaving v_i are made bold.

  After this step is complete, the resulting digraph will be known as $G'' = (V'', E'')$.

In the first step, each vertex j^U (j^L) represents the starting state v_j of a transition with the input operation related to $U(L)$, and each dashed edge (j^U, j^L; <-C_L, +C_U>) (or (j^L, j^U; <-C_U, +C_L>)) represents an external coordination message exchange relating to controllability. In the second step, each edge represents a transition of M. Note that in any traversal of G', any two adjacent transitions of G will be covered without creating a synchronization problem. Note also that for any two states v_j and v_k, the shortest paths from j^U (or j^L) to k^U (or k^L) can be used to generate the minimum-cost synchronizable transfer sequences which transfer M from state v_j to v_k. After the fourth step, a rural Chinese postman tour of G'' over the set of bold edges can be generated as a minimum-cost synchronizable global test sequence of M. If there is a requirement to minimize the total number of external coordination message exchanges, each dashed edge is given a sufficiently high cost so that a rural Chinese postman tour avoids those edges whenever possible.

Consider the digraph $G = (V, E)$ of a $2p$-FSM $M3$ shown in Figure 11. Figures 12 and 13 show the digraphs $G' = (V', E')$ and $G'' = (V'', E'')$. A rural Chinese postman tour over the bold edges of G'' given in Figure 13 yields a minimum-cost synchronizable global test sequence that does not use any external coordination message exchanges as the label of the path:

$t1, t3, t4, t5, t6, t4, t5, t1, t2,$

with a total of nine input/output pairs.

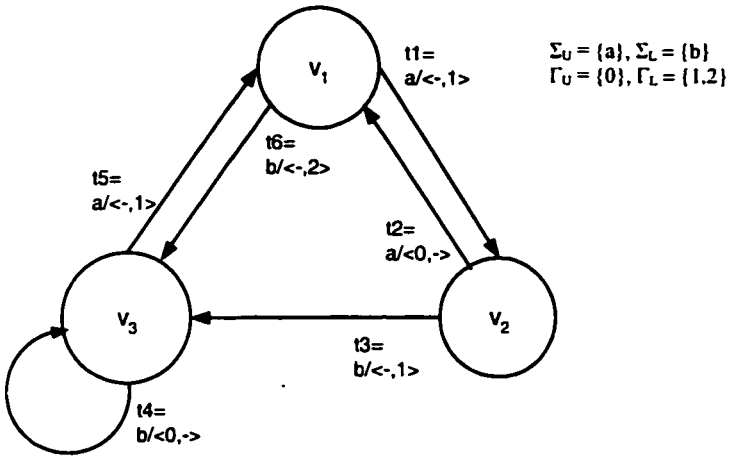

Figure 11. Digraph $G = (V, E)$ of 2p-FSM $M3$

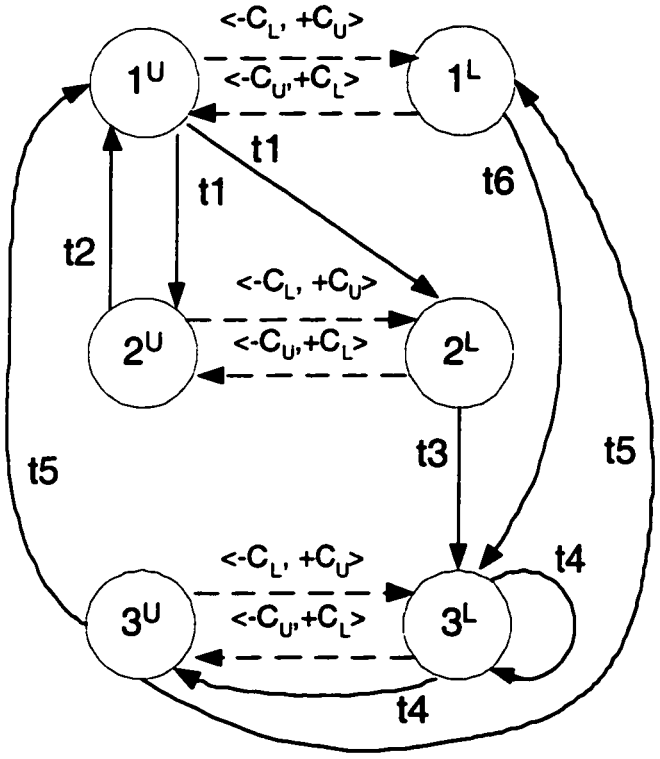

Figure 12. Digraph $G' = (V', E')$ of 2p-FSM $M3$

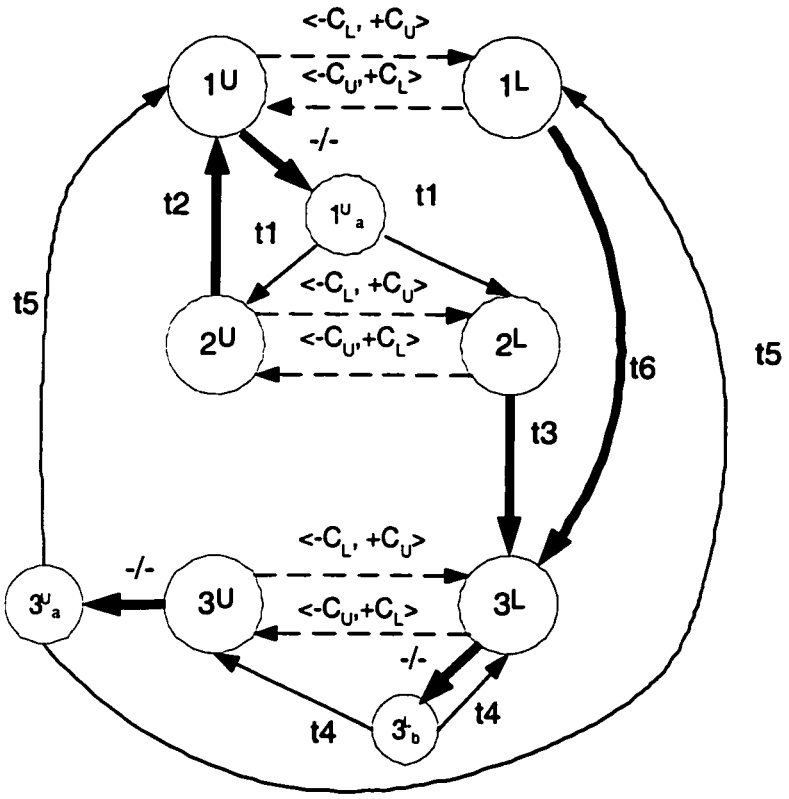

Figure 13. Digraph $G'' = (V'', E'')$ of 2*p*-FSM *M*3

In the second phase of the proposed method, the set T_0 of all transition pairs with a potential undetectable 1-shift output fault is created and the digraph $G'' = (V'', E'')$ is modified using the information in T_0 to form the digraph $G''' = (V''', E''')$. T_0 can be constructed from $G = (V, E)$ as follows:

for each vertex $v_j \in V$ *do*

for each edge e_{ij} *(say* t_{ij}*)* $= (v_i, v_j; x_j/y_j)$ *entering vertex* v_j *do*

for each edge e_{jp} *(say* t_{jp}*)*$= (v_j, v_p; x_{j+1}/y_{j+1})$ *leaving vertex* v_j *do*

if for some output $a_k \in \Gamma_b$ a_k *is in* y_j *XOR* a_k *is in* y_{j+1} *AND* $x_{j+1} \notin \Sigma_b$ *then add* $(t_{ij}, t_{jp}, <-O_b, +O_h>)$ *to* T_0*, where h is the tester sending the input* x_{j+1} *to the IUT in* t_{jp} *and k is the tester involved in the shift.*

By applying this algorithm, we discover 6 pairs of transitions in which a potential undetectable 1-shift output fault may occur, i.e.:

T_0 = {($t1$, $t2$, <-O_L, +O_U>), ($t2$, $t1$, <-O_L, +O_U>), ($t2$, $t6$, <-O_U, +O_L>), ($t3$, $t4$, <-O_U, +O_L>), ($t4$, $t5$, <-O_L, +O_U>), ($t6$, $t4$, <-O_U, +O_L>)}.

The digraph $G'' = (V'', E'')$ can now be modified to form the digraph $G''' = (V''', E''')$, using the information in T_0:

- Step 1. For each (t_{ij}, t_{jp}, <-O_k, +O_h>) triple in T_0, identify the vertex v_j in V'' (i.e., either j^U or j^L) such that v_j is $tail(t_{ij})$. Add a new vertex v_j* in V''' (if one does not exist already).
- Step 2. A dashed edge from v_j* to v_j is inserted in E''', with the label "t_{ij}, t_{jp}, <-O_k, +O_h>" for each (t_{ij}, t_{jp}, <-O_k, +O_h>) triple in T_0. This dashed edge indicates an external coordination message exchange relating to observability. If there exists a triple (t_{lm}, t_{mq}, <-O_t, +O_u>) in T_0 where, in order to traverse the transition pair (t_{lm}, t_{mq}), the same dashed edge from v_j* to v_j must be traversed as is traversed for the transition pair (t_{ij}, t_{jp}), then an alternate label "t_{lm}, t_{mq}, <-O_t, +O_u>" is added to this dashed edge.
- Step 3. For each solid edge t_{ij} in E'' whose label is not -/-, if the edge t_{ij} leaves vertex v in V'' then it will leave the same vertex in V'''. If the edge t_{ij} is also contained in some transition pair (t_{ij}, t_{jk}) in a triple in T_0 and arrives at a vertex v_j in V'', then the edge t_{ij} will go to v_j* in V''', otherwise it will go to v_j in V'''. The resulting digraph is $G''' = (V''', E''')$
- Step 4. After constructing an RCPT of G''' over the bold edges, post processing of the graph may be required in the case where alternate labels exist for dashed edges indicating coordination message exchanges relating to observability. The correct

alternative must be chosen in order to coincide with the transition pair selected in a particular subsequence of the tour. As well, we adopt the convention that any two consecutive transitions on a path in G that would be covered in the same order in a synchronizable traversal of G''', but with the inclusion of external coordination message exchanges relating to both observability and controllability, will have the controllability message exchange processed first. For example, in an RCPT over the bold edges of $G''' = (V''', E''')$ of the $2p$-FSM $M3$ shown in Figure 14, the test subsequence $t2$, $t1$ would require only the inclusion of the $<-O_L, +O_U>$ coordination message exchange relating to observability, while the test subsequence $t2$, $t6$ would require only the inclusion of $<-C_L, +C_U>$, $<-O_U, +O_L>$.

Note that any two consecutive transitions on a path in G that would be covered in the same order in a synchronizable traversal of G''' do so without any possibility of a potential undetectable 1-shift output fault. Indeed, an RCPT of G''' over the set of bold edges will yield a minimum-cost synchronizable global test sequence with no possibility of potential undetectable 1-shift output faults. A proof for this statement is given in the next section. If there is a requirement to minimize the total number of external coordination message exchanges, each dashed edge is given a sufficiently high cost so that an RCPT avoids those edges whenever possible. Note, however, that the existence of duplicate edges (of the form $(v_i^x, v_j^U; x/y)$ and $(v_i^x, v_j^L; x/y)$ as described in Step 4 of the first phase of the proposed method) may result in a case where two consecutive transitions t_{ij} and t_{jk} on a path in G''' may be covered with either a coordination message exchange relating to observability or a coordination message exchange relating to controllability inserted between them. For example, in Figure 14, the transitions $t4$ and $t5$

may be traversed in G''' of 2p-FSM $M3$ as $t4$, <-C_U, +C_L>, $t5$ or $t4$, <-O_L, +O_U>, $t5$. In these cases we ensure that the path containing the observability message exchange is selected by assigning the cost of observability message exchange edges lower than the cost of controllability message exchange edges.

Figure 14 shows the digraph G''', where an RCPT over the bold edges yields a minimum-cost synchronizable global test sequence that has no potential undetectable 1-shift output faults as the label of:

$t1, t3, <-O_U, +O_L>, t4, <-O_L, +O_U>, t5, t6, <-O_U, +O_L>, t4, <-O_L, +O_U>, t5, t1,$

$<-O_L, +O_U>, t2,$

for an addition of five external coordination message exchanges relating to observability.

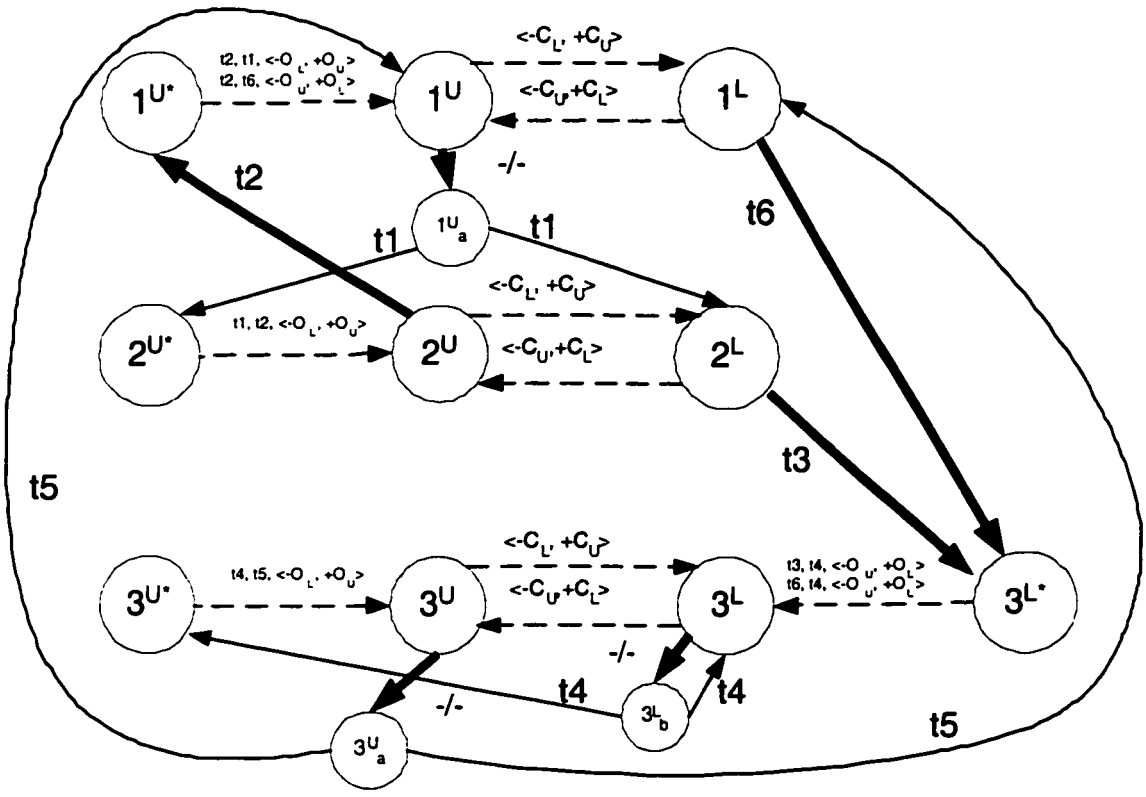

Figure 14. Digraph $G'''=(V''', E''')$ for 2p-FSM $M3$

4.3 Showing Absence of Undetectable 1-shift Output Faults

We make the claim in the previous section that an RCPT of $G''' = (V''', E''')$ over the set of bold edges will be a minimum-cost synchronizable global test sequence with no possibility of potential undetectable 1-shift output faults. Here we show the validity of this claim through the examination of the graph construction method.

Theorem 1: Given a $2p$-FSM that is minimal, deterministic and represented by a strongly connected digraph $G = (V, E)$, the proposed method constructs a global test sequence that is synchronizable and has no potential undetectable 1-shift output faults.

Proof: After the first phase of the proposed method, any two consecutive transitions on a path in $G = (V, E)$ that do not represent a pair of synchronizable transitions become separated in $G'' = (V'', E'')$ by a dashed edge (less null edges) representing an external coordination message exchange relating to controllability. Any two adjacent edges of G that remain adjacent in G'' (less null edges) form a synchronizable pair of transitions. Thus, any two solid edges (less null edges) in the digraph G'' that represent two consecutive transitions on a path in G form a synchronizable sequence of transitions, either through their own properties (i.e. they are adjacent in G'') or through the insertion of an external coordination message exchange relating to controllability. A global test sequence generated from G'' would therefore be synchronizable.

By a similar argument, after the second phase of the proposed method any two solid edges (less null edges) in $G''' = (V''', E''')$ that represent two consecutive transitions on a path in G with a potential undetectable 1-shift output fault become separated by a dashed edge representing an external coordination message exchange relating to observability.

Any two adjacent solid edges (less null edges) in G''' form a synchronizable pair of transitions without potential undetectable 1-shift output faults. Thus, any two solid edges (less null edges) in the digraph G''' that represent two consecutive transitions on a path in G form a synchronizable sequence of transitions without any potential undetectable 1-shift output faults, either through their own properties (i.e. they are adjacent edges in G''') or through the insertion of external coordination message exchanges relating to observability and/or controllability.

A global test sequence generated from G''' would therefore be synchronizable, and would have no potential undetectable 1-shift output faults.

Q.E.D.

As we have shown, the proposed method can eliminate the existence of any potential undetectable 1-shift output fault of an implementation of an FSM during testing. The proposed method can also be used to show the absence of any potential undetectable 1-shift output fault in any synchronizable global test sequence that can be derived from a given FSM. Consider the digraph $G = (V, E)$ of the $2p$-FSM $M4$ shown in Figure 15. The associated digraph $G'' = (V'', E'')$ is given in Figure 16. When we attempt to form the set T_0 of all transition pairs with a potential undetectable 1-shift output fault, T_0 is shown to be empty. Therefore, there is no possibility of an undetectable 1-shift output fault in any synchronizable global test sequence generated from G''. In this case, G'' may be used to generate a synchronizable global test sequence with no possibility of potential undetectable 1-shift output faults, and using no external coordination message exchanges relating to observability.

Although computing an RCPT of a given graph is known to be NP-complete [AA88], [AA88] and [CU95] present a class of graphs in which an RCPT can be found in polynomial time. By applying their findings to our method, we can say that if an edge induced spanning subgraph $G[E_C]$ of $G''' = (V''', E''')$ is weakly connected, E_C being the set of bold edges in G''', then finding an RCPT of G''' over E_C can be done in polynomial time. We improve our likelihood of finding a weakly connected subgraph by assuming that since nodes connected by edges representing coordination message exchanges relating to controllability are always strongly connected, they may be considered as single nodes.

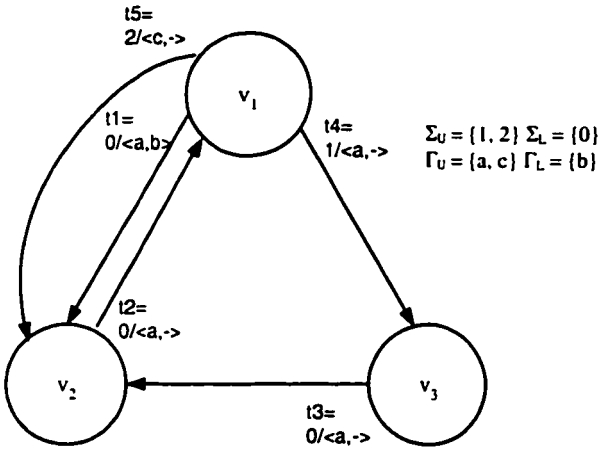

Figure 15. Digraph $G = (V, E)$ of 2p-FSM $M4$

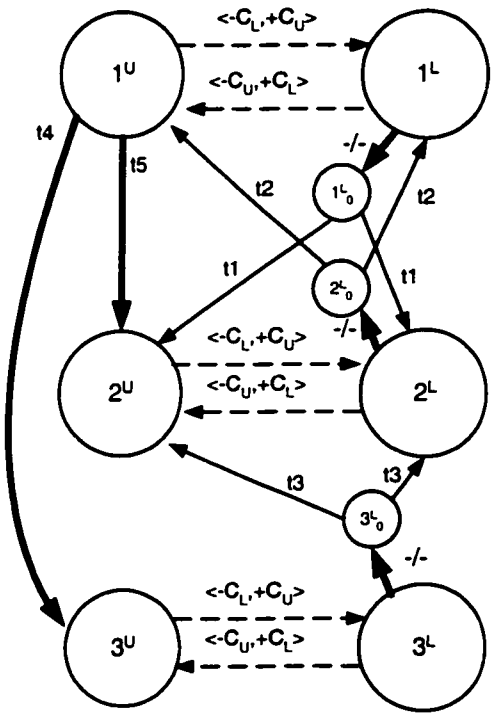

Figure 16. Digraph $G'' = (V'', E'')$ of $2p$-FSM $M4$

4.4 Comparison with Other Methods

Using the FSM shown in Figure 9 (from [CR99]) as an example reveals no difference in the resulting synchronizable global test sequences generated by the proposed method and by the method of [CR99], as there is only one way to obtain a Chinese postman tour for that graph. However, there are cases where [CR99] may or may not obtain a synchronizable global test sequence free from external coordination message exchanges relating to controllability and observability, while our method guarantees that, if such a synchronizable global test sequence exists, then it will be found.

For example, consider the digraph $G = (V, E)$ of the $2p$-FSM $M5$ shown in Figure 17. There are 24 possible Chinese postman tours that could be used to generate a synchronizable global test sequence for this FSM, shown in Table 1, one of which would be selected by [CR99] to first generate a transition tour and then to obtain a

synchronizable global test sequence. Note that although a global test sequence based on any of the 24 Chinese postman tours contains no controllability problems, it will contain observability problems. This is because any instance of $t1$ or $t2$ followed directly by $t4$ or $t5$ (or vice versa) will result in a potential undetectable 1-shift output fault. A method like [CR99] that obtains a global test sequence before checking for controllability or observability problems would therefore require the insertion of an external coordination message exchange relating to observability into any of the synchronizable global test sequences shown in Table 1. With our method, an RCPT over the bold edges of the digraph $G''' = (V''', E''')$ obtained from G is guaranteed to result in a synchronizable global test sequence free from external coordination message exchanges relating to observability, at the cost of one extra transition. The digraphs $G' = (V', E')$, $G'' = (V'', E'')$ and $G''' = (V''', E''')$ that were constructed from $G = (V, E)$ of the $2p$-FSM $M5$ using our method are shown in Figures 18, 19 and 20. T_0 for this example is: $\{(t1, t4. <\text{-}O_U, +O_L>), (t1, t5, <\text{-}O_U, +O_L>), (t2, t4, <\text{-}O_U, +O_L>), (t2, t5, <\text{-}O_U, +O_L>), (t4, t1, <\text{-}O_L, +O_U>), (t4, t2, <\text{-}O_L, +O_U>), (t5, t1, <\text{-}O_L, +O_U>), (t5, t2, <\text{-}O_L, +O_U>)\}$

It can be seen, for example, that a synchronizable global test sequence which is the label of $<t1, t3, t4, t6, t2, t3, t5>$, obtained by following an RCPT of G''' over the bold edges, has no controllability or observability problems, nor does it require the insertion of any external coordination message exchange to ensure the absence of controllability or observability problems.

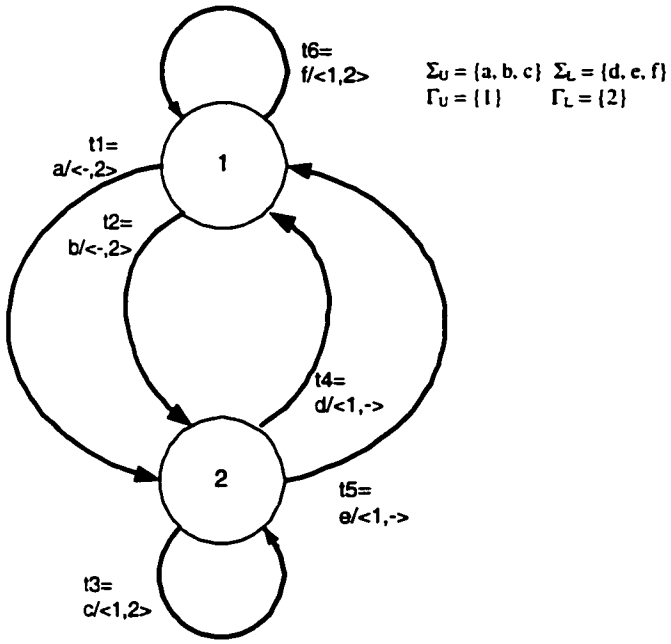

Figure 17. Digraph $G = (V, E)$ of $2p$-FSM $M5$

Table 1 – List of possible CPTs of $2p$-FSM $M5$

$t1, t3, t4, t2, t5, t6$	$t2, t3, t4, t1, t5, t6$	$t6, t1, t3, t4, t2, t5$
$t1, t3, t4, t6, t2, t5$	$t2, t3, t4, t6, t1, t5$	$t6, t1, t3, t5, t2, t4$
$t1, t3, t5, t2, t4, t6$	$t2, t3, t5, t1, t4, t6$	$t6, t1, t4, t2, t3, t5$
$t1, t3, t5, t6, t2, t4$	$t2, t3, t5, t6, t1, t4$	$t6, t1, t5, t2, t3, t4$
$t1, t4, t2, t3, t5, t6$	$t2, t4, t1, t3, t5, t6$	$t6, t2, t3, t4, t1, t5$
$t1, t4, t6, t2, t3, t5$	$t2, t4, t6, t1, t3, t5$	$t6, t2, t3, t5, t1, t4$
$t1, t5, t2, t3, t4, t6$	$t2, t5, t1, t3, t4, t6$	$t6, t2, t4, t1, t3, t5$
$t1, t5, t6, t2, t3, t4$	$t2, t5, t6, t1, t3, t4$	$t6, t2, t5, t1, t3, t4$

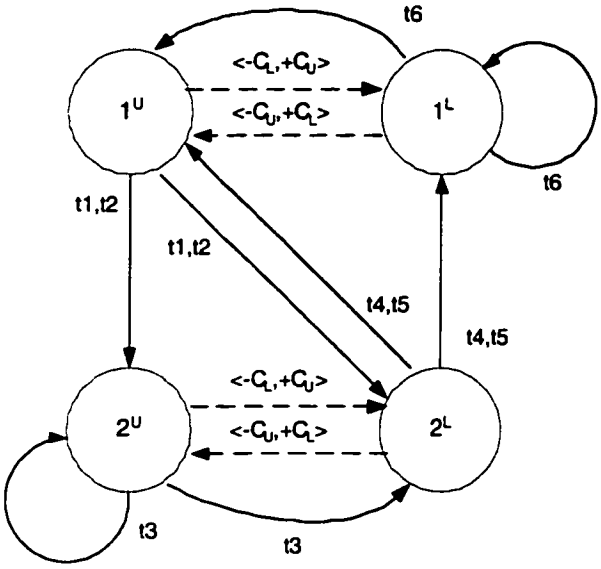

Figure 18. Digraph $G' = (V', E')$ of $2p$-FSM *M5*.

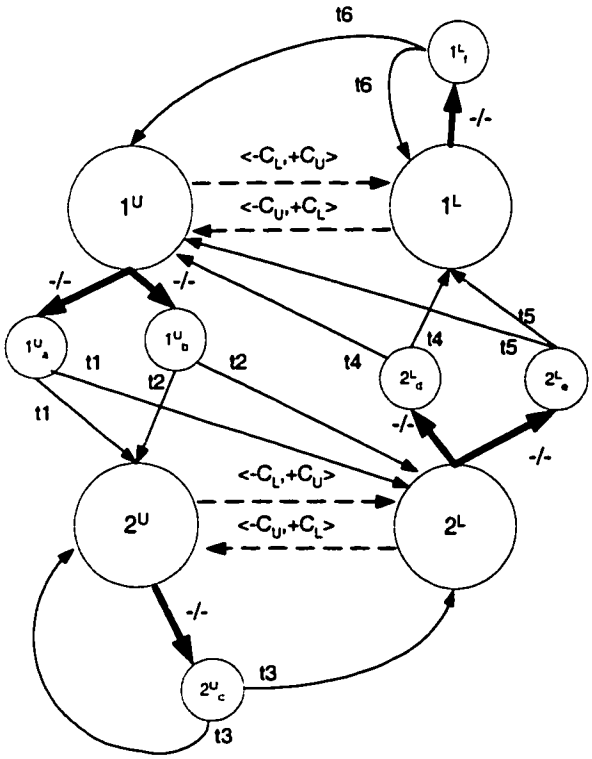

Figure 19. Digraph $G'' = (V'', E'')$ of $2p$-FSM *M5*

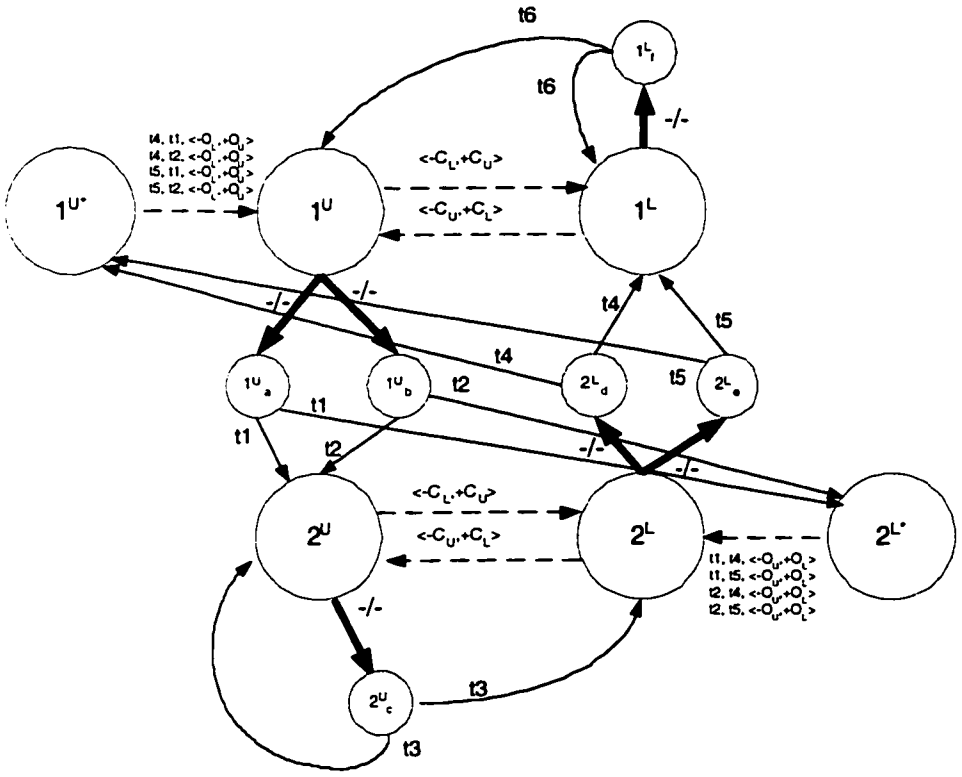

Figure 20. Digraph $G''' = (V''', E''')$ of 2*p*-FSM *M5*

[HR00] identifies two cases of potential undetectable 1-shift output faults that are used to determine where in the local test sequences to insert the external coordination message exchanges relating to observability. Case 1, which is referred to as a backward output-shifting fault in [HR00] and [YT98], requires an external coordination message to inform tester k that it should expect to receive output a_k from the IUT. Case 2, which is referred to as a forward output-shifting fault in [HR00] and [YT98], requires an external coordination message to inform tester k that it should have received output a_k already. The method proposed in [HR00] uses a different message for detecting forward output-shifting faults (the post transition framing message) than for backward output-shifting faults (the pre transition framing message). We will see that in our proposed method, it is not necessary to formally distinguish between the two cases in order to properly place the

external coordination message exchanges relating to observability, nor are different types of messages required for different types of output shifts.

Consider a synchronizable global test sequence which is the label of the path

$t1$, $t3$, $<-O_U, +O_L>$, $t4$, $<-O_L, +O_U>$, $t5$, $t6$, $<-O_U, +O_L>$, $t4$, $<-O_L, +O_U>$, $t5$, $t1$, $<-O_L, +O_U>$, $t2$, derived from an RCPT over the bold edges of the digraph $G''' = (V''', E''')$ of $2p$-FSM $M3$, shown in Figure 14. For demonstration purposes, in each case of an external coordination message exchange relating to observability we annotate the message "bak" if it is of Case 1, and "fwd" if it is of Case 2. The resulting synchronizable global test sequence ω now reads:

$a/<-, 1>$, $b/<-, 1>$, $<-O_U, +O_L>^{bak}$, $b/<0, ->$, $<-O_L, +O_U>^{bak}$, $a/<-, 1>$, $b/<-, 2>$, $<-O_U, +O_L>^{bak}$, $b/<0, ->$, $<-O_L, +O_U>^{bak}$, $a/<-, 1>$, $a/<-, 1>$, $<-O_L, +O_U>^{fwd}$, $a/<0, ->$.

The following two local test sequences can then be generated from ω:

- $\omega_U = a/<-, ->, -/<-, ->, +O_L, -/<0, ->, -O_L, a/<-, ->, -/<-, ->, +O_L, -/<0, ->, -O_L, a/<-, ->, a/<-, ->, -O_L, a/<0, ->$
- $\omega_L = -/<-, 1>, b/<-, 1>, -O_U, b/<-, ->, +O_U, -/<-, 1>, b/<-, 2>, -O_U, b/<-, ->, +O_U, -/<-, 1>, -/<-, 1>, +O_U, -/<-, ->$

Consider the local test sequence ω_U and the first "$+O_L$" in that test sequence, indicating the expected receipt of an external coordination message relating to observability from the Lower Tester. Note that this "$+O_L$" message is preceded by the transition "-/<-, ->" and succeeded by the transition "-/<0, ->." Since only the succeeding transition contains an output to the Upper Tester from the IUT, it is clear that this output "0" is the only one that could result in a potential undetectable 1-shift output fault. It is also clear that the output could only shift from the transition succeeding the "$+O_L$"

message to the transition preceding it, or a backward output-shifting fault in the terminology of [HR00]. We know then that the "$+O_L$" message in this case is intended to tell the Upper Tester to "prepare to receive an output from the IUT."

Similarly, the last "$+O_U$" message in the local test sequence ω_L is preceded by the transition "-/<-, 1>" and succeeded by the transition "-/<-, ->." Since only the preceding transition contains an output to the Lower Tester from the IUT, that output could only result in a potential forward output-shifting fault in the terminology of [HR00], and the "$+O_U$" message could only mean "you should have received an output from the IUT by now."

In this way, 1-shift output faults in our method do not need to be formally designated as forward output-shifting faults or backwards output-shifting faults, nor do we require two types of coordination message exchanges, as is the case with [HR00]'s post-transition framing message and pre-transition framing message. A single message type, the external coordination message exchange relating to observability, is sufficient to detect all potential undetectable 1-shift output faults.

It is somewhat more challenging to compare the method described in [LB94] with our proposed method, as the two methods make use of different test architectures. [LB94] assumes the use of the general distributed test architecture, where testers communicate amongst themselves indirectly via their interactions with the IUT, while the proposed method assumes a multi-cast channel, as in [CR99], whereby testers may communicate directly through the exchange of external coordination messages. However, with a minor augmentation, the proposed method can be shown to be at least as effective as the approach in [LB94], even using the general distributed test architecture.

In the case of an FSM where one or more synchronizable global test sequences exists, both the method in [LB94] and our proposed method will be successful in finding one. This is because in [LB94], a transition may not be added to the existing global test subsequence unless it is an eligible successor of the final transition in the sequence constructed so far. The proposed method in this thesis entails a tour of a digraph $G''' = (V''', E''')$ whereby each transition leaving a particular state is the eligible successor of all transitions (or edges representing coordination message exchanges) entering it. Every pair of synchronizable transitions that are traversable in the digraph $G = (V, E)$ (assuming a synchronizable test sequence exists) will therefore also be traversable in G''' (following the proposed method) without the necessity of resorting to external coordination message exchanges. Because of the heuristic nature of [LB94]'s method, it is difficult to assess which approach is more effective in constructing a minimum length synchronizable global test sequence. However, given an FSM where one or more synchronizable global test sequences do exist, while [LB94]'s method may or may not generate a minimum length synchronizable global test sequence, our method guarantees finding one with minimal length, without inserting any external coordination message exchanges relating to controllability. Consider the digraph $G = (V, E)$ of the $2p$-FSM *M6* shown in Figure 21 and disregard observability problems for the moment. Our method will construct a minimum length synchronizable global test sequence that is the label of:

$t3, t5, t6, t11, t10, t9, t7, t1, t2, t6, t9, t8, t5, t6, t11, t12, t4,$

for a total of 17 input/output pairs. While the method of [LB94] may construct the same synchronizable global test sequence, there are many others that it may consider from this

digraph that have a larger cost in terms of input/output pairs. For example, another synchronizable global test sequence for this digraph is one that is the label of:

*t*3, *t*5, *t*6, *t*11, *t*10, *t*9, *t*7, *t*1, *t*2, *t*6, *t*11, *t*12, *t*5, *t*6, *t*9, *t*8, *t*5, *t*6, *t*11, *t*12, *t*4,

for a total of 21 input/output pairs.

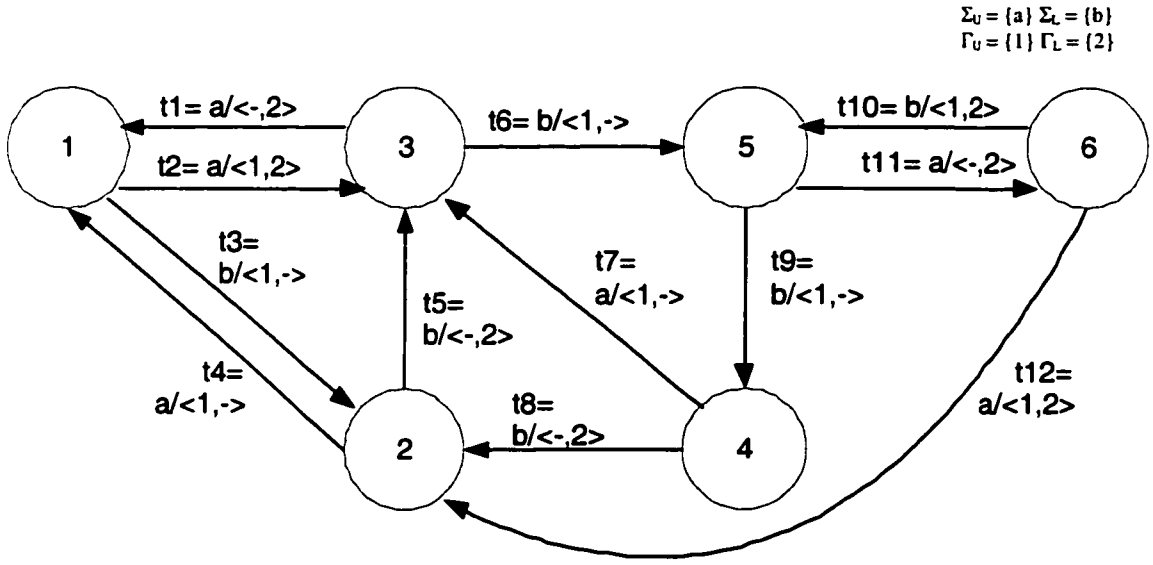

Figure 21. Digraph $G = (V, E)$ of 2*p*-FSM *M*6

In terms of observability problems, the proposed method can be augmented slightly to produce a synchronizable global test sequence with no potential undetectable 1-shift output faults and without using any external coordination message exchanges relating to observability, as the same test architecture used in [LB94] will be considered. The augmented method is as follows:

- Use the method proposed in this thesis to construct a digraph $G''' = (V''', E''')$ from the digraph $G = (V, E)$ of a given *np*-FSM.
- Find an RCPT of G''' over the set of bold edges that minimizes the number of external coordination message exchanges relating to observability and controllability.

- If an external coordination message exchange relating to controllability exists in the resulting global test sequence then stop, since the digraph G has no synchronizable global test sequence that does not contain an external coordination message exchange relating to controllability. In this case, the method of [LB94] would also fail to find a synchronizable global test sequence.
- Otherwise, for each instance of an external coordination message exchange relating to observability in the generated synchronizable global test sequence, remove the external coordination message exchange and append the synchronizable global test sequence with a synchronizable test subsequence that will ensure the absence of potential undetectable 1-shift output faults between the pairs of transitions identified by the external coordination message exchanges relating to observability.

As an example, consider the digraph of the $2p$-FSM $M1$ shown in Figure 6. Using the proposed method, the digraphs $G' = (V', E')$, $G'' = (V'', E'')$ and $G''' = (V''', E''')$ are constructed as shown in Figures 22, 23, and 24. T_0 as derived from G is:

$\{(t1, t3, <\text{-}O_U, +O_L>), (t1, t4, <\text{-}O_L, +O_U>), (t2, t3, <\text{-}O_U, +O_L>), (t3, t2, <\text{-}O_U, +O_L>),$

$(t4, t1, <\text{-}O_L, +O_U>)\}$.

An RCPT over the bold edges of G''' yields a minimum-length synchronizable global test sequence that is the label of:

$t1, <\text{-}O_U, +O_L>, t3, <\text{-}O_U, +O_L>, t2, t4,$

which requires no external coordination message exchanges relating to controllability. Removing the external coordination message exchanges relating to observability and appending the synchronizable global test sequence with the appropriate synchronizable test subsequences will result in a synchronizable global test sequence with the label of:

$t1, t3, t2, t4, t1, t4, t1, t3$.

This is the same synchronizable global test sequence with no potential undetectable 1-shift output faults that is found using the method of [LB94].

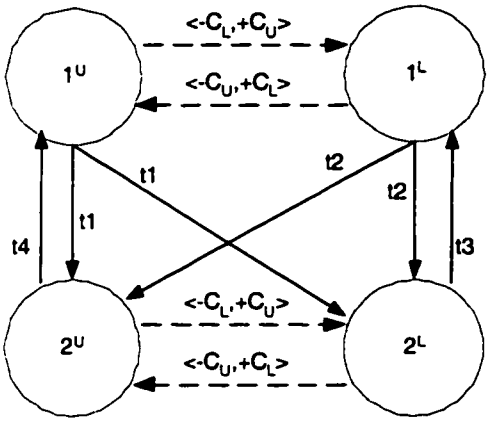

Figure 22. Digraph $G' = (V', E')$ of $2p$-FSM $M1$

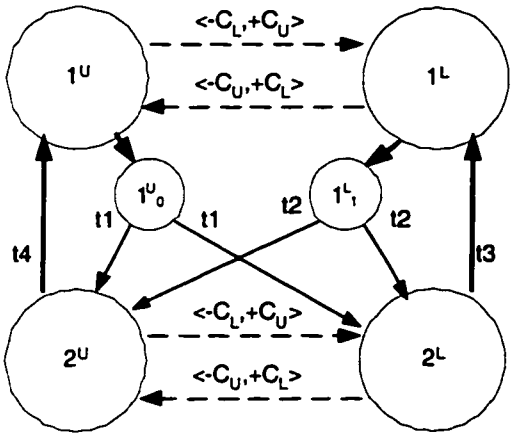

Figure 23. Digraph $G'' = (V'', E'')$ of $2p$-FSM $M1$

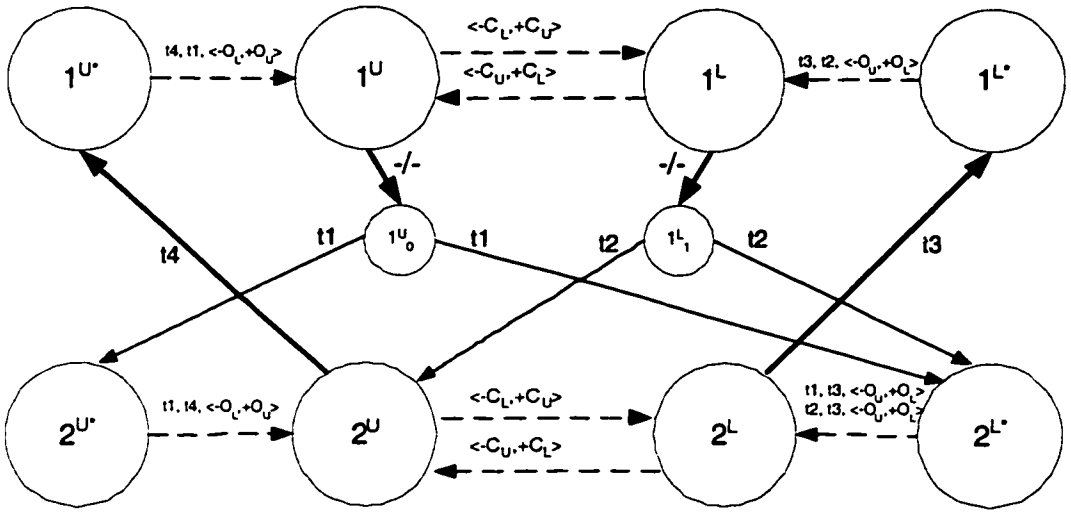

Figure 24. Digraph $G''' = (V''', E''')$ of $2p$-FSM $M1$

This augmented method as well as the method of [LB94] selects in an ad hoc manner the synchronizable test subsequences that will be appended to the synchronizable global test sequence in order to ensure the absence of potential undetectable 1-shift output faults. However, a set of rules may be applied to assist in the selection process. Synchronizable test subsequences may be selected according to the following steps:

- Step 1. **Identify $<-O_v, +O_w>$ to be replaced.** For each transition pair (t_{ij}, t_{jk}) from a triple $(t_{ij}, t_{jk}, <-O_v, +O_w>)$ in T_0 such that $label(t_{ij})$, $<-O_v, +O_w>$, $label(t_{jk})$ is a subsequence of a given synchronizable global test sequence for a given $2p$-FSM M, form a new column in a table to be constructed.
- Step 2. **Identify candidates to replace $<-O_v, +O_w>$.** Under the column for a transition pair (t_{ij}, t_{jk}) in the table, form a new row for each possible pair of transitions (t_{pi}, t_{ij}), (t_{ij}, t_{jq}), (t_{rj}, t_{jk}) and (t_{jk}, t_{ks}) that represent pairs of adjacent transitions including t_{ij} and t_{jk} in $G = (V, E)$ of M. Remove all non-synchronizable pairs of transitions in that column. One of the remaining transition pairs will be selected according to Step 3 and the selected transition pair will be appended on to the synchronizable global test

sequence in order to detect the potential undetectable 1-shift output fault for that column.

- Step 3. **Select one candidate to replace <-O_x, +O_x>.**

- a) For each column and for each row in that column, consider the transition pair (t_{lm}, t_{mn}) added in Step 2, and record *head*(t_{lm}) and *tail*(t_{mn}) from G' = (V', E') . In the case where t_{mn} is a duplicate edge (i.e., of the form (v_m^x, v_n^U; *x/y*) and (v_m^x, v_n^L; *x/y*) within G', $v^{U,L}$ may be written for *tail*(t_{mn}).

- b) For each column and for each row in that column, if the transition pair (t_{lm}, t_{mn}) in that row is such that:

  i) *tail*(t_{mn}) is the same as *head*(t_{xy}) of a transition pair (t_{xy}, t_{yz}) in another column, then that transition pair (t_{lm}, t_{mn}) shall be referred to as **right-attachable**. If *tail*(t_{mn}) is of the form $v^{U,L}$, then *head*(t_{xy}) from another column that is either v^U or v^L may be considered,

  ii) *head*(t_{lm}) is the same as *tail*(t_{yz}) of a transition pair (t_{xy}, t_{yz}) in another column, then that transition pair (t_{lm}, t_{mn}) shall be referred to as **left-attachable**. If *tail*(t_{yz}) from another column is of the form $v^{U,L}$, then *head*(t_{lm}) that is either v^U or v^L may be considered,

Transition pairs that are neither right-attachable or left-attachable are removed.

- c) For each column and for each row in that column, if t_{mn} from any (t_{lm}, t_{mn}) pair left after b) does not involve an input to the IUT from the tester involved in the shift for that column that occurs after the occurrence of the output being shifted, then find an eligible successor t_{no} of t_{mn} that does involve an input to the IUT from the tester involved in the shift and place it in brackets after t_{mn}. This is to indicate that in order

to detect a potential undetectable 1-shift output fault, (t_{lm}, t_{mn}) must be followed by a transition pair (t_{no}, t_{op}) from another column. If such a pair does not exist, (t_{lm}, t_{mn}) may only be used if it is included as the final transition pair in the synchronizable global test sequence and if its inclusion would detect the potential 1-shift output fault for that column. Add empty brackets after t_{mn} in this case, indicating (t_{lm}, t_{mn}) may only be used as the final pair in the synchronizable global test sequence. Otherwise, it is removed.

- d) Select one transition pair (t_{lm}, t_{mn}) from each column to append onto the synchronizable global test sequence such that t_{lm} from the first pair chosen is an eligible successor to the last transition in the synchronizable global test sequence, and each subsequent transition pair is chosen so that its head is the same as the tail of its immediate predecessor.

For example, consider the potential undetectable 1-shift output fault between $t1$ and $t3$ in the minimum length synchronizable global test sequence for the $2p$-FSM $M1$, which is the label of $t1$, $t3$, $t2$, $t4$. According to Step 1 described above, this transition pair is first inserted as the first column in Table 2. In Step 2, all possible pairs that represent adjacent transitions in $G = (V, E)$ of $M1$ that include either $t1$ or $t3$ (i.e., ($t3$, $t1$), ($t1$, $t4$), ($t4$, $t1$), ($t2$, $t3$), ($t3$, $t2$)) are added as rows to the column. All transition pairs in this case are synchronizable except ($t3$, $t1$), which is discarded. In Step 3a), the head and tail of each transition pair from $G' = (V', E')$ of $M1$ is recorded. For example, in the first transition pair ($t1$, $t4$), $t1$ leaves and $t4$ enters node 1^U. In Step 3b), transition pairs ($t4$, $t1$) and ($t3$, $t2$) are discarded, as they cannot be preceded or followed by any transition pair in the other column without the insertion of an additional transition. In Step 3c), transition

pair ($t1$, $t4$) is examined to determine the result of the removal of the "b" in $t1$ as the result of a 1-shift output fault from $t1$ to $t3$. It would be detected by the Upper Tester, since the Upper Tester would not be able to send the input "0" in $t4$ without first receiving the "b" in $t1$. An insertion of a "b" into $t3$ in a ($t2$, $t3$) transition pair would not be detectable unless $t3$ was followed by a transition where the Upper Tester sends an input to the IUT. There is not enough information to determine whether it would be detectable if $t3$ was the last transition in the appended synchronizable global test sequence. Since there exists no eligible successor for $t3$ that involves an input to the IUT from the Upper Tester, the transition pair ($t2$, $t3$) is discarded. In Step 3d), the test subsequence $t1$, $t4$, $t1$, $t3$ can then be chosen to be appended to the synchronizable global test sequence for $M1$.

As noted in Section 3.3, the transition pairs appended to the synchronizable global test sequence for $M1$ result in potential undetectable 1-shift output faults that did not exist before they were appended. Specifically, there is now a potential undetectable 1-shift output fault in each of transition pairs ($t4$, $t1$) and ($t1$, $t4$). However, Table 3 shows that for each new potential undetectable 1-shift output fault that is caused by the appended part, there exists a transition pair in the original synchronizable global test sequence that detects that fault.

Table 2. Selecting transition pairs for potential undetectable 1-shift output faults – FSM *M*1

Note: $\Sigma_U = 0$, $\Sigma_L = 1$, $\Gamma_U = b$, $\Gamma_L = a, c$

	$t1$ -> $t3$ shift 0/<*b*, *a*>, 1/<-, *c*>			$t3$ <- $t2$ shift 1/<-, *c*>, 1/<*b*, ->	
	~~1/<-, *c*>, 0/<*b*, *a*>~~ $t3, t1$			~~1/<-, *c*>, 0/<*b*, *a*>~~ $t3, t1$	
1^U	0/<*b*, *a*>, 0/<*b*, ->^{U} $t1, t4$	1^U	1^U	0/<*b*, *a*>, 1/<-, *c*> ()^L $t1, t3$ ()	1^L
2^U	~~0/<*b*, ->, 0/<*b*, *a*>~~ $t4, t1$	$2^{U,L}$		~~0/<*b*, ->, 1/<*b*, ->~~ $t4, t2$	
1^L	~~1/<*b*, ->, 1/<-, *c*> ()~~ $t2, t3$ ()	1^L	1^L	1/<*b*, ->, 0/<*b*, ->^{U} $t2, t4$	1^U
2^L	~~1/<-, *c*>, 1/<*b*, ->~~ $t3, t2$	$2^{U,L}$	1^L	~~1/<*b*, ->, 1/<-, *c*> ()~~^{L} $t2, t3$ ()	1^L

Table 3. Transition pairs in synchronizable global test sequence for FSM *M*1 that detect potential undetectable 1-shift output faults in appended part

Pairs in appended part causing potential fault	Pairs in original test sequence to detect that fault
$t4, t1$ (0/<*b*, ->, 0/<*b*, *a*>)	$t1, t3$ (0/<*b*, *a*>, 1/<-, *c*>)
$t1, t4$ (0/<*b*, *a*>, 0/<*b*, ->)	$t1, t3$ (0/<*b*, *a*>, 1/<-, *c*>)

4.5 *k*-shift Output Faults

Since our definition of global test sequence requires an input $x_i \in I$ for each transition, there can be only one instance of a potential undetectable 1-shift output fault in any two consecutive transitions of a synchronizable global test sequence for a $2p$-FSM. This is because, given two transition t_j and t_{j+1} of a synchronizable global test sequence, the definition of potential undetectable 1-shift output fault requires that the output being shifted be contained in either t_j or t_{j+1}, but not both. Furthermore, any potential 1-shift output fault between t_j and t_{j+1} (or vice versa) will be detectable if the tester involved in the shift (i.e., T_{shift}) is the same tester that is sending the input to the IUT in t_{j+1}. That means a potential undetectable 1-shift output fault will only occur when the tester

sending the input to the IUT is not T_{shift}. The output being shifted may be contained in either t_j or t_{j+1}, but not both. Hence, there can only be one instance of a potential undetectable 1-shift output fault in any two consecutive transitions of a synchronizable global test sequence for a $2p$-FSM. Similarly for an np-FSM, the maximum number of potential undetectable 1-shift output faults in two consecutive transitions is n-1.

The potential undetectable k-shift output faults would be determined in much the same way as potential undetectable 1-shift output faults.

Definition: Given a $2p$-FSM M and a synchronizable global test sequence $\omega = x_1/y_1, x_2/y_2, \ldots, x_m/y_m$ of M, where $x_i \in I$ and $y_i \in O$, $1 \leq i \leq m$, the observability problem will manifest itself as an undetectable ***k*-shift output fault** in an implementation N of M when, in any two transitions (henceforth called **terminal transitions**) with labels x_j/y_j and x_{j+k}/y_{j+k}, $k \geq 2$, the following conditions exist:

- The terminal transitions must satisfy the conditions for an undetectable 1-shift output fault,
- All intermediate transitions between the terminal transitions must be such that the k-shift output fault is not detected by the testers involved in the intermediate transitions.

For example, the synchronizable global test sequence of the $2p$-FSM $M3$ described in Section 4.2, which is the label of the path:

a/<-, 1>, *b*/<-, 1>, *b*/<0, ->, *a*/<-, 1>, *b*/<-, 2>, *b*/<0, ->, *a*/<-, 1>, *a*/<-, 1>, *a*/<0, ->,

contains a potential undetectable 2-shift output fault between the third transition in the sequence (*b*/<0, ->) and the first (*a*/<-, 1>).

Theorem 2: Given a synchronizable global test sequence for a $2p$-FSM where all potential 1-shift output faults are detectable when applied in the distributed test

architecture (with or without the insertion of external coordination message exchanges relating to observability), and where each output associated with a particular transition within the synchronizable global test sequence is unique, there will be no undetectable k-shift output faults, $k \geq 2$, in the synchronizable global test sequence.

Proof: We assume the existence of a synchronizable global test sequence for a $2p$-FSM in which all 1-shift output faults are detectable, or each potential undetectable 1-shift output fault has resulted in the insertion of an external coordination message exchange relating to observability. To prove the absence of undetectable k-shift output faults, we examine the direction of the shift and all combinations of the intermediate transitions given two terminal transitions that satisfy the conditions in the definition of an undetectable k-shift output fault. For the purposes of demonstration, a number of examples are given in Table 4. Note that in all cases the shift involves the Upper Tester; however, the same arguments in the proof can be applied if the testers were reversed.

1. Let t_j and t_{j+k} be two transitions that exhibit a potential undetectable k-shift output fault in accordance with the definition, and let T_{shift} represent the tester involved in the shift, while T_{noshift} represents the other tester.
2. The shift must be of the form "j to $j+k$" or "$j+k$ to j." That is, the shift must be a forward shifting fault where an output a_{shift} is removed from $y_j = \langle a_1, a_2, \ldots, a_n \rangle$ and inserted into y_{j+k}, or it must be a backward shifting fault where an output a_{shift} is removed from $y_{j+k} = \langle a_1, a_2, \ldots, a_n \rangle$ and inserted into y_j.
3. If the shift is of the form "j to $j+k$", then one of two cases can occur. Either there is at least one transition t_l between t_j and t_{j+k} that contains an output to T_{shift} from the IUT, or no transition between t_j and t_{j+k} contains an output to T_{shift} from the IUT.

3.1 If there is at least one transition t_l that contains an output to T_{shift} (Case 1 of Table 4), T_{shift} is expecting to receive the unique output associated with t_j before it receives the unique output associated with t_l. In the case of a k-shift output fault from t_j to t_{j+k}, T_{shift} will receive the unique output associated with t_l first, and will therefore detect the k-shift output fault.

3.2 If no transition between t_j and t_{j+k} contains an output to T_{shift}, then one of two cases must be true regarding t_{j+1}. If t_{j+1} has an input to the IUT from T_{shift} (Case 2a of Table 4), then T_{shift} would not send that input until it had received the output associated with t_j, and the k-shift would therefore be detected. If t_{j+1} has an input to the IUT from $T_{noshift}$, then a potential undetectable 1-shift output fault would exist between t_j and t_{j+1}. This situation would have resulted in the insertion of an external coordination message exchange relating to observability into the synchronizable global test sequence between t_j and t_{j+1} and would enable T_{shift} to detect the k-shift output fault.

4 If the shift is of the form "$j+k$ to j", then one of two cases can occur. Either there is at least one transition t_l between t_j and t_{j+k} that contains an output to T_{shift} from the IUT, or no transition between t_j and t_{j+k} contains an output to T_{shift} from the IUT.

4.1 If there is at least one transition t_l that contains an output to T_{shift} (see Case 3 of Table 4), T_{shift} is expecting to receive the unique output associated with t_{j+k} after it receives the unique output associated with t_l. In the case of a k-shift output fault from t_{j+k} to t_j, T_{shift} will receive the unique output associated with t_{j+k} first, and will therefore detect the k-shift output fault.

4.2 If no transition between t_j and t_{j+k} contains an output to T_{shift} (see Case 4 of Table 4), then a potential undetectable 1-shift output fault would exist between t_{j+k-1} and t_{j+k}. This situation would have resulted in the insertion of an external coordination message exchange relating to observability into the synchronizable global test sequence between t_{j+k-1} and t_{j+k}, and would enable T_{shift} to detect the k-shift output fault.

Q.E.D.

Theorem 3: There exists an FSM M whereby a synchronizable global test sequence constructed from M may contain undetectable k-shift output faults, $k \geq 2$, if the outputs associated with each transition within that synchronizable global test sequence are not unique.

Proof: If the assumption regarding unique outputs is removed, k-shift output faults may be undetected when the synchronizable global test sequence is applied. Consider the following test subsequence (input and output languages are the same as in Table 4), which is the label *a*/<-, 2>, *a*/<1, 2>, *b*/<1, -> of a transition sequence $t1$, $t2$, $t3$.

A shift of the output "1" from $t3$ to $t1$ may be undetected if the Upper Tester manages to send the input "*a*" for $t2$ before receiving the shifted output. The Upper Tester could interpret this shifted output as the one generated by $t2$, and the second output "1" in $t2$ as the one generated by $t3$, thus passing the faulty implementation. The shift would be detectable if the shifted output was received prior to sending the input "a" in $t2$, but this detectability could not be assured without the use of a global clock.

Q.E.D.

Table 4. Examples of k-shift output faults

$2p$-FSM
$\Sigma_U = a, \Sigma_L = b$
$\Gamma_U = 1_1, 1_2, 1_3, 1_4, \ldots \Gamma_L = 2_1, 2_2, 2_3, 2_4, \ldots$

T_{shift} in all cases is the Upper Tester.

Assume: k-shift output fault, t_j to t_{j+k}

ex: $t_j = a/<1_j, 2_j>$, $t_{j+k} = b/<-, 2_{j+k}>$, and after the k-shift output fault $t_j = a/<-, 2_j>$, $t_{j+k} = b/<1_j, 2_{j+k}>$.

Case 1:

$a/<1_1, 2_1>, b/<1_2, 2_2>, a/<-, 2_3>, b/<-, 2_4>,$
where $t_l = b/<1_2, 2_2>$

Case 2:

a) $a/<1_1, 2_1>, a/<-, 2_2>, b/<-, 2_3>, b/<-, 2_4>$

b) $a/<1_1, 2_1>, <-O_U, +O_L>, b/<-, 2_2>, b/<-, 2_3>, b/<-, 2_4>$

Assume: k-shift output fault, t_{j+k} to t_j

ex: $t_j = a/<-, 2_j>$, $t_{j+k} = b/<1_{j+k}, 2_{j+k}>$, and after the k-shift output fault $t_j = a/<1_{j+k}, 2_j>$, $t_{j+k} = b/<-, 2_{j+k}>$

Case 3:

$a/<-, 2_1>, a/<-, 2_2>, a/<1_3, 2_3>, b/<1_4, 2_4>,$
where $t_l = a/<1_3, 2_3>$

Case 4:

$a/<-, 2_1>, b/<-, 2_2>, b/<-, 2_3>, <-O_U, +O_L>, b/<1_4, 2_4>$

4.6 Extending the Proposed Method to *np*-FSMs

The proposed method is generalized to accommodate np-FSMs, where $n > 2$. Consider the digraph $G = (V, E)$ of an np-FSM M, where n testers interacting with the IUT are labelled 1, 2, 3, ... , n. In the first phase, a digraph $G' = (V', E')$ is constructed from G according to the following steps:

- Step 1. For each vertex $v_i \in V$:

- create a set of vertices $i^1, i^2, \ldots, i^n$ in V' and,
- $\forall j, k, j \leq n, k \leq n, j \neq k$, create a set of dashed edges $(i^j, i^k; <-C_k,+C_j>)$ and $(i^k, i^j; <-C_j,+C_k>)$ in E' that indicate the external coordination message exchanges relating to observability.

- Step 2. For each edge $e_{jk}=(v_j, v_k; x/y) \in E$ create the following edge(s) in E':
  - $(j^q, k^q; x/y)$, if $x \in \Sigma_q$ and $\nexists$ an $a_r \neq$ - in y, $r \neq q$, $1 \leq r \leq n$.
  - $(j^q, k^q; x/y)$ and $(j^q, k^r; x/y)$, if $x \in \Sigma_q$ and $\forall r, r \neq q, \exists$ an $a_r \neq$ - in y, $1 \leq r \leq n$.
- Step 3. For each vertex $v \in V'$ where only dashed edges are arriving and leaving, remove from E' dashed edges arriving and leaving and then remove v from V'. After this step is complete, the resulting digraph will be known as $G' = (V', E')$.
- Step 4. Initially, let $V'' = V'$ and $E'' = E'$.

  For each vertex $v_i \in V'$, if there exists at least two edges $(v_i, v_j^k; x/y)$ in E', $1 \leq k \leq n$, then
  - create a null vertex v_i^x in V'',
  - create a solid bold edge $(v_i, v_i^x; -/-)$ in E'',
  - create edges $(v_i^x, v_j^k; x/y)$ in E'', $\forall k$ where $(v_i, v_j^k; x/y)$ exists in E' and
  - eliminate edges $(v_i, v_j^k; x/y)$ from E''.

  Any remaining solid edges leaving v_i are made bold. The resulting digraph will be known as $G'' = (V'', E'')$.

In the second phase of the proposed method, the set T_0 of triples $(t_{ij}, t_{jp}, <-O_k, +O_h>)$ where (t_{ij}, t_{jp}) is a transition pair with a potential undetectable 1-shift output fault, and $<-O_k, +O_h>$ represents an external coordination message exchange relating to

observability sent by tester h and received by tester k. The digraph G'' is modified using the information in T_0 to form the digraph $G''' = (V''', E''')$ such that in any traversal of G''' any two adjacent edges in G that are covered by the traversal of G'' will not create a potential controllability or observability problem. T_0 is constructed by the following algorithm:

for each vertex $v_j \in V$, $1 \leq j \leq n$, *do*

for each edge e_{ij} *(say* t_{ij}*)* $= (v_i, v_j; x_j/y_j)$ *entering vertex* v_j *do*

for each edge e_{jp} *(say* t_{jp}*)* $= (v_j, v_p; x_{j+1}/y_{j+1})$ *leaving vertex* v_j *do*

if for some output $a_k \in \Gamma_k$, a_k *is in* y_j *XOR* a_k *is in* y_{j+1}, *AND* $x_{j+1} \notin \Sigma_k$, *then add* $(t_{ij}, t_{jp}, <-O_k, +O_h>)$ *to* T_0, *where h is the tester sending the input* x_{j+1} *to the IUT in* t_{jp} *and k is the tester involved in the shift.*

Note that in this algorithm, for each tester involved in a possible shift in the transition pair (t_{ij}, t_{jp}), a new triple will be added to T_0.

The digraph $G'' = (V'', E'')$ can now be modified to form the digraph $G''' = (V''', E''')$, using the information in T_0:

- Step 1. For each $(t_{ij}, t_{jp}, <-O_k, +O_h>)$ triple in T_0, identify the vertex v_j in V'' such that v_j is $tail(t_{ij})$. Add a new vertex v_j* in V''' (if one does not exist already).
- Step 2. A dashed edge from v_j* to v_j is inserted in E''', with the label "t_{ij}, t_{jp}, $<-O_k, +O_h>$" for each $(t_{ij}, t_{jp}, <-O_k, +O_h>)$ in T_0. This dashed edge indicates an external coordination message exchange relating to observability. Any additional triples $(t_{ij}, t_{jp}, <-O_r, +O_s>)$ in T_0 will have the label "$<-O_r, +O_s>$" appended onto the label of this edge. If there exists a triple $(t_{lm}, t_{mq}, <-O_t, +O_u>)$ in T_0 where, in order to traverse the transition pair (t_{lm}, t_{mq}), the same dashed edge from v_j* to v_j must be traversed as is

traversed for the transition pair (t_{ij}, t_{jp}), then an alternate label "t_{lm}, t_{mq}, <-O_t, +O_u>" is added to this dashed edge. The alternate label may also be appended with additional labels of coordination message exchanges relating to observability as required.

- Step 3. For each solid edge t_{ij} in E'' whose label is not -/-, if that edge t_{ij} leaves a vertex v in V'' then it will leave the same vertex in V'''. If the edge t_{ij} is also contained in some transition pair (t_{ij}, t_{jk}) in a triple in T_0 and arrives at a vertex v_j in V'', then the edge t_{ij} will go to v_j* in V''', otherwise it will go to v_j in V'''. The resulting digraph will be known as $G''' = (V''', E''')$.
- Step 4. After constructing an RCPT of G''' over the bold edges, post processing of the graph may be required in the case where alternate labels exist for dashed edges indicating coordination message exchanges relating to observability. The correct alternative must be chosen in order to coincide with the transition pair selected in a particular subsequence of the tour. As well, we adopt the convention that any two consecutive transitions on a path in G that would be covered in the same order in a synchronizable traversal of G''', but with the inclusion of external coordination message exchanges relating to both observability and controllability, will have the controllability message exchange processed first. For example, in an RCPT over the bold edges of $G''' = (V''', E''')$ of the 3p-FSM $M7$ shown in Figure 28, the test subsequence (t9, t1) would require only the inclusion of the (<-O_2, +O_1>) coordination message exchange relating to observability, while the test subsequence (t9, t4) would require only the inclusion of (<-C_3, +C_1>, <-O_1, +O_3>, <-O_2, +O_3>).

Consider the digraph $G = (V, E)$ of the 3p-FSM $M7$ shown in Figure 25, where:

- V is the set of vertices $\{A, B, C\}$ representing the states of $M7$,

- The three testers interacting with the IUT are labelled 1, 2, and 3,
- E is the set of directed edges representing the transitions of $M7$, and
- one of the vertices $v_A \in V$ corresponds to the initial state s_0 of $M7$.

Figures 26, 27 and 28 show the digraphs $G' = (V', E')$, $G'' = (V'', E'')$ and $G''' = (V''', E''')$, obtained after applying the above method to G. T_0 for this example is : $\{(t1, t2, <-O_1, +O_2>), (t1, t2, <-O_3, +O_2>), (t1, t6, <-O_2, +O_1>), (t1, t6, <-O_3, +O_1>), (t2, t8, <-O_1, +O_2>), (t2, t8, <-O_3, +O_2>), (t2, t9, <-O_3, +O_1>), (t3, t1, <-O_2, +O_1>), (t3, t1, <-O_3, +O_1>), (t3, t4, <-O_2, +O_3>), (t3, t5, <-O_1, +O_2>), (t4, t5, <-O_1, +O_2>), (t4, t5, <-O_3, +O_2>), (t5, t2, <-O_1, +O_2>), (t5, t7, <-O_2, +O_3>), (t6, t7, <-O_1, +O_3>), (t6, t7, <-O_2, +O_3>), (t7, t3, <-O_1, +O_3>), (t7, t3, <-O_2, +O_3>), (t7, t9, <-O_2, +O_1>), (t8, t3, <-O_1,+O_3>), (t9, t1, <-O_2, +O_1>), (t9, t4, <-O_1, +O_3>), (t9, t4, <-O_2, +O_3>), (t9, t5, <-O_3, +O_2>)\}$.

As is the case when the proposed method is applied to a $2p$-FSM, the method when applied to an np-FSM may also result in a case where two consecutive transitions t_{ij} and t_{jk} on a path in G''' may be selected so that either a coordination message exchange relating to controllability or some combination of controllability and observability message exchanges is inserted between the two. For example, the test subsequence $t8$, $t3$ may be traversed in G''' of $3p$-FSM $M7$ as $t8$, $<-C_3, +C_1>$, $t3$ or $t8$, $<-C_3, +C_2>$, $<-O_1, +O_3>$, $t3$. Similar to the case with the $2p$-FSM, we ensure that the path containing the coordination message exchange(s) relating to observability is selected.

A rural Chinese postman tour over the bold edges of G''' as shown in Figure 28 can then be constructed. This tour will yield a minimum-cost synchronizable global test sequence that uses no external coordination message exchanges relating to controllability,

but seven external coordination message exchanges relating to observability as the label of:

($t1$, <-O_1, +O_2>, <-O_3, +O_2>, $t2$, $t3$, <-O_2, +O_3>, $t4$, <-O_1, +O_2>, <-O_3, +O_2>, $t5$, $t6$, <-O_1, +O_3>, <-O_2, +O_3>, $t7$, $t8$, $t9$).

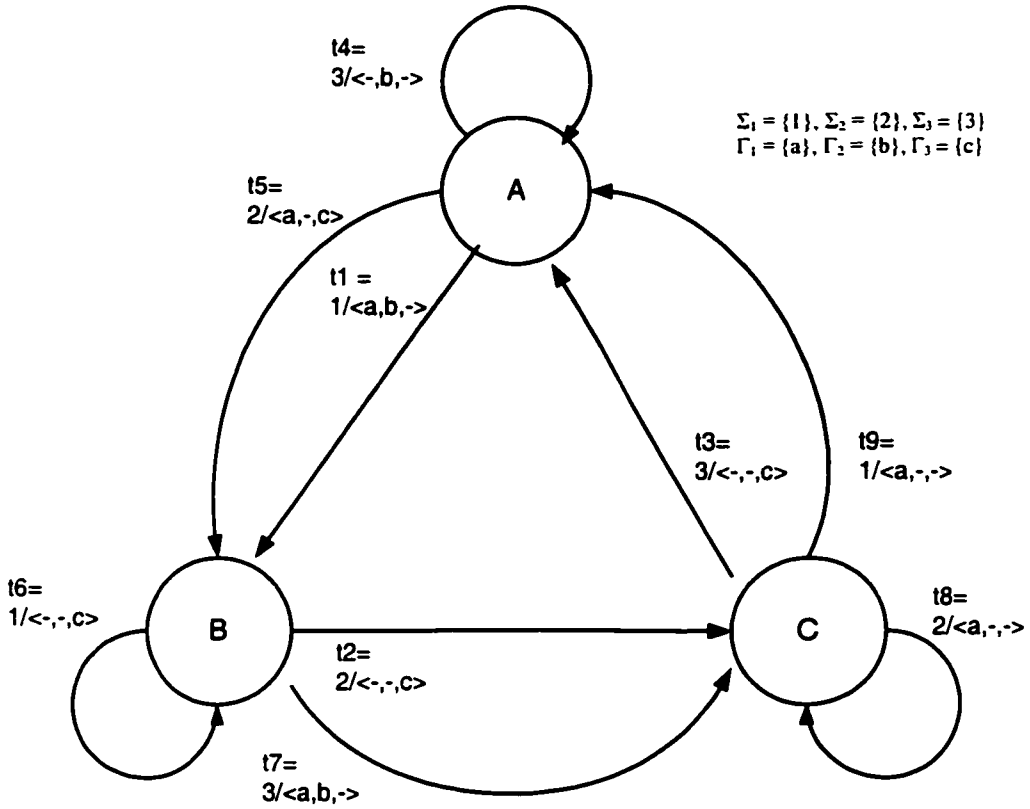

Figure 25. Digraph $G = (V, E)$ of 3p-FSM $M7$

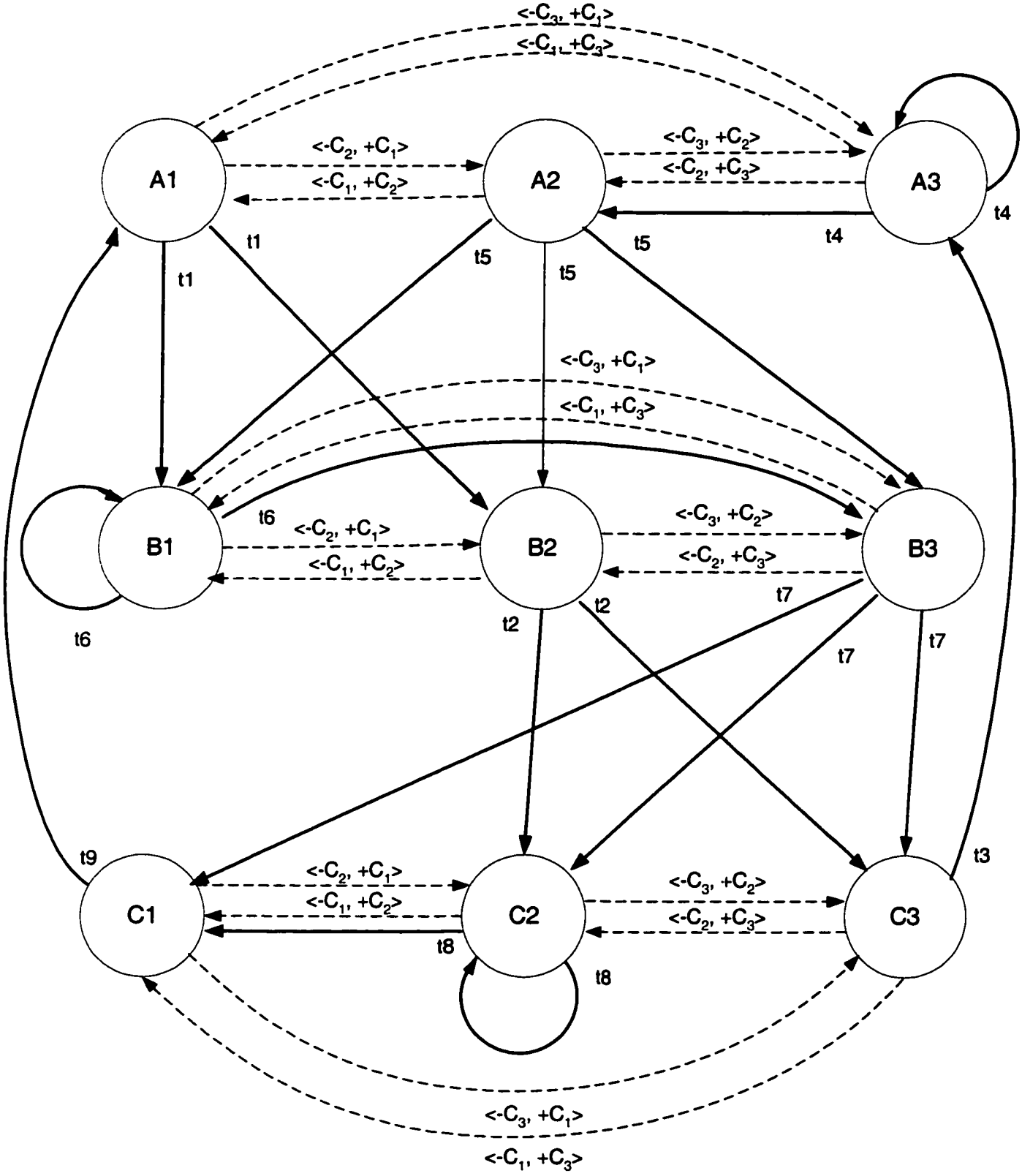

Figure 26. Digraph $G' = (V', E')$ of 3p-FSM $M7$

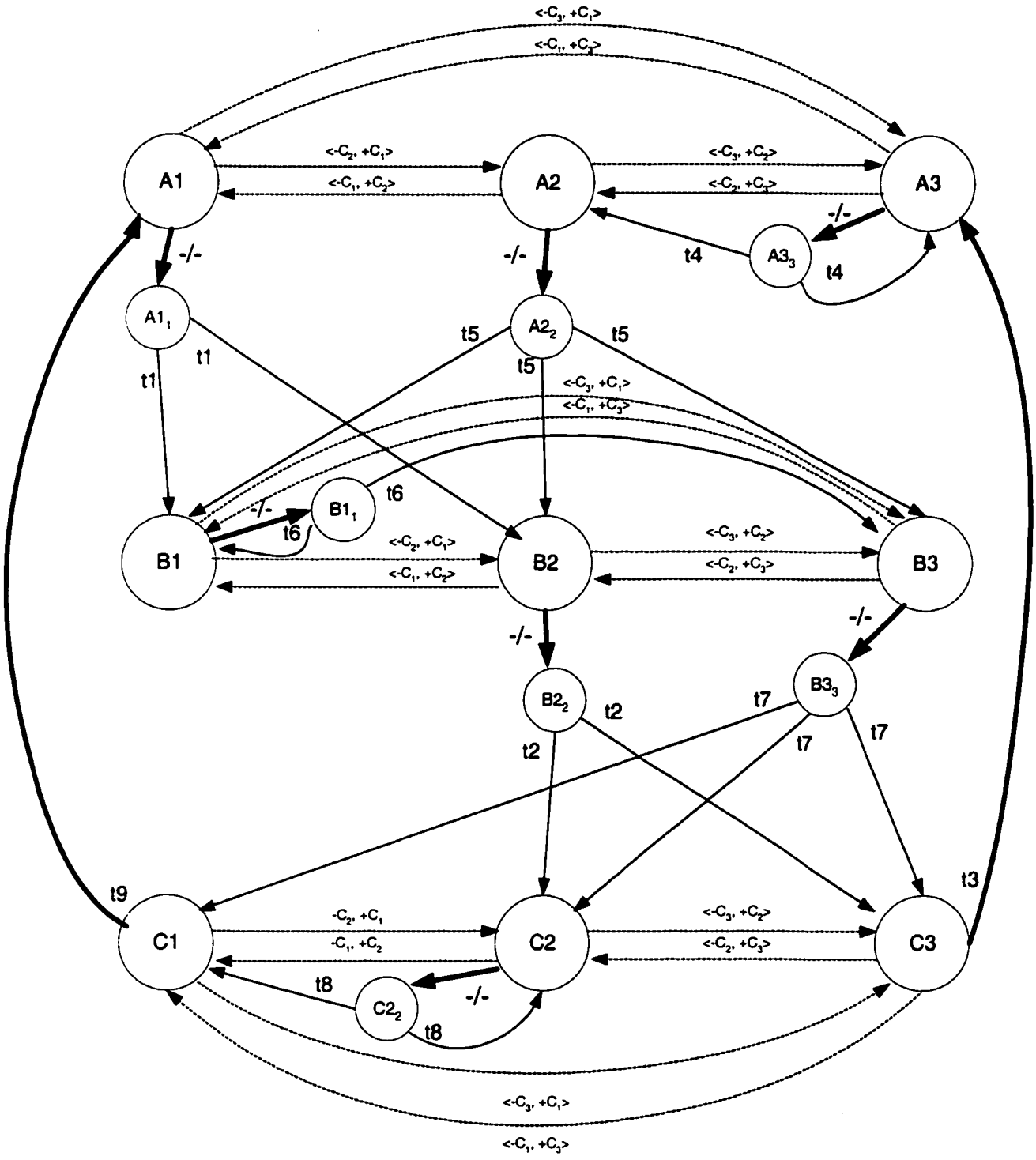

Figure 27. Digraph $G'' = (V'', E'')$ of 3p-FSM $M7$

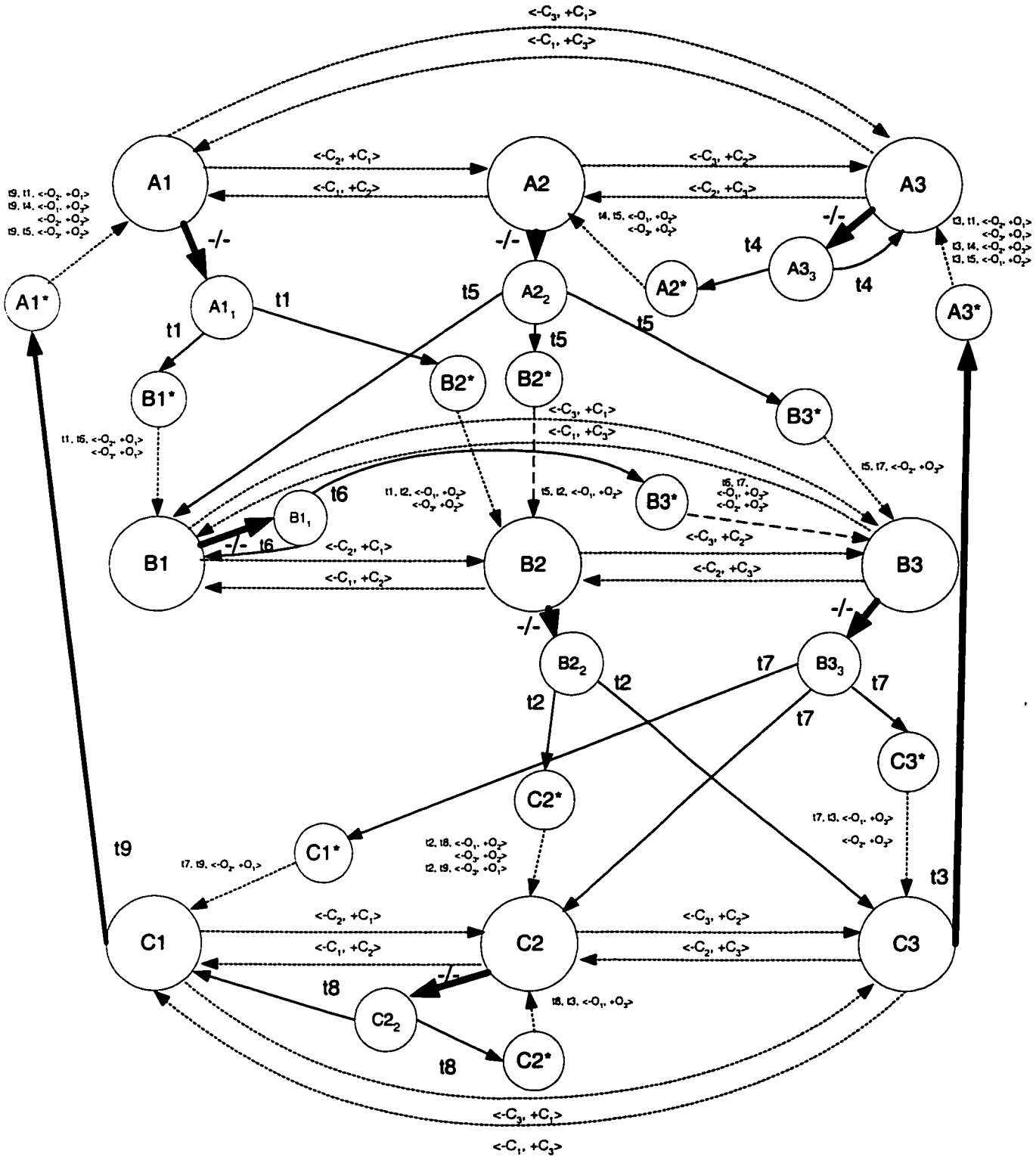

Figure 28. Digraph $G''' = (V''', E''')$ of 3p-FSM $M7$

Chapter Five

Conclusions

5.1 Final Remarks

Two general approaches can be considered in order to solve the problems of controllability and observability that may occur during the application of a global test sequence in a distributed test architecture. In the first general approach, first a global test sequence is generated by applying some test construction method to the given FSM, and then the global test sequence is examined for instances of controllability and observability problems and corrective actions are taken. The corrective actions depend on whether the distributed test architecture being used supports direct communication between testers through the use of external coordination message exchanges, or indirect communication between testers through their interaction with the IUT. In the second general approach, which is introduced in this thesis, controllability and observability problems are considered before a global test sequence is constructed. This minimizes both the length of the global test sequence and the number of corrective actions required to eliminate controllability and observability problems.

We have studied the problem of controllability and observability in distributed testing, proposing a method whereby a minimum-length synchronizable global test sequence with no potential undetectable 1-shift output faults may be generated. If an FSM being implemented is such that a synchronizable global test sequence with no potential undetectable 1-shift output faults exists that does not require an external coordination message exchange, the proposed method will find it. If external

coordination message exchanges are required, then the proposed method can generate a synchronizable global test sequence such that the number of external coordination message exchanges is minimized. The proposed method is shown to be flexible such that it may be used whether the testers in the distributed test architecture are able to communicate amongst themselves directly or indirectly.

5.2 Summary of Contributions

Below we list the major contributions of the thesis:

We have proposed a method that generates a minimum-length synchronizable global test sequence with no possibility of potential observability problems when applied to a $2p$-FSM in a distributed test architecture.

We have proven that, given a $2p$-FSM that is minimal, deterministic and whose underlying graph is strongly connected, our proposed method constructs a synchronizable global test sequence with no potential undetectable 1-shift output faults.

We have shown the proposed method to perform at least as well as other methods proposed in the literature, and we have shown that it performs better than each one under certain conditions.

We have proven that, in a synchronizable global test sequence for a $2p$-FSM, if no potential undetectable 1-shift output faults exist, then there can exist no potential undetectable k-shift output faults, $k > 1$, provided that each output in the sequence is unique.

We have extended the proposed method so that it can be applied to an np-FSM in a distributed architecture, $n > 2$.

5.3 Directions for Future Research

It would be interesting to see this work improved and/or extended in the following directions:

We have shown that, in a synchronizable global test sequence for a $2p$-FSM, if no potential undetectable 1-shift output faults exist, then there can exist no potential undetectable k-shift output faults, $k > 1$, provided that each output in the sequence is unique. It would be interesting to determine under what conditions, if any, a similar claim could be made for np-FSMs, $n > 2$.

We have shown that our proposed method can be augmented slightly to produce a synchronizable global test sequence with no potential undetectable 1-shift output faults without using any external coordination message exchanges, using the same test architecture as in [LB94]. The example used in this thesis seemed to indicate that, even though additional potential undetectable 1-shift output faults are introduced by appending test subsequences to the original synchronizable global test sequence, they are rendered detectable by transition pairs that already exist in the original synchronizable global test sequence. It would be interesting to see a set of rules that yields test subsequences that, when appended to the original synchronizable global test sequence, result only in potential 1-shift output faults detectable by transition pairs within the original synchronizable global test sequence.

Our proposed method assumes a fault model that includes only output faults, and assumes transfer faults do not exist. It would be interesting to construct a method that would be effective if the fault model were to be expanded. Specifically, how would we ensure that we could detect every combination of transfer faults and output faults

(particularly 1-shift output faults) in an implementation N of the specification M of a given FSM?

We have given the conditions by which an RCPT of $G''' = (V''', E''')$ over the set of bold edges of G''' may be done in polynomial time. Following a similar method, [CU95] allows for some dashed edges in their derived graph to be made bold in order to increase the likelihood of finding a weakly connected subgraph. It will be interesting to find some sufficient conditions for the existence of a polynomial time algorithm to derive an RCPT over the set of bold edges of G'''.

REFERENCES

[AA88] A. Aho, A. Dahbura, D. Lee, and M. Uyar, "An Optimization Technique for Protocol Conformance Test Generation Based on UIO Sequences and Rural Chinese Postman Tours," *Protocol Specification, Testing, and Verification*, S. Aggarwal and K. Sabnini, eds., North-Holland, Amsterdam, 1988, pp 75-86.

[BM99] M. Benattou, L. Cacciari, R. Pasini, and O. Rafiq, "Principles and Tools for Testing Open Distributed Systems," *Testing of Communicating Systems: Methods and Applications*, G. Csopaki, S. Dibuz, K. Tarney, eds, Kluwer Academic Publishers: Boston, 1999, pp 77-92.

[BU91] S. Boyd and H. Ural, "The synchronization problem in protocol testing and its complexity," *Information Processing Letters*, Vol. 40, pp. 131-136, 1991.

[CR99] L. Cacciari and O. Rafiq, "Controllability and Observability in Distributed Testing," *Information and Software Technology*, vol. 41, pp. 767-780, 1999.

[CU95] W. Chen and H. Ural, "Synchronizable Checking Sequences Based on Multiple UIO Sequences," *IEEE/ACM Transactions on Networking*, Vol 3, pp 152-157, 1995.

[FK95] K. Farooqui, L. Logrippo, J. de Meer, "The ISO reference model for open distributed processing: an introduction," *Computer Networks and ISDN Systems*, vol. 27, pp. 1215-1229, 1995.

[GU95] S. Guyot and H. Ural, "Synchronizable Checking Sequences Based on UIO Sequences," *IFIP IWPTS'95*, Evry, France, pp. 395-407, Sept. 1995.

[HR00] R.M. Hierons, "Generating minimal synchronized test sequences that detect output-shifting faults", *private communication*, Sep. 2000.

[II95] CAN/CSA-ISO/IEC Information technology – Opens Systems Interconnection – Conformance testing methodology and framework, 9646-1, Part 1: General Concepts, 1995.

[IS94] ISO/IEC JTC1/SC21/WG1, Revised Working Draft on Formal Methods in Conformance Testing, 1994.

[IS95] ISO/IEC Open Distributed Processing, Reference Model, 10748, Parts 1-4, 1995.

[LB94] G. Luo, R. Dssouli, G. v. Bochmann, P. Venkataram and A. Ghedamsi, "Test generation with respect to distributed interfaces," *Computer Standards and Interfaces*, vol. 16, pp. 119-132, 1994.

[RA95] K. Raymond and L. Armstrong, *Open Distributed Processing*, Chapman and Hall: London, 1995, pp 1-33.

[SB84] B. Sarikaya and G. v. Bochmann, "Synchronization and Specification Issues in Protocol Testing," *IEEE Transactions on Communications*, Vol. 32, pp 389-395, Apr. 1984.

[TY98] K.C. Tai and Y.C. Young, "Synchronizable Test Sequences of Finite State Machines," *Computer networks and ISDN systems*, Vol 13, pp. 1111-1134, Jul. 1998.

[UW93] H. Ural and Z. Wang, "Synchronizable test sequence generation using UIO sequences," *Computer Communications*, Vol.16, pp. 653-661, 1993.

[YT98] Y.C. Young and K.C. Tai, "Observation Inaccuracy in Conformance Testing with Multiple Testers," *1st IEEE Workshop on Application - Specific Software Engineering and Technology*, pp 80-85, March 1998.