Transition-based coobservability in Distributed Discrete-Event Systems

by

Ying Huang

A thesis submitted to the
Department of Electrical and Computer Engineering
in conformity with the requirements for
the degree of Master of Science (Engineering)

Queen's University
Kingston, Ontario, Canada
June 2005

Abstract

For a given discrete-event plant G, to find decentralized supervisors that under their control the system behaves exactly like the legal behavior E, there are two conditions: controllability and coobservability. When E is controllable and observable but not coobservable, communications of event occurrences between supervisors will be needed. A new property called *transition-based coobservability* is defined and it is proven that transition-based coobservability together with controllability are necessary and sufficient conditions for solving the decentralized problem. To check transition-based coobservability, an automaton, called a modified M-machine, is constructed. It is proven that the system is not transition-based coobservable if and only if there is a path from the initial state to the marked state in this automaton. An algorithm is also given to compute a communication protocol between supervisors so that the system is transition-based coobservable and the communications are consistent.

Acknowledgments

I would like to thank my supervisor Dr. Karen Rudie for her constant support, encouragement and patience. Without her, this thesis would not have been completed. Her enthusiasm and integral view of research has made a deep impression on me. Besides being an excellent supervisor, Karen was as close as a good friend to me. I am glad that I have come to get know her in my life.

I am indebted to Dr. Feng Lin from Wayne State University whose stimulating suggestions helped me start this thesis. He also took effort in reading and providing me with valuable comments on earlier version of this thesis.

I want to thank the DES group members—Lenko Grigorov, Michael Wood, Sarah Whittaker, and Iakov Romanovski—for all their help, support, interest and valuable comments to my work.

I appreciate the examiners of my thesis, especially Dr. K. Salomaa, for their valuable comments during my defense.

I would like to thank Bernice Ison for her cheerful assistance.

I am grateful for receiving the McLaughlin Fellowship and additional financial support from the Graduate School at Queen's University.

Especially, I would like to give my special thanks to my husband for his love and support and my baby girl Gloria who shows me what a miracle a life can be.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

People used beacon towers to deliver messages more than 2000 years ago. Telephone, one of the most important communication styles nowadays, was invented by Graham Bell in 1876. Today the internet makes the world smaller and more interesting by allowing the exchange of information in no time. Communication, as an incredibly powerful force, is transforming human life from ancient to modern. In this thesis, we show how communication can be applied to a type of discrete-event system (DES) to help solve its control problem.

A DES is a system that changes its state upon the occurrence of an event. The states in the system have symbolic values instead of numerical values as in traditional continuous systems. Almost 30 years ago, Ramadge and Wonham initiated the framework of modelling and synthesis of controllers (supervisors) for *discrete-event systems* [8]. The standard formulation of DES control has been widely extended in a number of ways which include modular supervisory control, hierarchical supervisory control, timed DES, dynamic DES and partial observation control. We deal with partial observation DES control in this thesis.

1.1 Main Contributions

The main contributions of this thesis are:

- A definition of transition-based coobservability. It is proved to be a necessary property to solve the decentralized supervisory control problem when communications between supervisors are involved.
- A method for checking transition-based coobservability.
- An algorithm to find a consistent communication protocol between two supervisors so that they cooperate to ensure legal behavior.

1.2 Thesis Outline

Chapter 2 presents some background material on discrete-event systems in general, with the focus on supervisory control with partial observation. A definition of transition-based coobservability is given. Chapter 3 gives a theorem that demonstrates the relationship between transition-based coobservability and decentralized supervisory control problems. The proof of the theorem is also given in this section, along with a simple example. Chapter 4 presents a special automaton named a modified M-machine which can be used to check the property of transition-based coobservability. Chapter 5 gives an algorithm that uses the modified M-machine to find a consistent communication solution which solves a type of decentralized supervisory control problem. The last Chapter contains discussion of the thesis and possible future work.

Chapter 2

Background and Related Concepts

2.1 Modelling Discrete-Event Systems

A DES is an abstract process that is being characterized by sequences of actions or events. When the system operates freely, without any interference or control, we call it a *plant*. A plant may generate some undesirable sequences, called *illegal behavior*. There are several formalisms used to model DESs. The most explicit model is the *finite state machine* (FSM). A FSM always resides in one of its finite number of *states*. The transitions between states are associated with *events*. The alphabet represents the set of all events that can possibly happen in the plant.

Formally a plant is modelled as

$$G = (\Sigma, Q^G, \delta^G, q_0^G)$$

where Σ is the alphabet, Q^G is a finite set of states, δ^G is a *transition function* $\Sigma \times Q^G \to Q^G$, and q_0^G is the initial state of the plant.

For any event $\sigma \in \Sigma$, $q^G \in Q^G$, when we say $\delta^G(\sigma, q^G)$ is defined we mean when the plant resides at state q^G, event σ may happen and after σ happens the plant will change to state $\delta^G(\sigma, q^G)$. When $\delta^G(\sigma, q^G)$ is defined, we write $\delta^G(\sigma, q^G)!$.

The transition function δ^G can be naturally extended to a partial function $\Sigma^* \times Q^G$ such that

$$\delta^G(\epsilon, q^G) := q^G$$

$$(\forall t \in \Sigma^*)(\forall \sigma \in \Sigma)\, \delta^G(t\sigma) := \delta^G(\sigma, \delta^G(t, q_0^G))$$

where ϵ represents the empty string and Σ^* represents all possible finite strings over Σ union ϵ.

The language generated by a plant G (also called the closed behavior of G), denoted by $L(G)$, is a language

$$L(G) := \{t \,|\, t \in \Sigma^*, \delta^G(t, q_0^G)!\}$$

This language represents all possible sequences that the plant G can generate.

If we denote concatenation of two strings s and t by st, then s is called a *prefix* of the string st. Therefore each nonempty string $s \in \Sigma^*$ has at least two prefixes: ϵ and s.

For a language $L \in \Sigma^*$, $\bar{L}$ is the set of all prefix of strings in L, formally it is defined as

$$\bar{L} = \{s \in \Sigma^* \,|\, \exists t \in \Sigma^*,\, st \in L\}$$

A language is *prefix-closed* if $L = \bar{L}$. The language generated by a plant G is

always prefix-closed.

An uncontrolled plant may generated undesired behaviors. We use an automaton, called E, to represent the legal behavior of a plant G:

$$E = (\Sigma, Q^E, \delta^E, q_0^E)$$

The language generated by E is denoted by $L(E)$.

To force a plant G to behave in a legal way, we need a controller or supervisor, named S, to control some event occurrences based on its view of the plant's behavior. The supervisor may not have the ability to control all the events in the alphabet Σ, therefore Σ is partitioned into two disjoint subsets Σ_c and Σ_{uc}, which comprise the *controllable events* and the *uncontrollable events*, respectively.

Formally, a supervisor S is a pair (T, ψ), where T is an automaton and ψ is a feedback map:

$$T = (\Sigma, X, \xi, x_0)$$

where Σ is the same alphabet as in G, X is the state space of the supervisor, ξ is the transition function, and x_0 is the initial state.

The feedback map ψ is defined as follows:

$$\psi : \Sigma \times X \to \{0, 1\}$$

$$\text{For all } \sigma \in \Sigma_{uc},\ x \in X,\ \psi(\sigma, x) = 1$$

$$\text{For all } \sigma \in \Sigma_c,\ x \in X,\ \psi(\sigma, x) \in \{0, 1\},$$

where the number 0 represents a disable control action and the number 1 represents

an enable action. The supervisor S keeps track of events occurring in G. The feedback map indicates whether σ should be disabled at the corresponding state in G.

With the supervision of S, a plant G behaves in a constrained way which is described by an automaton S/G:

$$S/G := (Q \times X, \Sigma, (\delta \times \xi)^{\psi}, (q_0, x_0))$$

where $(\delta \times \xi)^{\psi} : \Sigma \times Q \times X \to Q \times X$ is defined as follows:

$$(\delta \times \xi)^{\psi}(\sigma, q, x) := \begin{cases} (\delta(\sigma, q), \xi(\sigma, x)), & \text{if } \delta(\sigma, q)!, \ \xi(\sigma, x))! \text{ and } \psi(\sigma, x) = 1; \\ \text{undefined}, & \text{otherwise.} \end{cases}$$

The centralized control problem was introduced by Ramadge and Wonham [6]:

> Given a plant G over alphabet Σ (with controllable events Σ_c), given some language $L(E) \subseteq L(G)$, find a supervisor S such that $L(S/G) = L(E)$.

It was proven that the necessary and sufficient condition to solve the above problem is *controllability*. A language $K \subseteq L(G)$ is a *controllable sublanguage* of $L(G)$ if

$$\bar{K}\Sigma_{uc} \cap L(G) \subseteq \bar{K}$$

If we think of K as the legal language $L(E)$, controllability can be explained informally as follows: the legal language is controllable if for any sequence t that starts as a legal sequence ($t \in \bar{K}$), the occurrence of an uncontrollable event ($\sigma \in \Sigma_{uc}$)

following that sequence that is possible ($s\sigma \in L(G)$)) will not lead the sequence out of the legal range ($s\sigma \in \bar{K}$).

2.2 Partial Observation Control

In the real world, a supervisor may not observe all the events that a plant generates due to distance, environment or other reasons. The alphabet Σ is partitioned into two disjoint sets: Σ_o (observable events) and Σ_{uo} (unobservable events),where

$$\Sigma = \Sigma_o \cup \Sigma_{uo}$$

Supervisory control under partial observation was proposed by Lin and Wonham [5]. To formally describe what a supervisor "sees", we use a mapping called the *natural (canonical) projection* $P : \Sigma^* \to \Sigma_o^*$ which is defined as follows:

$$\begin{aligned} P(\epsilon) &= \epsilon \\ P(\sigma) &= \epsilon, \quad \sigma \in \Sigma \backslash \Sigma_o \\ P(\sigma) &= \sigma, \quad \sigma \in \Sigma_o \\ P(t\sigma) &= P(t)P(\sigma), \quad t \in \Sigma^*, \sigma \in \Sigma \end{aligned}$$

Informally speaking, P erases all unobservable events from a sequence and keeps observable events in the same order.

A prefix-closed language K is *observable* with respect to G and P if for all $t, t' \in$

Σ^*, $\sigma \in \Sigma$,

$$P(t) = P(t') \Rightarrow (t'\sigma \in K \wedge t \in K \wedge t\sigma \in L(G) \Rightarrow t\sigma \in K)$$

Decentralized discrete-event system control problem was first proposed in [2] and [13]. This type of problem is in a setting when a plant needs more than one partial observation supervisor to achieve the desired behavior. Suppose we need n supervisors and Supervisor i only observes some set $\Sigma_{i,o} \in \Sigma$, projection P_i is defined for each Supervisor i, where $i = 1, ..., n$:

$$\begin{aligned} P_i(\epsilon) &= \epsilon \\ P_i(\sigma) &= \epsilon, \quad \sigma \in \Sigma \backslash \Sigma_{i,o} \\ P_i(\sigma) &= \sigma, \quad \sigma \in \Sigma_{i,o} \\ P_i(t\sigma) &= P_i(t)P_i(\sigma), \quad t \in \Sigma^*, \sigma \in \Sigma \end{aligned}$$

When a supervisor can only control some subset of controllable events, we call it a *local* supervisor; otherwise, it is a *global* one. The set of controllable events of a supervisor may not be the same as the set of observable events, i.e., a supervisor has the ability to disable an event even if it cannot "see" it.

In this thesis, we consider two local supervisors $S_1 = (T_1, \phi)$ and $S_2 = (T_2, \psi)$ acting on a plant G, where $T_1 = (X, \Sigma, \delta, x_0)$ and $T_2 = (Y, \Sigma, \eta, y_0)$. The *conjunction* of S_1 and S_2 [7] is the supervisor

$$S_1 \wedge S_2 := (T_1 \times T_2, \phi * \psi),$$

where

$$T_1 \times T_2 := (X \times Y, \Sigma, \delta \times \eta, (x_0, y_0))$$

The new supervisor's transition function and feedback map are defined as follows. For $\sigma \in \Sigma$, $x \in X$ and $y \in Y$,

$$(\delta \times \eta)(\sigma, x, y) := \begin{cases} (\delta(\sigma, x), \eta(\sigma, y)), & \text{if } \delta(\sigma, x)! \wedge \eta(\sigma, y)!; \\ \text{undefined}, & \text{otherwise.} \end{cases}$$

$$(\phi \times \psi)(\sigma, x, y) := \begin{cases} \text{disable}, & \text{if } \phi(\sigma, x) = \mathit{disable} \vee \psi(\sigma, y) = \mathit{disable}\ ; \\ \text{enable}, & \text{otherwise.} \end{cases}$$

Under the control of $S_1 \wedge S_2$, an event is allowed to happen only when both supervisors enable it; if either S_1 or S_2 issues a disablement command, the event will be disabled.

The decentralized problem is described in [13]:

> Given a plant G over an alphabet Σ, a prefix-closed language $L(E)$ such that $\emptyset \neq L(E) \subseteq L(G)$, and sets $\Sigma_{1,c}$, $\Sigma_{2,c}$, $\Sigma_{1,o}$, $\Sigma_{2,o} \subseteq \Sigma$, do there exist supervisors S_1 and S_2 such that $S_1 \wedge S_2$ is a supervisor for G and
>
> $$L(S_1 \wedge S_2/G) = L(E)$$
>
> .
>
> Here, for $i = 1, 2$, local supervisor S_i can observe only events in $\Sigma_{i,o}$ and can control only events in $\Sigma_{i,c}$.

This problem is solved in [2, 13]. Supervisors exist if the system satisfies *controllability* and *coobservability*. The definition of controllability can be found in [6]. Informally, it says that events that could take a sequence outside the legal language can always be disabled.

Given G over an alphabet Σ, sets $\Sigma_{1,c}$, $\Sigma_{2,c}$, $\Sigma_{1,o}$, $\Sigma_{2,o} \subseteq \Sigma$, projection $P_1 : \Sigma^* \to \Sigma_{1,o}^*$, $P_2 : \Sigma^* \to \Sigma_{2,o}^*$, a prefix-closed language $K \subseteq L(G)$ is *coobservable* with respect to G, P_1, P_2 if for all $s, s', s'' \in \Sigma^*$,

$$P_1(s) = P_1(s') \wedge P_2(s) = P_2(s'') \Longrightarrow$$

$$(\forall \sigma \in \Sigma_{1,c} \cap \Sigma_{2,c})\, s \in L(E) \wedge s\sigma \in L(G) \wedge s'\sigma, s''\sigma \in L(E) \Rightarrow s\sigma \in L(E) \quad \text{(conjunct 1)}$$

$$\wedge\, (\forall \sigma \in \Sigma_{1,c} \setminus \Sigma_{2,c})\, s \in L(E) \wedge s\sigma \in L(G) \wedge s'\sigma \in L(E) \Rightarrow s\sigma \in L(E) \quad \text{(conjunct 2)}$$

$$\wedge\, (\forall \sigma \in \Sigma_{2,c} \setminus \Sigma_{1,c})\, s \in L(E) \wedge s\sigma \in L(G) \wedge s''\sigma \in L(E) \Rightarrow s\sigma \in L(E) \quad \text{(conjunct 3)}$$

Since supervisors only have partial observation of the plant, they may not distinguish sequences which look the same to them but require different control actions. With this in mind, we can interpret coobservability as follows. If the event needed to be disabled is controllable by both supervisors (i.e., conjunct 1), then we only need one of the supervisors to be able to have a clear view of the strings s, s', s'' to take control action of σ. If the event is only controllable by one supervisor (i.e., conjunct

2 and conjunct 3), then that supervisor's view must be sufficient to make the control decision.

2.3 Transition-Based Coobservability

When controllability and observability are satisfied but coobservability is not satisfied in decentralized problems, it means that if given more information supervisors could have a clearer view of the plant and make the right control actions. One of the solutions is to involve communications between two supervisors, i.e., upon an event occurring, the supervisor who can "see" it communicates the event label to the other supervisor so that the other supervisor also "knows" it is happening. In this thesis, we sometimes use an expression such as "a transition $(1, \alpha, 2)$ is communicated from Supervisor 1 to Supervisor 2", which is shorthand for "when Supervisor 1 observes the occurrence of α from state 1 (which will lead Supervisor 1 to state 2), Supervisor 1 will communicate symbol α to Supervisor 2".

With communications, a supervisor has a direct observation through its own view and also it has an indirect "observation" by the communications of specific event occurrences from other supervisors. We can, therefore, say the observation is no longer based on events but on transitions. A supervisor may not see α from state a but may "see" α from state b. According to this change, we modified projection P to a *transition-based projection* P_G so that it still captures the supervisor's view.

Given an automaton $G = (\Sigma, Q^G, \delta^G, q_0^G)$, we identify some of its transitions as observable and the rest as unobservable. This yields two disjoint sets: δ_o, the set of observable transitions, and δ_{uo}, the set of unobservable transitions.

Transition-based projection $P_G : \Sigma \times Q^G \to \Sigma$ is defined as follows.

For $\sigma \in \Sigma$, $q \in Q^G$,

$$P_G(\epsilon, q) := \epsilon$$

$$P_G(\sigma, q) := \begin{cases} \sigma & \text{if } (q, \sigma, \delta^G(q, \sigma)) \in \delta_o \\ \epsilon & \text{otherwise} \end{cases}$$

P_G can be extended to a function on $\Sigma^* \times Q^G$ as follows.

For $s \in \Sigma^*$, $\sigma \in \Sigma$, $q \in Q^G$,

$$P_G(s\sigma, q) = P_G(s, q)P_G(\sigma, \delta^G(s, q))$$

That is, the transition-based projection of a sequence is made up of the transition-based projection for each event occurring along that sequence. For example, as shown in Figure 2.1, in the sequence $\alpha\beta\gamma\alpha$, $(1, \beta, 2)$ and $(3, \alpha, 4)$ are unobservable,

$$\begin{aligned} P_G(\alpha\beta\gamma\alpha, 0) &= P_G(\alpha, 0)P_G(\beta, 1)P_G(\gamma, 2)P_G(\alpha, 3) \\ &= \alpha\gamma \end{aligned}$$

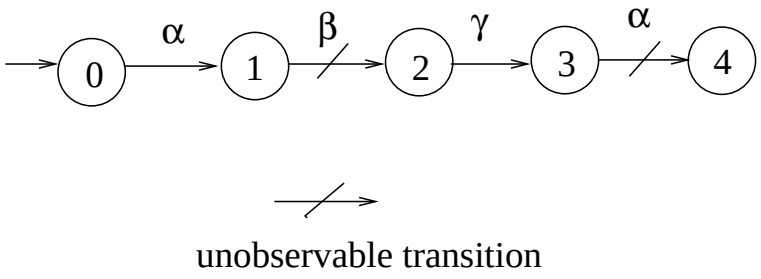

Figure 2.1: An Example of Transition-Based Projection

Coobservability can also be modified to *transition-based coobservability*. The definition of transition-based coobservability is as follows. Given a plant G over alphabet Σ, a legal automaton E such that $L(E) \subseteq L(G)$, sets $\Sigma_{1,c}, \Sigma_{2,c} \subseteq \Sigma$, transition-based projections $P_{1,G} : \Sigma^* \times Q^E \to \Sigma^*$, $P_{2,G} : \Sigma^* \times Q^E \to \Sigma^*$, the language $L(E)$ is transition-based coobservable w.r.t. $G, P_{1,G}, P_{2,G}$ if for all $s, s', s'' \in \Sigma^*$,

$$
\begin{aligned}
&P_{1,G}(s, q_0) = P_{1,G}(s', q_0) \wedge P_{2,G}(s, q_0) = P_{2,G}(s'', q_0) \Longrightarrow \\
&\qquad (\forall \sigma \in \Sigma_{1,c} \cap \Sigma_{2,c})\, s \in L(E) \wedge s\sigma \in L(G) \wedge s'\sigma, s''\sigma \in L(E) \Rightarrow s\sigma \in L(E) \\
&\qquad \wedge (\forall \sigma \in \Sigma_{1,c} \setminus \Sigma_{2,c})\, s \in L(E) \wedge s\sigma \in L(G) \wedge s'\sigma \in L(E) \Rightarrow s\sigma \in L(E) \\
&\qquad \wedge (\forall \sigma \in \Sigma_{2,c} \setminus \Sigma_{1,c})\, s \in L(E) \wedge s\sigma \in L(G) \wedge s''\sigma \in L(E) \Rightarrow s\sigma \in L(E)
\end{aligned}
$$

Chapter 3

Control with Unobservable Transitions

3.1 Motivation

To motivate our formulation of decentralized supervisory control with transition-based observability, we consider the automaton in Figure 3.1 as the legal automaton E, and the plant to be controlled as displayed in Figure 3.2.

To construct a proper supervisor, a supervisor must satisfy the condition that if a transition is unobservable to the supervisor, then it will not change state upon the occurrence of this transition. By intuition, we construct supervisors by replacing the event labels of all unobservable transitions in E with ϵ, which will produce a nondeterministic finite-state automaton (NFA). Then we convert the NFA to an equivalent deterministic finite-state automaton (DFA) [3] and at each state add self-loops for events left undefined by the partial transition function. This yields a supervisor whose transition function is now fully defined, as displayed in Figure 3.3.

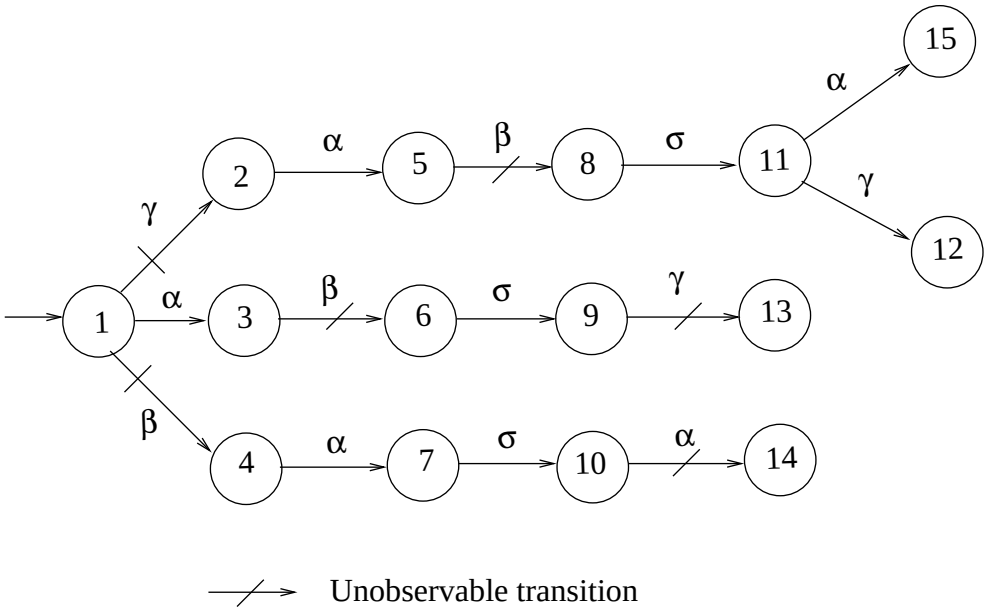

Figure 3.1: Legal Automaton E

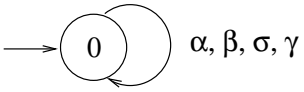

Figure 3.2: Plant G

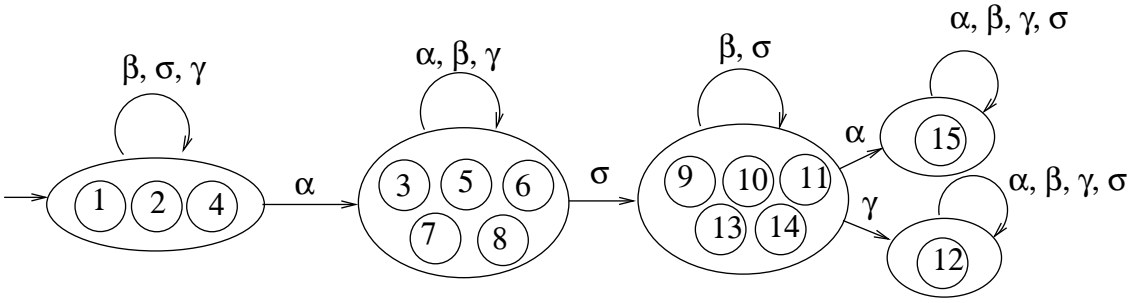

Figure 3.3: Supervisor

If two sequences have the same transition-based projection, then they should lead to the same state in the supervisor, because if the supervisor cannot distinguish two strings, it should take the same control action upon the occurrence of either sequence. In the above example, $\alpha\beta\sigma\gamma$ and $\beta\alpha\sigma\alpha$ look the same to the supervisor, i.e., from its point of view $\alpha\sigma$ is observed in both cases. However, in the structure of the supervisor, $\alpha\beta\sigma\gamma$ leads to state $\{12\}$ and $\beta\alpha\sigma\alpha$ leads to state $\{15\}$. The reason for this confusion is because a sequence the supervisor recognized does not correspond to the same sequence that occurred in the system. After the two sequences happen in the system, the supervisor should actually reach the same state, namely {(9),(10),(11),(13),(14)}, instead of two different states. Consequently, in the closed-loop system, we will consider where in a supervisor the projected version of a sequence leads, not where the original sequence leads. The next section details our decentralized supervisory controlled system.

3.2 Transition-Based DES Control

Consider a discrete-event system to be controlled that is represented by an automaton

$$G = (\Sigma, Q^G, \delta^G, q_0^G)$$

The legal behavior of the system is also characterized by an automaton

$$E = (\Sigma, Q^E, \delta^E, q_0^E)$$

where all states in Q^E are reachable.

Since $L(E) \subseteq L(G)$, we can assume that E is a subautomaton of G [1], such that

$$Q^E \subseteq Q^G \tag{3.1}$$

$$q_0^E = q_0^G \tag{3.2}$$

$$\delta^E(t, q_0^E) = \delta^G(t, q_0^G) \text{ , for all } t \in L(E). \tag{3.3}$$

Consider two supervisors acting on G,

$$S_1 = (T_1, \psi_1),\ S_2 = (T_2, \psi_2)$$

where $T_i (i = 1, 2)$ is an automaton

$$T_i = (X_i, \Sigma, \xi_i, x_{0,i})$$

and ψ_i is the feedback map. It is assumed that ξ_i is a fully defined function.

The behavior of the supervised system, denoted by $L(S_1 \wedge S_2/G)$, is defined recursively as follows:

$$\epsilon \in L(S_1 \wedge S_2/G)$$

$$(\forall s \in \Sigma^*, \sigma \in \Sigma) s\sigma \in L(S_1 \wedge S_2/G) \Longleftrightarrow s\sigma \in L(G) \wedge s \in L(S_1 \wedge S_2/G)$$
$$\wedge \psi_1(\sigma, \xi_1((P_{1,G}(s, q_0^E)), x_{0,1})) = enable$$
$$\wedge \psi_2(\sigma, \xi_2((P_{2,G}(s, q_0^E)), x_{0,2})) = enable$$

where for $i = 1, 2$, $\xi_i(P_{i,G}(s, q_0^E), x_{0,i})$ is the transition function associated with S_i with transition-based projection as one of its argument. Since supervisor S_i will make transitions based partly on events it directly observes and partly on event occurrences communicated to it, we will not be interested in whether S_i recognizes a sequence s (in the automata theory sense) but rather to which state the "visible" events lead. The function $\xi_i(P_{i,G}(s, q_0^E), x_{0,i})$ captures this notion, i.e., it defines where in S_i the visible part of a sequence leads.

The closed-loop behavior $L(S_1 \wedge S_2/G)$ is governed not only by G and the supervisors, but also by where a sequence of events leads to in the supervisors. Since two sequences that have the same transition-based projection will lead to the same state in the supervisor, the decentralized controlled system is implemented in a feasible way.

The prototype of Theorem 1 is Theorem 4.1 in [13].

Theorem 1. *Given a plant G and a nonempty prefix-closed legal language $L(E)$, there exist two supervisors S_1 and S_2 who have transition-based partial observation of the system, and guarantee $L(S_1 \wedge S_2/G) = L(E)$ if and only if $L(E)$ is controllable and transition-based coobservable with respect to G and $P_{i,G}$.*

Proof

First we prove necessity: If there exist supervisors S_1, S_2 that make $L(S_1 \wedge S_2/G) =$

$L(E)$, then $L(E)$ is controllable and transition-based coobservable.

To show that $L(E)$ is controllable, we need

$$L(\bar{E})\Sigma_{uc} \cap L(G) \subseteq L(\bar{E})$$

Take $s \in L(\bar{E})$, $\sigma \in \Sigma_{uc}$, such that $s\sigma \in L(G)$

$$\begin{aligned} s \in L(\bar{E}) &\Rightarrow s \in L(E) \qquad\qquad (L(E) = L(\bar{E}) \text{ by assumption}) \\ &\Rightarrow s \in L(S_1 \wedge S_2/G) \text{ and } \psi_i(\sigma, \xi_i(P_{i,G}(s, q_0^E), x_{0,i})) = \mathit{enable}, \text{ for } i = 1, 2 \\ &\quad (\text{Since } \sigma \in \Sigma_{uc}) \\ &\Rightarrow s\sigma \in L(S_1 \wedge S_2/G) \\ &\Rightarrow s\sigma \in L(E) \end{aligned}$$

Secondly, we prove that $L(E)$ is transition-based coobservable with regard to $P_{1,G}$ and $P_{2,G}$.

By contradiction, suppose $L(E)$ is not transition-based coobservable, then there exist $s, s', s'' \in \Sigma^*$, $\sigma \in \Sigma$ such that at least one conjunct in the definition of transition-based coobservability fails to hold.

Conjunct 1 fails:

If conjunct 1 fails, that means for sequences s ,s' and s'' such that $P_{1,G}(s, q_0^E) = P_{1,G}(s', q_0^E)$ and $P_{2,G}(s, q_0^E) = P_{2,G}(s'', q_0^E)$, there exists $\sigma \in \Sigma_{1,c} \cap \Sigma_{2,c}$ such that $s'\sigma, s''\sigma \in L(\bar{E})$, $s \in L(\bar{E})$, $s\sigma \in L(G)$, but $s\sigma \notin L(\bar{E})$.

$$s, s', s'' \in L(\bar{E}) \Rightarrow s, s', s'' \in L(S_1 \wedge S_2/G)$$

$$P_{1,G}(s, q_0^E) = P_{1,G}(s', q_0^E) \Rightarrow \xi_1(P_{1,G}(s, q_0^E), x_{0,1}) = \xi_1(P_{1,G}(s', q_0^E), x_{0,1}) \tag{3.4}$$

$$\begin{aligned} s'\sigma \in L(\bar{E}) &\Rightarrow s'\sigma \in L(S_1 \wedge S_2/G) \\ &\Rightarrow \psi_1(\sigma, \xi_1(P_{1,G}(s', q_0^E), x_{0,1})) = enable \end{aligned} \tag{3.5}$$

Substituting (3.4) into (3.5),we have

$$\psi_1(\sigma, \xi_1(P_{1,G}(s, q_0^E), x_{0,1})) = enable \tag{3.6}$$

Similarly,

$$P_{2,G}(s, q_0^E) = P_{2,G}(s'', q_0^E) \text{ and } s''\sigma \in L(\bar{E}) \Rightarrow \psi_2(\sigma, \xi_2(P_{2,G}(s, q_0^E), x_{0,2})) = enable \tag{3.7}$$

From (3.6), (3.7) , $s\sigma \in L(G)$ and $s \in L(S_1 \wedge S_2/G)$

$$\begin{aligned} &\Rightarrow s\sigma \in L(S_1 \wedge S_2/G) \\ &\Rightarrow s\sigma \in L(\bar{E}) \end{aligned}$$

which contradicts our starting assumption.

Conjunct 2 fails:

If conjunct 2 fails, that means for some sequences s and s' such that $P_{1,G}(s, q_0^E) =$

$P_{1,G}(s', q_0^E)$, there exists $\sigma \in \Sigma_{1,c} \setminus \Sigma_{2,c}$ such that $s'\sigma \in L(\bar{E})$, $s \in L(\bar{E})$, $s\sigma \in L(G)$, but $s\sigma \notin L(\bar{E})$.

$$s, s' \in L(\bar{E}) \Rightarrow s, s' \in L(S_1 \wedge S_2/G)$$

$$\begin{aligned} s\sigma \notin L(\bar{E}) &\Rightarrow s\sigma \notin L(S_1 \wedge S_2/G) \\ &\Rightarrow \psi_1(\sigma, \xi_1(P_{1,G}(s, q_0^E), x_{0,1})) = disable \\ &\quad \text{or } \psi_2(\sigma, \xi_2(P_{2,G}(s, q_0^E), x_{0,2})) = disable \\ &\quad (\text{since } s\sigma \in L(G) \text{ and } s \in L(S_1 \wedge S_2/G)) \end{aligned} \tag{3.8}$$

Because $\sigma \notin \Sigma_{2,c}$, $\psi_2(\sigma, \xi_2(P_{2,G}(s, q_0^E), x_{0,2})) = enable$. Then

$$\psi_1(\sigma, \xi_1(P_{1,G}(s, q_0^E), x_{0,1})) = disable \quad (\text{by (3.8)}) \tag{3.9}$$

$$\begin{aligned} P_{1,G}(s, q_0^E) = P_{1,G}(s', q_0^E) &\Rightarrow \xi_1(P_{1,G}(s, q_0^E), x_{0,1}) = \xi_1(P_{i,G}(s', q_0^E), x_{0,1}) \\ &\Rightarrow \psi_1(\sigma, \xi_1(P_{i,G}(s, q_0^E), x_{0,1})) = \psi_1(\sigma, \xi_1(P_{i,G}(s', q_0^E), x_{0,1})) \\ &\Rightarrow \psi_1(\sigma, \xi_1(P_{i,G}(s', q_0^E), x_{0,1})) = disable \quad (\text{by (3.9)}) \\ &\Rightarrow s'\sigma \notin L(S_1 \wedge S_2/G) \\ &\Rightarrow s'\sigma \notin L(\bar{E}) \quad (\text{contradiction of the assumption } s'\sigma \in \bar{E}) \end{aligned}$$

Conjunct 3 fails: analogous to the case of conjunct 2 failing.

Now we proof sufficiency: If $L(E)$ is controllable and transition-based coobservable with regard to G, $P_{1,G}$ and $P_{2,G}$, then there exist two supervisors S_1 and S_2 that guarantee $L(S_1 \wedge S_2/G) = L(E)$.

Before the proof, we need to construct supervisors S_1 and S_2.

For $i = 1, 2$,

$$T_i := (X_i, \Sigma, \xi_i, x_{0,i})$$

where X_i is the set of nonempty subsets of Q^E, Σ is the same alphabet as the plant G. Transition function ξ_i and initial state $x_{0,i}$ are defined as follows.

For all $\sigma \in \Sigma$, for all $x_i \in X_i$,

$$\xi_i(\sigma, x_i) := \begin{cases} \{\delta^E(s,q) | s \in \Sigma^*, q \in x_i, P_{i,G}(s,q) = \sigma\} \\ \qquad \text{if this is not empty} \\ x_i \qquad \text{otherwise} \end{cases}$$

$$x_{0,i} := \{\delta^E(s, q_0^E) | s \in \Sigma^*, P_{i,G}(s, q_0^E) = \epsilon\}$$

The feedback map ψ_i is defined as follows:

$$(\forall \sigma \in \Sigma_{i,c}, x_i \in X_i)\psi_i(\sigma, x_i) = enable \Leftrightarrow (\exists q \in x_i)\, \delta^E(\sigma, q) \text{ is defined.}$$

$$(\forall \sigma \notin \Sigma_{i,c}, x_i \in X_i)\psi_i(\sigma, x_i) = enable$$

The controllers S_1 and S_2 constructed here keep track of all strings that could have been generated so far.

The following claim indicates that if s is a sequence in the legal language E, the state where s leads to is a substate of where the transition-based projection of s leads to in the transition structure of supervisor S_i.

Claim 1. *For all $s \in \Sigma^*$,*

$$\delta^E(s, q_0^E) \in \xi_i(P_{i,G}(s, q_0^E), x_{0,i})$$

Proof. By induction on the length of strings.

Basis: Take $s \in \Sigma^*, |s| = 0, s = \epsilon$

$$\xi_i(\epsilon, x_{0,i}) = x_{0,i} \qquad \text{(by definition of } \xi_i)$$

$$P_{i,G}(\epsilon, q_0^E) = \epsilon \Rightarrow \delta^E(\epsilon, q_0^E) \in x_{0,i}$$
$$\Rightarrow \delta^E(\epsilon, q_0^E) \in \xi_i(\epsilon, x_{0,i})$$

Inductive hypothesis:

Assume for all $s \in \Sigma^*$, $|s| \leq n, (n \geq 0)$

$$\delta^E(s, q_0^E) \in \xi_i(P_{i,G}(s, q_0^E), x_{0,i})$$

We need to prove for any $\sigma \in \Sigma$, $s \in \Sigma^*$,$|s| \leq n$,

$$\delta^E(s\sigma, q_0^E) \in \xi_i(P_{i,G}(s\sigma, q_0^E), x_{0,i})$$

In this thesis, we use $\delta_{i,o}$ to represent the set of transitions in G which are observable to Supervisor i ($i = 1, 2$) and $\delta_{i,uo}$ for the unobservable transition set. Any transition $(\delta^E(s, q_0^E), \sigma, \delta^E(s\sigma, q_0^E))$ in E corresponds to the transition $(\delta^G(s, q_0^G), \sigma, \delta^G(s\sigma, q_0^G))$ in G (by (3.1), (3.2) and (3.3)). Therefore, each transition in E is either in $\delta_{i,o}$ or in $\delta_{i,uo}$. We will use these two sets in the following discussion to characterize a transition in E. Also, each state q^E of E corresponds to a state q^G of G, so we also abuse notation by sometimes using a state in Q^G as the second argument to $P_{i,G}(\cdot, \cdot)$.

case 1: $(\delta^E(s, q_0^E), \sigma, \delta^E(s\sigma, q_0^E)) \in \delta_{i,o}$**.**

$$
\begin{aligned}
(\delta^E(s, q_0^E), \sigma, \delta^E(s\sigma, q_0^E))) \in \delta_{i,o} \Rightarrow\ & P_{i,G}(\sigma, \delta^E(s, q_0^E)) = \sigma \\
& \text{and } \delta^E(s, q_0^E) \in \xi_i(P_{i,G}(s, q_0^E), x_{0,i}) \\
& \text{(by inductive hypothesis)} \\
& \Rightarrow \delta^E(\sigma, \delta^E(s, q_0^E)) \in \xi_i(\sigma, \xi_i(P_{i,G}(s, q_0^E), x_{0,i})) \\
& \text{(by definition of } \xi_i) \\
& \Rightarrow \delta^E(s\sigma, q_0^E) \in \xi_i(P_{i,G}(s\sigma, q_0^E), x_{0,i}) \\
& \text{(by definition of } \xi_i \text{ and the fact that} \\
& (\delta^E(s, q_0^E), \sigma, \delta^E(s\sigma, q_0^E)) \in \delta_{i,o})
\end{aligned}
$$

case 2: $(\delta^E(s, q_0^E), \sigma, \delta^E(s\sigma, q_0^E)) \notin \delta_{i,o}$**.**

Since $P_{i,G}(s\sigma, q_0^E) = P_{i,G}(s, q_0^E)$, $\xi_i(P_{i,G}(s\sigma, q_0^E), x_{0,i}) = \xi_i(P_{i,G}(s, q_0^E), x_{0,i})$.

Suppose the last transition in s visible to S_i is an event γ from the state to which sequence t leads. In this case, s can be represented as $t\gamma t'$, where $t \in \Sigma^*$ and $t' \in \Sigma^*$. Then $s\sigma$ can be represented as $t\gamma t'\sigma$ and

$$P_{i,G}(\gamma, \delta^E(t, q_0^E)) = \gamma \text{ and } P_{i,G}(t'\sigma, \delta^E(t\gamma, q_0^E)) = \epsilon$$

$P_{i,G}(\gamma t'\sigma, \delta^E(t, q_0^E)) = \gamma$

and $\delta^E(t, q_0^E) \in \xi_i(P_{i,G}(t, q_0^E), x_{0,i})$ (by inductive hypothesis $|t| < n$)

$\Rightarrow \delta^E(\gamma t'\sigma, \delta^E(t, q_0^E)) \in \xi_i(\gamma, \xi_i(P_{i,G}(t, q_0^E), x_{0,i}))$ (by definition of ξ_i)

$\Rightarrow \delta^E(s\sigma, q_0^E) \in \xi_i(\gamma, \xi_i(P_{i,G}(t, q_0^E), x_{0,i}))$

$\Rightarrow \delta^E(s\sigma, q_0^E) \in \xi_i(P_{i,G}(t\gamma, q_0^E), x_{0,i})$

(since $P_{i,G}(t\gamma, q_0^E) = P_{i,G}(t, q_0^E)\gamma$, because

$(\delta^E(t, q_0^E), \gamma, \delta(t\gamma, q_0^E)) \in \delta_{i,o}$)

$\Rightarrow \delta^E(s\sigma, q_0^E) \in \xi_i(P_{i,G}(t\gamma t'\sigma, q_0^E), x_{0,i})$

(since $P_{i,G}(t\gamma t'\sigma, q_0^E) = P_{i,G}(t\gamma, q_0^E)$)

$\Rightarrow \delta^E(s\sigma, q_0^E) \in \xi_i(P_{i,G}(s\sigma, q_0^E), x_{0,i})$

□

The next claim asserts that for any two states in E combined together in a single state of the supervisor, there exist two distinct sequences in $L(E)$ leading to the states that have the same transition-based projection.

Claim 2. *For any state $q \in E$, for any string $s \in L(E)$, if $q \in \xi_i(P_{i,G}(s, q_0^E), x_{0,i})$, then there exists an $s' \in \Sigma^*$ such that $\delta^E(s', q_0^E) = q$ and $P_{i,G}(s, q_0^E) = P_{i,G}(s', q_0^E)$.*

Proof. We prove by induction on the length of string s.

Basis:

Take $s \in L(E)$, $|s| = 0$, $s = \epsilon$, and $q \in \xi_i(P_{i,G}(s, q_0^E), x_{0,i})$. Then

$$P_{i,G}(s, q_0^E) = \epsilon \Rightarrow \xi_i(P_{i,G}(s, q_0^E), x_{0,i}) = x_{0,i}$$

$$\Rightarrow \text{there exists } s' \in \Sigma^*,\ P_{i,G}(s', q_0^E) = \epsilon \text{ and } \delta^E(s', q_0^E) = q$$

(by the definition of $x_{0,i}$)

$$\Rightarrow P_{i,G}(s, q_0^E) = P_{i,G}(s', q_0^E)$$

Inductive hypothesis:

Suppose for all $s \in L(E)$, $|s| \leq n$, $n \geq 0$, for all $\hat{q} \in Q^E$, if $\hat{q} \in \xi_i(P_{i,G}(s, q_0^E), x_{0,i})$, then there exists $s' \in \Sigma^*$ such that $\delta^E(s', q_0^E) = \hat{q}$ and $P_{i,G}(s, q_0^E) = P_{i,G}(s', q_0^E)$.

Then we consider $q \in \xi_i(P_{i,G}(s\sigma, q_0^E), x_{0,i})$, where $\sigma \in \Sigma$.

Case 1: $(\delta^E(s, q_0^E), \sigma, \delta^E(s\sigma, q_0^E)) \notin \delta_{i,o}$

Since $P_{i,G}(s\sigma, q_0^E) = P_{i,G}(s, q_0^E)$, then $\xi_i(P_{i,G}(s, q_0^E), x_{0,i}) = \xi_i(P_{i,G}(s\sigma, q_0^E), x_{0,i})$. By the inductive hypothesis, for all q in $\xi_i(P_{i,G}(s, q_0^E), x_{0,i})$, which is also in $\xi_i(P_{i,G}(s\sigma, q_0^E), x_{0,i})$, there exists s' such that $\delta^E(s', q_0^E) = q$ and $P_{i,G}(s, q_0^E) = P_{i,G}(s', q_0^E)$, i.e., $P_{i,G}(s\sigma, q_0^E) = P_{i,G}(s', q_0^E)$

Case 2: $(\delta^E(s, q_0^E), \sigma, \delta^E(s\sigma, q_0^E)) \in \delta_{i,o}$

$q \in \xi_i(P_{i,G}(s\sigma, q_0^E), x_{0,i})$

$\Rightarrow q \in \xi_i(\sigma, \xi_i(P_{i,G}(s, q_0^E), x_{0,i}))$

$\Rightarrow q = \delta^E(t, q')$ for some $q' \in \xi_i(P_{i,G}(s, q_0^E), x_{0,i})$ and $P_{i,G}(t, q') = \sigma$

(by the definition of ξ_i)

Since $\delta^E(t, q')$ is defined and all states of E are reachable, there exists a sequence t' in $L(E)$ such that $\delta^E(t', q_0^E) = q'$ and $P_{i,G}(t', q_0^E) = P_{i,G}(s, q_0^E)$ (by the inductive hypothesis).

Let $s' = t't$, then $\delta^E(t't, q_0^E) = q$ and

$$\begin{aligned} P_{i,G}(t't, q_0^E) &= P_{i,G}(t', q_0^E) P_{i,G}(t, q') \\ &= P_{i,G}(s, q_0^E)\sigma \\ &= P_{i,G}(s\sigma, q_0^E) \end{aligned}$$

□

Now we continue the proof. We need to show if legal language $L(E)$ is controllable and transition-based coobservable with regard to $G, P_{1,G}$, and $P_{2,G}$, then the two supervisors S_1, S_2 we have constructed guarantee that $L(S_1 \wedge S_2/G) = L(E)$.

We proceed by induction on the length of strings.

Basis:

Take $s \in \Sigma^*, |s| = 0$, $s = \epsilon$

$L(E) \neq \emptyset \Rightarrow \delta^E(\epsilon, q_0^E)$ is defined

$\Rightarrow \delta^E(\epsilon, q_0^E) \in x_{0,i}$ (by definition of $x_{0,i}$ and the fact that $P_{i,G}(\epsilon, q_0^E) = \epsilon$)

$\Rightarrow x_{0,i}$ is nonempty

$\Rightarrow \epsilon \in L(S_1 \wedge S_2/G)$

$L(E) \neq \emptyset \Rightarrow \epsilon \in L(E)$

So, $\epsilon \in L(S_1 \wedge S_2/G) \Leftrightarrow \epsilon \in L(E)$.

Inductive hypothesis: Assume for all $s \in \Sigma^*$, $|s| < n (n \geq 0)$

$$s \in L(S_1 \wedge S_2/G) \Leftrightarrow s \in L(E)$$

We need to show that for all $\sigma \in \Sigma$

$$s\sigma \in L(S_1 \wedge S_2/G) \Leftrightarrow s\sigma \in L(E)$$

Suppose $\delta^E(s, q_0^E) = x_E$, $\xi_1(P_{1,G}(s, q_0^E), x_{0,1}) = x_a$, $\xi_2(P_{2,G}(s, q_0^E), x_{0,2}) = x_b$. By Claim 1, $x_E \in x_a$ and $x_E \in x_b$.

Case 1: $\sigma \in \Sigma_{1,c} \cap \Sigma_{2,c}$

$s\sigma \in L(E) \Rightarrow s\sigma \in L(G)$ and $s\sigma \in L(E)$ and $s \in L(E)$

$\Rightarrow s\sigma \in L(G)$ and $s\sigma \in L(E)$ and $s \in L(S_1 \wedge S_2/G)$ (by inductive hypothesis)

$\Rightarrow s\sigma \in L(G)$ and $s \in L(S_1 \wedge S_2/G)$ and $s\sigma \in L(E)$ and

$\psi_1(\sigma, x_a) = enable$ and $\psi_2(\sigma, x_b) = enable$

(Because $s\sigma \in L(E), \delta^E(\sigma, x_E)$ is defined. By the definition of ψ_i,

$\psi_1(\sigma, x_a) = enable$ and $\psi_2(\sigma, x_b) = enable$.)

$\Rightarrow s\sigma \in L(S_1 \wedge S_2/G)$

$s\sigma \in L(S_1 \wedge S_2/G) \Rightarrow s\sigma \in L(G)$ and $s \in L(S_1 \wedge S_2/G)$

and $\psi_1(\sigma, \xi_1(P_{1,G}(s, q_0^E), x_{0,1})) = enable$

and $\psi_2(\sigma, \xi_2(P_{2,G}(s, q_0^E), x_{0,2})) = enable$

$\Rightarrow s\sigma \in L(G)$ and $s \in L(E)$ (by inductive hypothesis)

and there exists $x' \in \xi_1(P_{1,G}(s, q_0^E), x_{0,1})$,

such that $\delta^E(\sigma, x')$ is defined;

and there exists $x'' \in \xi_2(P_{2,G}(s, q_0^E), x_{0,2})$,

such that $\delta^E(\sigma, x'')$ is defined

(by definition of ψ_i and $\sigma \in \Sigma_{1,c} \cap \Sigma_{2,c}$)

$\Rightarrow s\sigma \in L(G)$ and $s \in L(E)$

and there exists $x' \in Q^E$ such that $\delta^E(\sigma, x')$ is defined

and there exists $s' \in L(E)$ such that $\delta^E(s', q_0^E) = x'$

and $P_{1,G}(s', q_0^E) = P_{1,G}(s, q_0^E)$

and there exists $x'' \in Q^E$ such that $\delta^E(\sigma, x'')$ is defined

and there exists $s'' \in L(E)$ such that $\delta^E(s'', q_0^E) = x''$

and $P_{2,G}(s'', q_0^E) = P_{2,G}(s, q_0^E)$

(by Claim 2)

$\Rightarrow s\sigma \in L(G)$ and $s \in L(E)$ and there exists s', such that $s'\sigma \in L(E)$

and there exists s'', such that $s''\sigma \in L(E)$

and $P_{1,G}(s', q_0^E) = P_{1,G}(s, q_0^E)$ and $P_{2,G}(s'', q_0^E) = P_{2,G}(s, q_0^E)$

$\Rightarrow$ there exist s', $s'' \in \Sigma^*$, such that $P_{1,G}(s', q_0^E) = P_{1,G}(s, q_0^E)$

and $P_{2,G}(s'', q_0^E) = P_{2,G}(s, q_0^E)$ and

$s \in L(E)$ and $s\sigma \in L(G)$ and $s'\sigma \in L(E)$ and $s''\sigma \in L(E)$

$\Rightarrow s\sigma \in L(E)$ (by transition-based coobservability)

Case 2: $\sigma \in \Sigma_{1,c}, \sigma \notin \Sigma_{2,c}$

$s\sigma \in L(E) \Rightarrow s\sigma \in L(G)$ and $s\sigma \in L(E)$ and $s \in L(E)$

$\Rightarrow s\sigma \in L(G)$ and $s\sigma \in L(E)$

and $s \in L(S_1 \wedge S_2/G)$ (by inductive hypothesis)

$\Rightarrow s\sigma \in L(G)$ and $s \in L(S_1 \wedge S_2/G)$ and $s\sigma \in L(E)$ and $\psi_1(\sigma, x_a) = \mathit{enable}$

(Since $s\sigma \in L(E), \delta^E(\sigma, x_E)$ is defined. By the definition of ψ_1,

$\psi_1(\sigma, x_a) = \mathit{enable}$. Since $\sigma \notin \Sigma_{2,c}$, $\psi_2(\sigma, x_a) = \mathit{enable}$.)

$\Rightarrow s\sigma \in L(S_1 \wedge S_2/G)$

The other direction, $s\sigma \in L(S_1 \wedge S_2/G) \Rightarrow s\sigma \in L(E)$, follows the same reasoning as in Case 1.

Case 3: $\sigma \in \Sigma_{2,c}, \sigma \notin \Sigma_{1,c}$

Analogous to Case 2.

Case 4: $\sigma \notin \Sigma_{1,c}, \sigma \notin \Sigma_{2,c}$

The proof that $s\sigma \in L(E) \Rightarrow s\sigma \in L(S_1 \wedge S_2/G)$ follows the same reasoning as in Case 2.

$$
\begin{aligned}
s\sigma \in L(S_1 \wedge S_2/G) &\Rightarrow s\sigma \in L(G) \text{ and } s \in L(S_1 \wedge S_2/G) \\
&\Rightarrow s\sigma \in L(G) \text{ and } s \in L(E) \quad \text{(by inductive hypothesis)} \\
&\Rightarrow s\sigma \in L(E) \quad \text{(by property of controllability: } L(\bar{E})\Sigma_{u,c} \cap L(G) \subseteq L(\bar{E}))
\end{aligned}
$$

□

3.3 A Simple Example

Consider the plant G shown in Figure 3.4 and the legal behavior E shown in Figure 3.5. Suppose $\Sigma_{1,o} = \Sigma_{1,c} = \{\alpha, \gamma\}$ and $\Sigma_{2,o} = \Sigma_{2,c} = \{\beta, \gamma\}$.

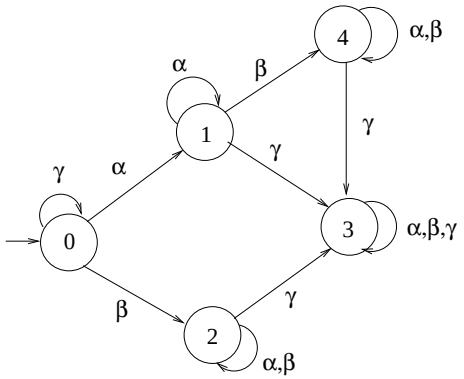

Figure 3.4: Plant G

The legal language $L(E)$ is not coobservable with respect to G because $P_1(\alpha\beta) = P_1(\alpha)$ but $\alpha\beta\gamma \notin L(\bar{E})$, $\alpha\gamma \in L(\bar{E})$. Supervisor 1 cannot determine whether α happens or $\alpha\beta$ happens, but it should disable γ after $\alpha\beta$ happens and enable γ after

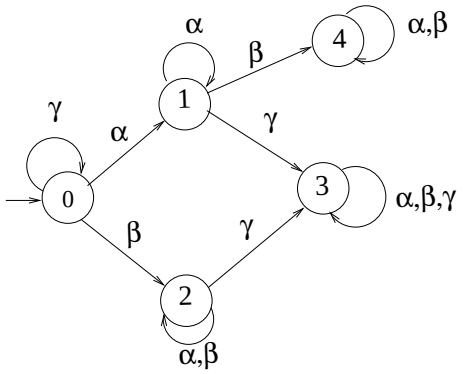

Figure 3.5: Legal behavior E

α happens. The same problem exists for Supervisor 2: it needs to disable γ after $\alpha\beta$ happens and enable γ after β, but $\alpha\beta$ and β look the same to Supervisor 2. Since we model the control actions of two supervisors operating in parallel, at least one supervisor needs to disable γ after $\alpha\beta$ occurs.

Now we consider if $(1, \beta, 4)$ is communicated from Supervisor 2 to Supervisor 1, i.e., we can say that Supervisor 1 "observes" β when the system is in state 1. According to our supervisor construction in the proof of Theorem 1, we construct two supervisors as follows.

For Supervisor 1, we replace all event labels in unobservable transitions with ϵ. This is displayed in Figure 3.6.

Then we convert the NFA of Figure 3.6 to its equivalent DFA (Figure 3.7), and add self-loops to those events not defined so that the automaton is fully defined.

Now we do the analogous operations to get Supervisor 2.

Supervisor 2 is displayed in Figure 3.9.

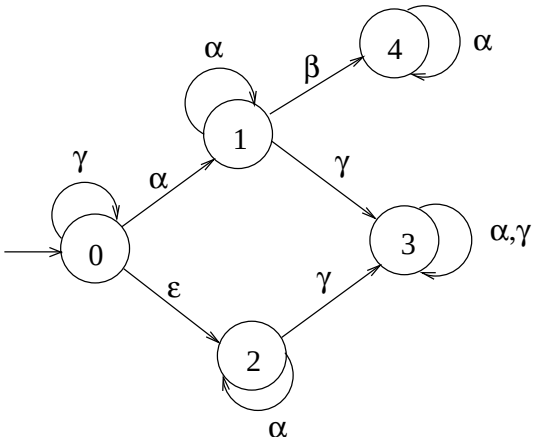

Figure 3.6: NFA of Supervisor 1

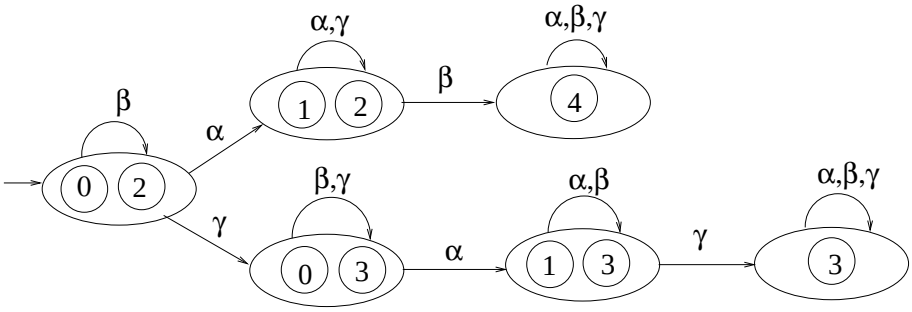

Figure 3.7: Supervisor 1

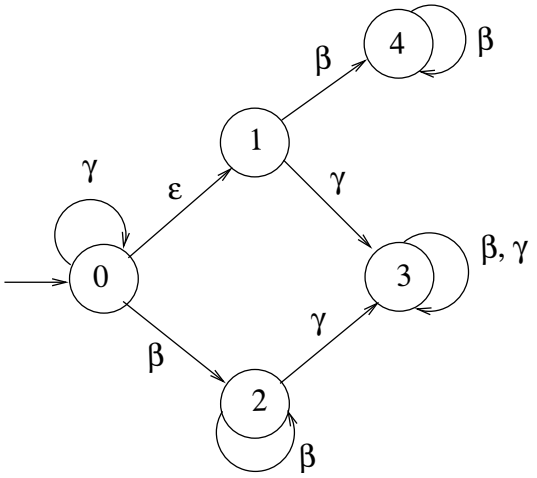

Figure 3.8: NFA of Supervisor 2

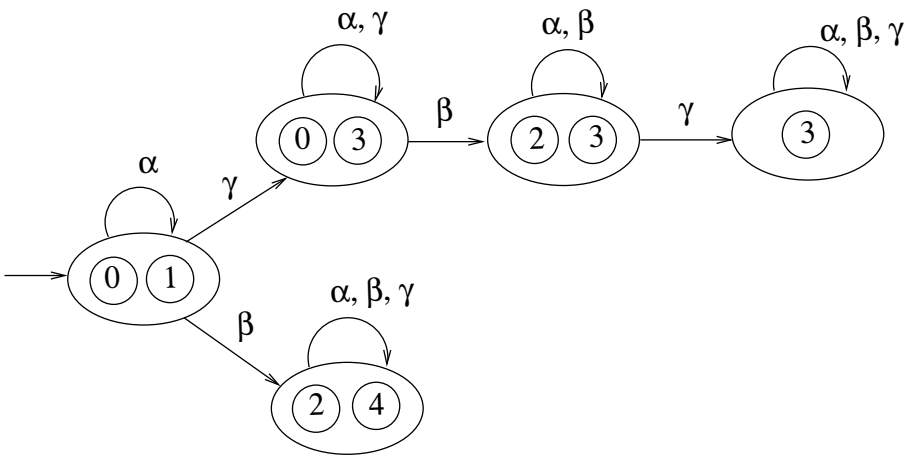

Figure 3.9: Supervisor 2

For Supervisor 1, the transition-based projection of $\alpha\beta$ leads to state {4}. According to the definition of ψ_1, $\psi_1(\gamma, \{4\}) = \mathit{disable}$. For Supervisor 2, the transition-based projection of $\alpha\beta$ leads to state {2,4} and according to the definition of ψ_2, $\psi_2(\gamma, \{2, 4\}) = \mathit{enable}$. So, the overall joint control action is to disable γ after $\alpha\beta$ occurs. In other words, with the communication from Supervisor 2 to Supervisor 1, the system is transition-based coobservable.

Chapter 4

Checking Transition-Based Coobservability

As we know from the preceding section, transition-based coobservability is one of the essential conditions for decentralized supervisory control with communications. In [11], an automaton, called an M-machine, is constructed to check coobservability. The M-machine traces all the possible sequences ambiguous to supervisors, then checks if those sequences violate the property of coobservability of $L(E)$ with regard to G. In this section, we will modify the structure of the M-machine so that it will check transition-based coobservability.

4.1 The Modified M-Machine

Given $G = (\Sigma, Q^G, \delta^G, q_0^G)$ and $E = (\Sigma, Q^E, \delta^E, q_0^E)$, where E is a subautomaton of G, we construct an automaton $\tilde{M}(G, E)$ which is similar to M in [11]. There is only one marked state which is denoted by d, and called the *dump state*.

The automaton $\tilde{M}$ is defined by

$$\tilde{M} := (\Sigma \times \{0, 1, ..., 7\}, Q^{\tilde{M}}, \delta^{\tilde{M}}, q_0^{\tilde{M}}, Q_m^{\tilde{M}})$$

where

$$\begin{aligned} Q^{\tilde{M}} &:= Q^E \times Q^E \times Q^E \times Q^G \cup \{d\} \\ q_0^{\tilde{M}} &:= (q_0^E, q_0^E, q_0^E, q_0^G) \\ Q_m^{\tilde{M}} &:= \{d\}. \end{aligned}$$

Before defining the transition function $\delta^{\tilde{M}}$, we first define a *transition type*. In particular, each transition will be labelled by an event label from Σ and a number, called a transition type, from 0 to 7. The numbers will be used to track sequences ambiguous to the supervisors. The following list defines transition types.

$$(q_1,q_2,q_3,q_4) \xrightarrow{\sigma,1} (\delta^E(\sigma,q_1),q_2,q_3,q_4) \quad (\text{if } \delta^E(\sigma,q_1) \text{ is defined})$$

$$(q_1,q_2,q_3,q_4) \xrightarrow{\sigma,2} (q_1,\delta^E(\sigma,q_2),q_3,q_4) \quad (\text{if } \delta^E(\sigma,q_2) \text{ is defined})$$

$$(q_1,q_2,q_3,q_4) \xrightarrow{\sigma,3} (q_1,q_2,\delta^E(\sigma,q_3),\delta^G(\sigma,q_4)) \quad (\text{if } \delta^E(\sigma,q_3) \text{ is defined and } \delta^G(\sigma,q_4) \text{ is defined})$$

$$(q_1,q_2,q_3,q_4) \xrightarrow{\sigma,4} (\delta^E(\sigma,q_1),\delta^E(\sigma,q_2),\delta^E(\sigma,q_3),\delta^G(\sigma,q_4)) \quad (\text{if } \delta^E(\sigma,q_1) \text{ is defined and } \delta^E(\sigma,q_2) \text{ is defined and } \delta^E(\sigma,q_3) \text{ is defined and } \delta^G(\sigma,q_4) \text{ is defined})$$

$$(q_1,q_2,q_3,q_4) \xrightarrow{\sigma,5} (q_1,\delta^E(\sigma,q_2),\delta^E(\sigma,q_3),\delta^G(\sigma,q_4)) \quad (\text{if } \delta^E(\sigma,q_2) \text{ is defined and } \delta^E(\sigma,q_3) \text{ is defined and } \delta^G(\sigma,q_4) \text{ is defined})$$

$$(q_1,q_2,q_3,q_4) \xrightarrow{\sigma,6} (\delta^E(\sigma,q_1),q_2,\delta^E(\sigma,q_3),\delta^G(\sigma,q_4)) \quad (\text{if } \delta^E(\sigma,q_1) \text{ is defined and } \delta^E(\sigma,q_3) \text{ is defined and } \delta^G(\sigma,q_4) \text{ is defined})$$

$$(q_1,q_2,q_3,q_4) \xrightarrow{\sigma,7} (\delta^E(\sigma,q_1),\delta^E(\sigma,q_2),q_3,q_4) \quad (\text{if } \delta^E(\sigma,q_1) \text{ is defined and } \delta^E(\sigma,q_2) \text{ is defined})$$

$$(q_1,q_2,q_3,q_4) \xrightarrow{\sigma,0} d \quad \text{if} \begin{cases} \sigma \in \Sigma_c \\ \delta^E(\sigma,q_1) \text{ is defined if } \sigma \in \Sigma_{1,c} \\ \delta^E(\sigma,q_2) \text{ is defined if } \sigma \in \Sigma_{2,c} \\ \delta^E(\sigma,q_3) \text{ is not defined} \\ \delta^G(\sigma,q_4) \text{ is defined} \end{cases} \tag{4.1}$$

For example, a transition from state (q_1,q_2,q_3,q_4) to state $(\delta^E(\sigma,q_1),q_2,q_3,q_4)$ on event σ will be a type 1 transition.

The aim of the M-machine is to trace all the possible sequences that could happen in the system and look the same to the supervisors, then check if any of these sequences violate transition-based coobservability. Therefore it keeps three copies of the legal automaton E and one copy of the plant G. For any three sequences s, s', and s'' which makes $P_{1,G}(s, q_0^E) = P_{1,G}(s', q_0^E)$ and $P_{2,G}(s, q_0^E) = P_{2,G}(s'', q_0^E)$, they would end up in a state of $\tilde{M}$ where the first component of the state records where s' leads to in E and the second component records where s'' leads to in E and the third component records where s leads to in E. For a state (q_1, q_2, q_3, q_4), if an event σ doesn't lead to the dump state, we only add transitions from the state with event label σ when σ is defined in q_1, q_2 and q_3 in E and q_4 in G.

Let us use the same example as in Section 3.3, namely, the example shown in Figures 3.4 and 3.5 (as reproduced in Figures 4.1 and 4.2), to explain how to construct the automaton $\tilde{M}$, then we show how, in general, $\tilde{M}$ can be used to check transition-based coobservability. In this example, $\Sigma_{1,c} = \Sigma_{1,o} = \{\alpha, \gamma\}$ and $\Sigma_{2,c} = \Sigma_{2,o} = \{\beta, \gamma\}$.

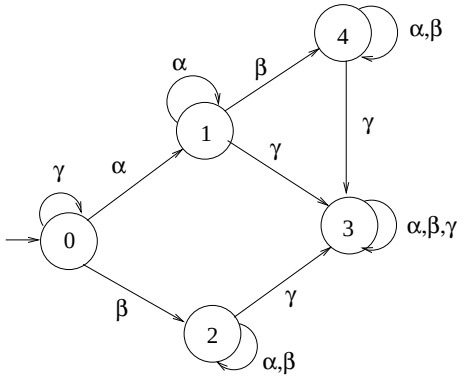

Figure 4.1: Plant G

The initial state of $\tilde{M}$ is $(q_0^E, q_0^E, q_0^E, q_0^G)$, i.e., state $(0, 0, 0, 0)$ in this example. Since α from state 0 in E is defined and observable only by Supervisor 1 , s', s'', s could be

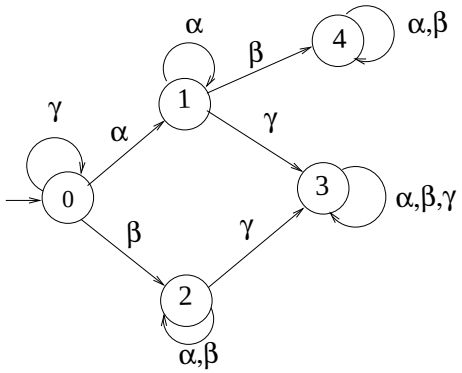

Figure 4.2: Legal behavior E

$$s' = \epsilon,\ s'' = \alpha,\ s = \epsilon \tag{4.2}$$

$$s' = \alpha,\ s'' = \alpha,\ s = \alpha \tag{4.3}$$

$$s' = \alpha,\ s'' = \epsilon,\ s = \alpha \tag{4.4}$$

For (4.2), α only happens in the second copy of E and there is no event happening in the first and the third copies. However s and s' look the same to Supervisor 1, $P_{1,G}(s', 0) = \epsilon$ and $P_{1,G}(s, 0) = \epsilon$, therefore $P_{1,G}(s', 0) = P_{1,G}(s, 0) = \epsilon$; and s and s'' look the same to the Supervisor 2, $P_{2,G}(s'', 0) = \epsilon$ and $P_{2,G}(s, 0) = \epsilon$, therefore $P_{2,G}(s'', 0) = P_{2,G}(s, 0) = \epsilon$. By the same reasoning, the cases for (4.3) and (4.4) guarantee that $P_{1,G}(s, q_0^E) = P_{1,G}(s', q_0^E)$ and $P_{2,G}(s, q_0^E) = P_{2,G}(s'', q_0^E)$. The three cases are of transition type 2, 4 and 6, respectively, so they lead to $(0, 1, 0, 0)$, $(1, 1, 1, 1)$ and $(1, 0, 1, 1)$, respectively (Figure 4.3).

From state $(0, 1, 0, 0)$, since β from state 0 of E is only observable by Supervisor 2 and β from state 1 of E is observable by both Supervisor 1 and Supervisor 2 after communication, the transitions from $(0, 1, 0, 0)$ with event label β include the

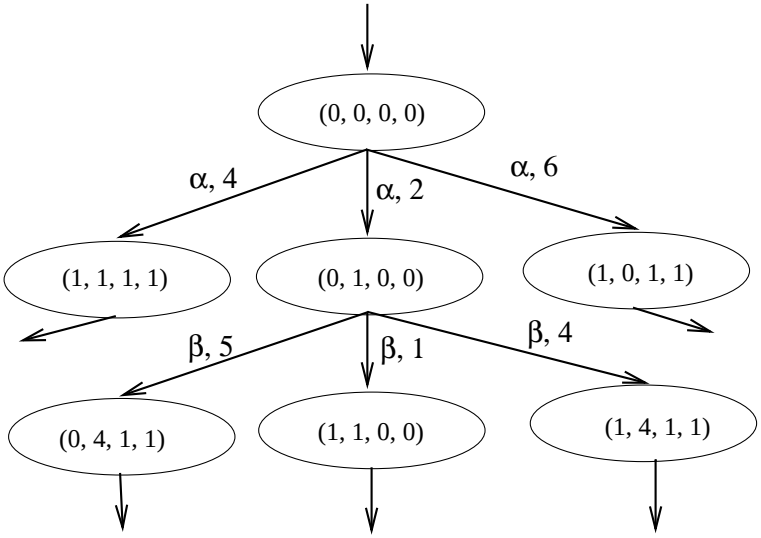

Figure 4.3: A Portion of the Modified M-machine $\tilde{M}$ for Example 4.1

following three cases, where the three events in each line show whether β really occurs or not in each copy of E:

$$\beta,\ \epsilon,\ \epsilon \tag{4.5}$$

$$\beta,\ \beta,\ \beta \tag{4.6}$$

$$\epsilon,\ \beta,\ \beta \tag{4.7}$$

For (4.6), namely when β occurs in all three copies, since $\delta^E(s', q_0^E) = 0$ and $\delta^E(s'', q_0^E) = 1$ and $\delta^E(s, q_0^E) = 0$, then $P_{1,G}(\beta, \delta^E(s', q_0^E)) = P_{1,G}(\beta, \delta^E(s, q_0^E)) = \epsilon$ and $P_{2,G}(\beta, \delta^E(s'', q_0^E)) = P_{2,G}(\beta, \delta^E(s, q_0^E)) = \beta$. The three cases are of transition type 1, 4 and 5, respectively, and lead to three different new states in $\tilde{M}$, as seen in Figure 4.3. The cases for (4.5) and (4.7) follow the same arguments. Following similar operations, we could construct the entire automaton $\tilde{M}$.

Scenario	Shorthand Symbol
$(\delta^E(s', q_0), \sigma, \delta^E(\sigma, \delta^E(s', q_0))) \in \delta_{1,o} \cap \delta_{2,o}$	1A
$(\delta^E(s', q_0), \sigma, \delta^E(\sigma, \delta^E(s', q_0))) \in \delta_{1,o} \backslash \delta_{2,o}$	1B
$(\delta^E(s', q_0), \sigma, \delta^E(\sigma, \delta^E(s', q_0))) \in \delta_{2,o} \backslash \delta_{1,o}$	1C
$(\delta^E(s', q_0), \sigma, \delta^E(\sigma, \delta^E(s', q_0))) \notin \delta_{2,o} \cup \delta_{1,o}$	1D
$(\delta^E(s'', q_0), \sigma, \delta^E(\sigma, \delta^E(s'', q_0))) \in \delta_{1,o} \cap \delta_{2,o}$	2A
$(\delta^E(s'', q_0), \sigma, \delta^E(\sigma, \delta^E(s'', q_0))) \in \delta_{1,o} \backslash \delta_{2,o}$	2B
$(\delta^E(s'', q_0), \sigma, \delta^E(\sigma, \delta^E(s'', q_0))) \in \delta_{2,o} \backslash \delta_{1,o}$	2C
$(\delta^E(s'', q_0), \sigma, \delta^E(\sigma, \delta^E(s'', q_0)))) \notin \delta_{2,o} \cup \delta_{1,o}$	2D
$(\delta^E(s, q_0), \sigma, \delta^E(\sigma, \delta^E(s, q_0))) \in \delta_{1,o} \cap \delta_{2,o}$	3A
$(\delta^E(s, q_0), \sigma, \delta^E(\sigma, \delta^E(s, q_0))) \in \delta_{1,o} \backslash \delta_{2,o}$	3B
$(\delta^E(s, q_0), \sigma, \delta^E(\sigma, \delta^E(s, q_0))) \in \delta_{2,o} \backslash \delta_{1,o}$	3C
$(\delta^E(s, q_0), \sigma, \delta^E(\sigma, \delta^E(s, q_0))) \notin \delta_{2,o} \cup \delta_{1,o}$	3D

Table 4.1: Shorthand Symbol for Different Scenarios

Generally, for each of the three sequences s', s'', and s which make $P_{1,G}(s, q_0^E) = P_{1,G}(s', q_0^E)$ and $P_{2,G}(s, q_0^E) = P_{2,G}(s'', q_0^E)$, if an event σ happens after the sequence, it could be observable by Supervisor 1 or Supervisor 2 or both supervisors or neither of the supervisors. Each scenario is identified by a symbol, as summarized in Table 4.1. We use $\delta_{1,o}$ to represent the set of observable transitions to Supervisor 1 and $\delta_{2,o}$ for the observable transitions set of Supervisor 2.

As we can see from Table 4.1 , since each of $s\sigma$, $s'\sigma$ and $s''\sigma$ has 4 possibilities, the total number of possible combinations for a triple ($s\sigma$, $s'\sigma$, $s''\sigma$) is 64. Now in Table 4.2 we define $\delta^{\tilde{M}}$ according to the 64 cases, where the numbers on the right side of each case represent the transition types.

To get a feel for Table 4.2, let us return to Example 4.1 of Figure 4.3. From state $(0, 0, 0, 0)$, since $(0, \alpha, 1) \in \delta_{1,o} \backslash \delta_{2,o}$, the scenarios are represented by $1B$, $2B$, and $3B$. From Table 4.2, category $1B, 2B, 3B$ corresponds to transition types 2, 4 and 6. From state (0,1,0,0), since $(0, \beta, 2) \in \delta_{2,o} \backslash \delta_{1,o}$ and $(1, \beta, 4) \in \delta_{1,o} \cup \delta_{2,o}$, the scenarios

1A,2A,3A	4	1A,2A,3B	6	1A,2A,3C	5	1A,2A,3D	3
1A,2B,3A	2	1A,2B,3B	2,4,6	1A,2B,3C	2	1A,2B,3D	2,3,5
1A,2C,3A	4	1A,2C,3B	6	1A,2C,3C	5	1A,2C,3D	3
1A,2D,3A	2	1A,2D,3B	2,4,6	1A,2D,3C	2	1A,2D,3D	2,3,5
1B,2A,3A	4	1B,2A,3B	6	1B,2A,3C	5	1B,2A,3D	3
1B,2B,3A	2	1B,2B,3B	2,4,6	1B,2B,3C	2	1B,2B,3D	2,3,5
1B,2C,3A	4	1B,2C,3B	6	1B,2C,3C	5	1B,2C,3D	3
1B,2D,3A	2	1B,2D,3B	2,4,6	1B,2D,3C	2	1B,2D,3D	2,3,5
1C,2A,3A	1	1C,2A,3B	1	1C,2A,3C	1,4,5	1C,2A,3D	1,3,6
1C,2B,3A	1,2,7	1C,2B,3B	1,2,7	1C,2B,3C	1,2,7	1C,2B,3D	1,2,3,4,5,6,7
1C,2C,3A	1	1C,2C,3B	1	1C,2C,3C	1,4,5	1C,2C,3D	1,3,6
1C,2D,3A	1,2,7	1C,2D,3B	1,2,7	1C,2D,3C	1,2,7	1C,2D,3D	1,2,3,4,5,6,7
1D,2A,3A	1	1D,2A,3B	1	1D,2A,3C	1,4,5	1D,2A,3D	1,3,6
1D,2B,3A	1,2,7	1D,2B,3B	1,2,7	1D,2B,3C	1,2,7	1D,2B,3D	1,2,3,4,5,6,7
1D,2C,3A	1	1D,2C,3B	1	1D,2C,3C	1,4,5	1D,2C,3D	1,3,6
1D,2D,3A	1,2,7	1D,2D,3B	1,2,7	1D,2D,3C	1,2,7	1D,2D,3D	1,2,3,4,5,6,7

Table 4.2: Transition Table of $\tilde{M}$

are $1C, 2A$ and $3C$ which map to transition types 1, 4, and 5.

In Table 4.2, it is easy to notice that some parts of the table are the same as other parts. If we separate the table into sixteen blocks and each block has four lines as shown in the table, then the transition types are the same for the same location in blocks (1,1) and (1,2); in blocks (1,2) and (2,2); in blocks (1,3) and (2,3); in blocks (1,4) and (2,4); in blocks (3,1) and (4,1); in blocks (3,2) and (4,2); in blocks (3,3) and (4,3); and in blocks (3,4) and (4,4). To see why this is so, consider, for example, blocks (1,1) and (1,2): The only difference between the scenarios is that in block (1,1) the first scenario is $1A$ and in block (1,2) it is $1B$. Since $1A$ denotes σ after s' is observable by both Supervisor 1 and Supervisor 2, and $1B$ denotes σ after s' is observable only by Supervisor 1, and for s' it only matters what Supervisor 1 sees, therefore they will have the same transition types. Analogously, since for s'' it only

Scenarios	Transition Types
1A(1B),2A(2C),3A	4
1A(1B),2A(2C),3B	6
1A(1B),2A(2C),3C	5
1A(1B),2A(2C),3D	3
1A(1B),2B(2D),3A	2
1A(1B),2B(2D),3B	2,4,6
1A(1B),2B(2D),3C	2
1A(1B),2B(2D),3D	2,3,5
1C(1D),2A(2C),3A	1
1C(1D),2A(2C),3B	1
1C(1D),2A(2C),3C	1,4,5
1C(1D),2A(2C),3D	1,3,6
1C(1D),2B(2D),3A	1,2,7
1C(1D),2B(2D),3B	1,2,7
1C(1D),2B(2D),3C	1,2,7
1C(1D),2B(2D),3D	1,2,3,4,5,6,7

Table 4.3: Simplified Transition Table of $\tilde{M}$

matters what Supervisor 2 sees, $2A$ and $2C$ will have the same transition types and $2B$ and $2D$ will have the same transition types if the other two scenarios are the same, which explains why the first line and the third line in each block have same transition types, and the second line and the fourth line are same. Therefore, we can simplify Table 4.2 to Table 4.3.

We also note that transition type 0 is not in Table 4.2; it leads to the dump state which is also the only marked state in $\tilde{M}$.

4.2 Using the Modified M-Machine to Check Transition-Based Coobservability

Now we show how the automaton $\tilde{M}$ we constructed in Section 1 can be used to check transition-based coobservability. The approach we take is based heavily on that of [11, 12].

Proposition 1. *Given plant G and legal behavior $L(E)$, $L(E)$ is not transition-based coobservable w.r.t. G and $P_{i,G}$ iff there is a path from the initial state to the dump state d in $\tilde{M}$.*

Proof. **If:**

If there exists a path to the dump state in the automaton $\tilde{M}$, then we need to prove that E is not transition-based coobservable w.r.t. G and $P_{i,G}$.

Suppose the sequence leading to the dump state is $s^*\sigma$, where $s^* \in \Sigma^*$, and $\sigma \in \Sigma$.

We extract s, s' and s'' from s^*, then prove these strings s, s' and s'' violate coobservability. The rules to produce s, s' and s'' are as follows. First we define the following filters from $\Sigma \times \{1, 2, ...7\}$ to $\Sigma \cup \{\epsilon\}$:

$$F_1(\sigma, i) := \begin{cases} \sigma & \text{if } i = 1, 4, 6, 7 \\ \epsilon & \text{otherwise} \end{cases}$$

$$F_2(\sigma, i) := \begin{cases} \sigma & \text{if } i = 2, 4, 5, 7 \\ \epsilon & \text{otherwise} \end{cases}$$

$$F_3(\sigma, i) := \begin{cases} \sigma & \text{if } i = 3, 4, 5, 6 \\ \epsilon & \text{otherwise} \end{cases}$$

Where the transition type i is understood, we use $F_j(\sigma)$ instead of $F_j(\sigma, i)\,(j = 1, 2, 3)$.

The mappings F_1, F_2 and F_3 can be extended to sequences of events in the natural way, namely,

$$F_i(\epsilon) := \epsilon \qquad i = 1, 2, 3$$

and for $s \in \Sigma^*$, $\sigma \in \Sigma$

$$F_i(s\sigma) := F_i(s)F_i(\sigma) \qquad i = 1, 2, 3$$

Now we use the $F_i(i = 1, 2, 3)$ to extract our counterexample triple:

$$s := F_3(s^*)$$
$$s' := F_1(s^*)$$
$$s'' := F_2(s^*)$$

The following results prove that the s, s' and s'' we get from the above operations is a counterexample to transition-based coobservability. Note that in the claims that follow, a hypothesis that some string is in $L(\tilde{M})\backslash L_m(\tilde{M})$ means that the string does not lead to the state $\{d\}$ in $\tilde{M}$. Consequently, we can assume that the last transition

in the string falls into one of the 64 categories given in Table 4.2.

Claim 3. *For $s^* \in L(\tilde{M})\backslash L_m(\tilde{M})$, $F_3(s^*) \in L(E)$ and if s^* leads to state (q_1, q_2, q_3, q_4) in $\tilde{M}$ then $F_3(s^*)$ leads to q_3 in E and $F_3(s^*)$ leads to q_4 in G.*

Proof. By induction on the length of s^*.

Basis:

$$|s^*| = 0 \Rightarrow s^* = \epsilon$$
$$\Rightarrow s^* \text{ leads to } (q_0^E, q_0^E, q_0^E, q_0^G) \text{ in } \tilde{M}$$

$F_3(s^*) = F_3(\epsilon) = \epsilon$ which leads to q_0^E in E and q_0^G in G.

Inductive hypothesis:

Suppose for all $|s^*| \leq n\,(n \geq 0)$, $F_3(s^*) \in L(E)$ and if s^* leads to (q_1, q_2, q_3, q_4) in $\tilde{M}$, then $F_3(s^*)$ leads to q_3 in E and $F_3(s^*)$ leads to q_4 in G.

First we consider where $F_3(s^*)$ leads to in E.

Take $s^* = \bar{s}\bar{\sigma}$ where $|\bar{s}| = n$. If $\bar{s}$ leads to (q_1, q_2, q_3, q_4) in $\tilde{M}$, then $F_3(s^*)$ leads to q_3 in E by the inductive hypothesis.

Case 1: $F_3(\bar{\sigma}) = \epsilon$

$$F_3(\bar{\sigma}) = \epsilon \Rightarrow F_3(\bar{s}\bar{\sigma}) = F_3(\bar{s})$$
$$\Rightarrow F_3(\bar{s}\bar{\sigma}) \in L(E) \quad \text{(by inductive hypothesis)}$$

Because $F_3(\bar{\sigma}) = \epsilon$, $\bar{\sigma}$ results from a transition whose type is 1, 2, or 7 where the third argument in the 4-tuple state (q_1, q_2, q_3, q_4) does not change. The sequence $F_3(\bar{s}\bar{\sigma})$ leads to the same state q_3 in E as $F_3(\bar{s})$ leads to.

Case 2: $F_3(\bar{\sigma}) = \bar{\sigma}$

Since $F_3(\bar{\sigma}) = \bar{\sigma}$ results from a transition type 3, 4, 5 or 6 and in each of those transition types, $\delta^E(\bar{\sigma}, q_3)$ is defined and since $F_3(\bar{s})$ ends at q_3 and $F_3(\bar{s}) \in L(E)$ (by the inductive hypothesis), we can deduce that $F_3(\bar{s})\bar{\sigma} \in L(E)$, i.e., $F_3(\bar{s}\bar{\sigma}) \in L(E)$.

By the inductive hypothesis $F_3(\bar{s})$ leads to q_3 in E. Since $\delta^E(\bar{\sigma}, q_3)$ is defined, $F_3(\bar{s})\bar{\sigma}$ leads to $\delta^E(\bar{\sigma}, q_3)$, which is the third argument of the 4-tuple state to which $\bar{s}\bar{\sigma}$ leads in $\tilde{M}$.

Now we check where $F_3(s^*)$ leads to in G. Since the fourth arguments in the 4-tuple states of $\tilde{M}$ change only when the third arguments do, $F_3(s^*)$ leads to q_4 in G. □

Claim 4. *For $s^* \in L(\tilde{M})\backslash L_m(\tilde{M})$, $F_1(s^*) \in L(E)$ and if s^* leads to state (q_1, q_2, q_3, q_4) in $\tilde{M}$ then $F_1(s^*)$ leads to q_1 in E.*

Proof. Analogous to **Claim 1**. □

Claim 5. *For $s^* \in L(\tilde{M})\backslash L_m(\tilde{M})$, $F_2(s^*) \in L(E)$ and if s^* leads to state (q_1, q_2, q_3, q_4) in $\tilde{M}$ then $F_2(s^*)$ leads to q_2 in E.*

Proof. Analogous to **Claim 1**. □

Claim 6. *For $s^* \in L(\tilde{M})\backslash L_m(\tilde{M})$, $P_{1,G}(F_3(s^*), q_0^E) = P_{1,G}(F_1(s^*), q_0^E)$.*

Proof. By induction on the length of s^*.

Basis:

$$\begin{aligned} |s^*| = 0 &\Rightarrow s^* = \epsilon \\ &\Rightarrow F_1(s^*) = F_3(s^*) = \epsilon \\ &\Rightarrow P_{1,G}(F_3(s^*), q_0^E) = P_{1,G}(F_1(s^*), q_0^E) = \epsilon \end{aligned}$$

Inductive hypothesis:

Suppose for all $|s^*| \leq n$, $n \geq 0$, $P_{1,G}(F_3(s^*), q_0^E) = P_{1,G}(F_1(s^*), q_0^E)$.

Take $s^* = \bar{s}\bar{\sigma}$ where $|\bar{s}| = n$. Then,

$$\begin{aligned} P_{1,G}(F_1(\bar{s}\bar{\sigma}), q_0^E) &= P_{1,G}(F_1(\bar{s})F_1(\bar{\sigma}), q_0^E) \\ &= P_{1,G}(F_1(\bar{s}), q_0^E)P_{1,G}(F_1(\bar{\sigma}), \delta^E(F_1(\bar{s}), q_0^E)) \end{aligned} \tag{4.8}$$

$$P_{1,G}(F_3(\bar{s}\bar{\sigma}), q_0^E) = P_{1,G}(F_3(\bar{s}), q_0^E)P_{1,G}(F_3(\bar{\sigma}), \delta^E(F_3(\bar{s}), q_0^E)) \tag{4.9}$$

By the inductive hypothesis,

$$P_{1,G}(F_1(\bar{s}), q_0^E) = P_{1,G}(F_3(\bar{s}), q_0^E) \tag{4.10}$$

For simplicity, we define $q := \delta^E(F_3(\bar{s}), q_0^E)$ and $q' := \delta^E(F_1(\bar{s}), q_0^E)$ (which, by Claim 1 and Claim 2, are well-defined).

So, to prove $P_{1,G}(F_1(\bar{s}\bar{\sigma}), q_0^E) = P_{1,G}(F_3(\bar{s}\bar{\sigma}), q_0^E)$, by (4.8), (4.9) and (4.10), we need to show

$$P_{1,G}(F_1(\bar{\sigma}), q') = P_{1,G}(F_3(\bar{\sigma}), q)$$

Because $s^* = \bar{s}\bar{\sigma}$ is a sequence in $\tilde{M}$ not leading to $\{d\}$, $\delta^E(\bar{\sigma}, q')$ is defined and the transition leading to it could have 4 possibilities, namely, $(q', \bar{\sigma}, \delta^E(\bar{\sigma}, q')) \in \delta_{1,o} \cap \delta_{2,o}$, $(q', \bar{\sigma}, \delta^E(\bar{\sigma}, q')) \in \delta_{1,o} \backslash \delta_{2,o}$, $(q', \bar{\sigma}, \delta^E(\bar{\sigma}, q')) \in \delta_{2,o} \backslash \delta_{1,o}$, or $(q', \bar{\sigma}, \delta^E(\bar{\sigma}, q')) \notin \delta_{1,o} \cup \delta_{2,o}$, and so could $(q, \bar{\sigma}, \delta^E(\bar{\sigma}, q))$, so we need to check 16 cases.

Case 1: $(q', \bar{\sigma}, \delta^E(\bar{\sigma}, q')) \in \delta_{1,o} \cap \delta_{2,o} \wedge (q, \bar{\sigma}, \delta^E(\bar{\sigma}, q)) \in \delta_{1,o} \cap \delta_{2,o}$

According to the structure of $\tilde{M}$ as given in Table 4.1, we need to check categories with scenarios $1A$ and $3A$. From the definition of the transition function of $\tilde{M}$ (Table 4.3), a transition type 2 or 4 would occur.

For transition type 4, $F_1(\bar{\sigma}) = F_3(\bar{\sigma}) = \bar{\sigma}$,

$$\begin{aligned} P_{1,G}(F_1(\bar{s}\bar{\sigma}), q_0^E) &= P_{1,G}(F_1(\bar{s}), q_0^E) P_{1,G}(\bar{\sigma}, q') \quad \text{(by (4.8))} \\ &= P_{1,G}(F_1(\bar{s}), q_0^E)\bar{\sigma} \quad \text{(since } (q', \bar{\sigma}, \delta^E(\bar{\sigma}, q')) \in \delta_{1,o} \cap \delta_{2,o}) \end{aligned}$$

$$\begin{aligned} P_{1,G}(F_3(\bar{s}\bar{\sigma}), q_0^E) &= P_{1,G}(F_3(\bar{s}), q_0^E) P_{1,G}(\bar{\sigma}, q) \quad \text{(by (4.9))} \\ &= P_{1,G}(F_3(\bar{s}), q_0^E)\bar{\sigma} \quad \text{(since } (q, \bar{\sigma}, \delta^E(\bar{\sigma}, q)) \in \delta_{1,o} \cap \delta_{2,o}) \end{aligned}$$

Therefore, $P_{1,G}(F_1(\bar{s}\bar{\sigma}), q_0^E) = P_{1,G}(F_3(\bar{s}\bar{\sigma}), q_0^E)$ (by (4.10)).

For transition type 2, $F_1(\bar{\sigma}) = F_3(\bar{\sigma}) = \epsilon$.

By (4.10), $P_{1,G}(F_1(\bar{s}\bar{\sigma}), q_0^E) = P_{1,G}(F_3(\bar{s}\bar{\sigma}), q_0^E)$.

Case 2: $(q', \bar{\sigma}, \delta^E(\bar{\sigma}, q')) \in \delta_{1,o} \cap \delta_{2,o} \wedge (q, \bar{\sigma}, \delta^E(\bar{\sigma}, q)) \in \delta_{1,o} \backslash \delta_{2,o}$

According to Table 4.1 and Table 4.3, possible transition types are 2, 4, or 6.

For transition type 2, $F_1(\bar{\sigma}) = F_3(\bar{\sigma}) = \epsilon$. For types 4 and 6, $F_1(\bar{\sigma}) = F_3(\bar{\sigma}) = \bar{\sigma}$. They all follow the same arguments as in **Case 1**.

Case 3: $(q', \bar{\sigma}, \delta^E(\bar{\sigma}, q')) \in \delta_{1,o} \cap \delta_{2,o} \wedge (q, \bar{\sigma}, \delta^E(\bar{\sigma}, q)) \in \delta_{2,o} \backslash \delta_{1,o}$

According to Table 4.1 and Table 4.3, possible transition types are 2 or 5.

For transition type 2, $F_1(\bar{\sigma}) = F_3(\bar{\sigma}) = \epsilon$. It follows the same reasoning as in **Case 1**.

For transition type 5, $F_1(\bar{\sigma}) = \epsilon$ and $F_3(\bar{\sigma}) = \bar{\sigma}$. Then,

$$P_{1,G}(F_1(\bar{\sigma}), q') = \epsilon$$

$$P_{1,G}(F_3(\bar{\sigma}), q) = P_{1,G}(\bar{\sigma}, q) = \epsilon \quad (\text{since } (q, \bar{\sigma}, \delta^E(\bar{\sigma}, q)) \in \delta_{2,o} \backslash \delta_{1,o})$$

Therefore,

$$P_{1,G}(F_1(\bar{s}\bar{\sigma}), q_0^E) = P_{1,G}(F_3(\bar{s}\bar{\sigma}), q_0^E) \quad (\text{by (4.8), (4.9) and (4.10)})$$

Case 4: $(q', \bar{\sigma}, \delta^E(\bar{\sigma}, q')) \in \delta_{1,o} \cap \delta_{2,o} \wedge (q, \bar{\sigma}, \delta^E(\bar{\sigma}, q)) \notin \delta_{2,o} \cup \delta_{1,o}$

According to Table 4.1 and Table 4.3, possible transition types are 2, 3, or 5.

For types 3 and 5, $F_1(\bar{\sigma}) = \epsilon$ and $F_3(\bar{\sigma}) = \bar{\sigma}$. The argument follows the same reasoning as in **Case 3** for transition type 5.

For transition type 2, $F_1(\bar{\sigma}) = F_3(\bar{\sigma}) = \epsilon$. It follows the same reasoning as in **Case 1**.

Case 5: $(q', \bar{\sigma}, \delta^E(\bar{\sigma}, q')) \in \delta_{2,o} \backslash \delta_{1,o} \wedge (q, \bar{\sigma}, \delta^E(\bar{\sigma}, q)) \in \delta_{1,o} \backslash \delta_{2,o}$

According to Table 4.1 and Table 4.3, possible transition types are 1, 2 or 7.

For transition types 1 and 7, $F_1(\bar{\sigma}) = \bar{\sigma}$ and $F_3(\bar{\sigma}) = \epsilon$. Then,

$$P_{1,G}(F_1(\bar{\sigma}), q') = P_{1,G}(\bar{\sigma}, q') = \epsilon \quad (\text{since } (q', \bar{\sigma}, \delta^E(\bar{\sigma}, q')) \in \delta_{2,o} \backslash \delta_{1,o})$$

$$P_{1,G}(F_3(\bar{\sigma}), q) = \epsilon$$

Therefore,

$$P_{1,G}(F_1(\bar{s}\bar{\sigma}), q_0^E) = P_{1,G}(F_3(\bar{s}\bar{\sigma}), q_0^E) \quad (\text{by (4.8), (4.9) and (4.10)})$$

For transition type 2, $F_1(\bar{\sigma}) = F_3(\bar{\sigma}) = \epsilon$, therefore it follows the same arguments as in **Case 1**.

The proofs for the remaining 11 cases are analogous to cases 1–5; consequently, the proofs are omitted.

□

Claim 7. *For* $s^* \in L(\tilde{M}) \backslash L_m(\tilde{M})$, $P_{2,G}(F_3(s^*), q_0^E) = P_{2,G}(F_2(s^*), q_0^E)$.

Proof. The proof follows the same argument as **Claim 4**. □

By claims 1, 2, and 3, we know that $s, s', s'' \in L(E)$ and s, s', s'' end in q_3, q_1 and q_2 in E respectively. As we assumed at the outset, $s^*\sigma$ leads to the dump state. Therefore, by the definition of $\delta^{\bar{M}}$, $\delta^E(\sigma, q_3)$ is not defined ($s\sigma \notin L(E)$); $\delta^G(\sigma, q_4)$ is defined ($s\sigma \in L(G)$); $\delta^E(\sigma, q_1)$ is defined ($s'\sigma \in L(E)$), if $\sigma \in \Sigma_{1,c}$; $\delta^E(\sigma, q_2)$ is defined

($s''\sigma \in L(E)$), if $\sigma \in \Sigma_{2,c}$. By Claim 4 and Claim 5, $P_{1,G}(s', q_0^E) = P_{1,G}(s, q_0^E)$ and $P_{2,G}(s'', q_0^E) = P_{2,G}(s, q_0^E)$. All of the above indicate that s, s', s'' violate transition-based coobservability.

Only If: Assume E is not transition-based coobservable w.r.t. G and $P_{i,G}$. We need to prove that there exists a path in $\tilde{M}$ that leads to the dump state.

First we need to show that $\tilde{M}$ keeps track of all possible occurrences of strings s, s', s'' in its 4-tuple states such that $P_{1,G}(s, q_0^E) = P_{1,G}(s', q_0^E)$ and $P_{2,G}(s, q_0^E) = P_{2,G}(s'', q_0^E)$. The following claim is at the heart of the proof.

Claim 8. *Given $s, s', s'' \in L(E)$ such that $P_{1,G}(s, q_0^E) = P_{1,G}(s', q_0^E)$ and $P_{2,G}(s, q_0^E) = P_{2,G}(s'', q_0^E)$, then there exists a sequence $s^* \in L(\tilde{M}) \backslash L_m(\tilde{M})$ such that $F_1(s^*) = s', F_2(s^*) = s''$, and $F_3(s^*) = s$.*

Proof. By induction on the length of string s.

Basis: Suppose $|s| = 0$, then $s = \epsilon$.

Since $s = \epsilon$, we can assume s', s'' have the following forms:

$$s' = s'(1)s'(2)...s'(m) \quad (m \text{ is a natural number})$$

where for $i = 1, ..., m, s'(i) \in \Sigma$, and

$$(q_0^E, s'(1), \delta^E(s'(1), q_0^E)) \notin \delta_{1,o}$$

and for $i \geqslant 2$,

$$(\delta^E(s'(1)...s'(i-1), q_0^E), s'(i), \delta^E(s'(i), \delta^E(s'(1)...s'(i-1), q_0^E)) \notin \delta_{1,o}^E$$

i.e., s' is a sequence that Supervisor 1 can not see, and

$$s'' = s''(1)s''(2)...s'(n) \quad (n \text{ is a natural number})$$

where for $j = 1, ..., n, s''(j) \in \Sigma$, and

$$(q_0^E, s''(1), \delta^E(s''(1), q_0^E)) \notin \delta_{2,o}$$

and for $j \geqslant 2$,

$$(\delta^E(s''(1)...s''(j-1), q_0^E), s''(j), \delta^E(s''(j), \delta^E(s''(1)...s''(j-1), q_0^E)) \notin \delta_{2,o}^E$$

i.e., s'' is a sequence that Supervisor 2 can not see.

We define a path s^* in $\tilde{M}$ as

$$(q_0^E, q_0^E, q_0^E, q_0^G,)\underrightarrow{(s'(1), 1)}...\underrightarrow{(s'(m), 1)}\underrightarrow{(s''(1), 2)}...\underrightarrow{(s''(n), 2)}(q_1, q_2, q_3, q_4)$$

where (q_1, q_2, q_3, q_4) is not the dump state.

We claim $s^* = s's''$ is a sequence generated by $\tilde{M}$. Because $(q_0^E, s'(1), \delta^E(s'(1), q_0^E)) \notin \delta_{1,o}$, by Table 4.1, it can be represented as $1C$ or $1D$. In either case, transition type 1 can occur (by checking Table 4.3). And since $s' \in L(E)$, m transitions of type 1 are allowed in $\tilde{M}$. Since transition type 1 does not change the second argument in the 4-tuple state, the second value in the 4-tuple state is still q_0^E. Because $(q_0^E, s''(1), \delta^E(s''(1), q_0^E)) \notin \delta_{2,o}$, by checking Table 4.1, it can be represented by $2B$ or $2D$. From the definition of $\tilde{M}$ (Table 4.3), transition type 2 can occur.

Since $s'' \in L(E)$, we can add n transitions of type 2 after s'.

Since F_1 erases transition type 2, $F_1(s's'') = s'$; F_2 erases transition type 1, so $F_2(s's'') = s'$; F_3 erases transition types 1 and 2, so $F_3(s's'') = \epsilon = s$.

Inductive hypothesis:

Suppose the claim holds for all $s \leq n$, $n \geqslant 0$. Consider $t = s\sigma$ where $|s| = n, \sigma \in \Sigma$ and there exist $t, t', t'' \in L(E)$ so that $P_{1,G}(t, q_0^E) = P_{1,G}(t', q_0^E)$ and $P_{2,G}(t, q_0^E) = P_{2,G}(t'', q_0^E)$.

Case 1: $(\delta^E(s, q_0^E), \sigma, \delta^E(\sigma, \delta^E(s, q_0^E))) \in \delta_{1,o} \cap \delta_{2,o}$

Since $P_{1,G}(t, q_0^E) = P_{1,G}(t', q_0^E)$, t' can be represented as $s'\sigma v'$ where $P_{1,G}(s, q_0^E) = P_{1,G}(s', q_0^E)$, $(\delta^E(s', q_0^E), \sigma, \delta^E(\sigma, \delta^E(s', q_0^E))) \in \delta_{1,o}$ and v' is a sequence unobservable to Supervisor 1 $s'\sigma$ denoted by $v'(0)v'(1)...v'(k)$.

Since $P_{2,G}(t, q_0^E) = P_{2,G}(t'', q_0^E)$, t'' can be represented as $s''\sigma v''$ where $P_{2,G}(s, q_0^E) = P_{2,G}(s'', q_0^E)$, $(\delta^E(s'', q_0^E), \sigma, \delta^E(\sigma, \delta^E(s'', q_0^E))) \in \delta_{2,o}$ and v'' is an unobservable sequence to Supervisor 2 denoted by $v''(0)v''(1)...v''(l)$.

From the inductive hypothesis, because $P_{1,G}(s, q_0^E) = P_{1,G}(s', q_0^E)$ and $P_{2,G}(s, q_0^E) = P_{2,G}(s'', q_0^E)$, there exists a sequence $s^* \in L(\tilde{M}) \backslash L_m(\tilde{M})$ such that

$$F_3(s^*) = s$$

$$F_1(s^*) = s'$$

$$F_2(s^*) = s''$$

Assume s^* ends at (q_1, q_2, q_3, q_4) in $\tilde{M}$. From Claim 1, Claim 2 and Claim 3, s' leads to q_1 in E, s'' leads to q_2, and s leads to q_3 in E and to q_4 in G. Because

$t' = s'\sigma v' \in L(E)$, and $L(E)$ is prefix-closed, $s'\sigma \in L(E)$, which, in turn, implies that $\delta^E(\sigma, q_1)$ is defined. By the same reasoning, $\delta^E(\sigma, q_2)$ and $\delta^E(\sigma, q_3)$ are also defined. As we assumed before, E is a sub-automaton of G, so $s\sigma \in L(E)$ implies that $s\sigma \in L(G)$, which means $\delta^G(\sigma, q_4)$ is also defined.

Since σ is observable by Supervisor 1 after s', it corresponds to either scenario $1A$ or $1B$ from Table 4.1; since σ is observable by Supervisor 2 after s'', it corresponds to either scenario $2A$ or $2C$; and since σ is observable by both Supervisor 1 and Supervisor 2 after s, it corresponds to scenario $3A$. Therefore after s^* occurs, four categories are possible: $1A$, $2A$, $3A$; $1A$, $2C$, $3A$; $1B$, $2A$, $3A$; or $1B$, $2C$, $3A$. From Table 4.3, in all of the cases, transition type 4 could happen, i.e., $s^*(\sigma, 4)$ is a legitimate path in $\tilde{M}$ and it leads to state $(\delta^E(\sigma, q_1), \delta^E(\sigma, q_2), \delta^E(\sigma, q_3), \delta^G(\sigma, q_4))$.

If we add

$$(v'(0), 1)(v'(1), 1)...(v'(k), 1)(v''(0), 2)(v''(1), 2)...(v''(l), 2)$$

after $s^*(\sigma, 4)$, it is still a legitimate path in $\tilde{M}$ and will not lead to the dump state, following the same reasoning as we used in the basis step.

$$
\begin{aligned}
F_3(s^*\sigma v'v'') &= F_3(s^*)F_3(\sigma)F_3(v'v'') \\
&= sF_3(\sigma)F_3(v'v'') \quad \text{(by inductive hypothesis)} \\
&= s\sigma F_3(v'v'') \quad (\sigma \text{ comes from transition type 4}) \\
&= s\sigma \quad \text{(all transitions in } v' \text{ are type 1, all transitions in } v'' \text{ are} \\
&\qquad \text{type 2 and } F_3 \text{ erases both)} \\
&= t
\end{aligned}
$$

$$
\begin{aligned}
F_1(s^*\sigma v'v'') &= F_1(s^*)F_1(\sigma)F_1(v'v'') \\
&= s'F_1(\sigma)F_1(v'v'') \quad \text{(by inductive hypothesis)} \\
&= s'\sigma F_1(v'v'') \quad (\sigma \text{ comes from transition type 4}) \\
&= s'\sigma v' \quad \text{(all transitions in } v' \text{ are type 1, all transitions in } v'' \text{ are} \\
&\qquad \text{type 2 and } F_1 \text{ erases } v'') \\
&= t'
\end{aligned}
$$

$$
\begin{aligned}
F_2(s^*\sigma v'v'') &= F_2(s^*)F_2(\sigma)F_2(v'v'') \\
&= s''F_2(\sigma)F_2(v'v'') \quad \text{(by inductive hypothesis)} \\
&= s''\sigma F_2(v'v'') \quad (\sigma \text{ comes from transition type 4}) \\
&= s''\sigma v'' \quad \text{(all transitions in } v' \text{ are type 1, all transitions in } v'' \text{ are} \\
&\qquad \text{type 2 and } F_2 \text{ erases } v') \\
&= t''
\end{aligned}
$$

Case 2: $(\delta^E(s, q_0^E), \sigma, \delta^E(\sigma, \delta^E(s, q_0^E))) \in \delta_{1,o} \backslash \delta_{2,o}$

Since $P_{1,G}(t, q_0^E) = P_{1,G}(t', q_0^E)$, t' can be represented as $s'\sigma v'$ where $P_{1,G}(s, q_0^E) = P_{1,G}(s', q_0^E)$, $(\delta^E(s', q_0^E), \sigma, \delta^E(\sigma, \delta^E(s', q_0^E))) \in \delta_{1,o}$ and v' is a sequence unobservable to Supervisor 1 denoted by $v'(0)v'(1)...v'(k)$.

Since $(\delta^E(s, q_0^E), \sigma, \delta^E(\sigma, \delta^E(s, q_0^E))) \notin \delta_{2,o}$, $P_{2,G}(t, q_0^E) = P_{2,G}(s, q_0^E)$. Since $P_{2,G}(t, q_0^E) = P_{2,G}(t'', q_0^E)$, this means $P_{2,G}(s, q_0^E) = P_{2,G}(t'', q_0^E)$.

The strings s, s', and t'' satisfy the inductive hypothesis. So there is a sequence $s^* \in L(\tilde{M}) \backslash L_m(\tilde{M})$, such that

$$
\begin{aligned}
F_3(s^*) &= s \\
F_1(s^*) &= s' \\
F_2(s^*) &= t''
\end{aligned}
$$

Assume s^* ends in (q_1, q_2, q_3, q_4) in $\tilde{M}$. From Claim 1, Claim 2 and Claim 3, s'

leads to q_1 in E, and s leads to q_3 in E and to q_4 in G. Because $t' = s'\sigma v' \in L(E)$, and $L(E)$ is prefix-closed, $s'\sigma \in L(E)$, which, in turn, implies that $\delta^E(\sigma, q_1)$ is defined. Since $t = s\sigma \in L(E)$, by the same reasoning, $\delta^E(\sigma, q_3)$ is also defined. As we assumed before, E is a sub-automaton of G, so $s\sigma \in L(E)$ implies that $s\sigma \in L(G)$, which means $\delta^G(\sigma, q_4)$ is also defined.

Since σ is observable by Supervisor 1 after s', it corresponds to either $1A$ or $1B$ from Table 4.1; since σ is observable only by Supervisor 1 after s, it corresponds to $3B$. From Table 4.3, in each category with scenarios $1A$ and $3B$ or scenarios $1B$ and $3B$, transition type 6 can occur, i.e., $s^*(\sigma, 6)$ is a legitimate path in $\tilde{M}$ and it leads to $(\delta^E(\sigma, q_1), q_2, \delta^E(\sigma, q_3), \delta^G(\sigma, q_4))$.

Then we add $(v'(0), 1)(v'(1), 1)...(v'(k), 1)$ following $s^*(\sigma, 6)$, which is still a legitimate path in $\tilde{M}$.

$$
\begin{aligned}
F_3(s^*\sigma v') &= F_3(s^*)F_3(\sigma)F_3(v') \\
&= sF_3(\sigma)F_3(v') \quad \text{(by inductive hypothesis)} \\
&= s\sigma F_3(v') \quad (\sigma \text{ comes from transition type 6}) \\
&= s\sigma \quad \text{(all transitions in } v' \text{ are type 1 and } F_3 \text{ erases type 1 transitions)} \\
&= t
\end{aligned}
$$

$$\begin{aligned} F_1(s^*\sigma v') &= F_1(s^*)F_1(\sigma)F_1(v') \\ &= s'F_1(\sigma)F_1(v') \quad \text{(by inductive hypothesis)} \\ &= s'\sigma F_1(v') \quad (\sigma \text{ comes from transition type 6}) \\ &= s'\sigma v' \quad \text{(all transitions in } v' \text{ are type 1 and } F_1 \text{ preserves type 1 transitions)} \\ &= t' \end{aligned}$$

$$\begin{aligned} F_2(s^*\sigma v') &= F_2(s^*)F_2(\sigma)F_2(v') \\ &= t''F_2(\sigma)F_2(v') \quad \text{(by inductive hypothesis)} \\ &= t''F_2(v') \quad (\sigma \text{ comes from transition type 6, } F_2 \text{ erases it}) \\ &= t'' \quad \text{(all transitions in } v' \text{ are type 1 and } F_2 \text{ erases type 1 transitions)} \end{aligned}$$

Case 3: $(\delta^E(s, q_0^E), \sigma, \delta^E(\sigma, \delta^E(s, q_0^E))) \in \delta_{2,o} \backslash \delta_{1,o}$

Since $(\delta^E(s, q_0^E), \sigma, \delta^E(\sigma, \delta^E(s, q_0^E))) \notin \delta_{1,o}$, $P_{1,G}(t, q_0^E) = P_{1,G}(s, q_0^E)$. Since $P_{1,G}(t, q_0^E) = P_{1,G}(t', q_0^E)$, this means $P_{1,G}(s, q_0^E) = P_{1,G}(t', q_0^E)$.

Since $P_{2,G}(t, q_0^E) = P_{2,G}(t'', q_0^E)$, t'' can be represented as $s''\sigma v''$ where $P_{2,G}(s, q_0^E) = P_{2,G}(s'', q_0^E)$, $(\delta^E(s'', q_0^E), \sigma, \delta^E(\sigma, \delta^E(s'', q_0^E))) \in \delta_{2,o}$ and v'' is a sequence unobservable to Supervisor 2 denoted by $v''(0)v''(1)...v'(l)$.

Since the strings s, t', and s'' satisfy the inductive hypothesis, there is a sequence $s^* \in L(\tilde{M}) \backslash L_m(\tilde{M})$, such that

$$F_3(s^*) = s$$

$$F_1(s^*) = t'$$

$$F_2(s^*) = s''$$

Assume s^* ends in (q_1, q_2, q_3, q_4) in $\tilde{M}$. Using reasoning analogous to that for Case 1 and Case 2, we can see that $\delta^E(\sigma, q_2)$, $\delta^E(\sigma, q_3)$ and $\delta^G(\sigma, q_4)$ are defined.

Since σ is observable by Supervisor 2 after s'', it corresponds to either $2A$ or $2C$ according to Table 4.1; since σ is observable only by Supervisor 2 after s, it corresponds to $3C$. From Table 4.3, in each category with scenarios $2A$ and $3C$ or scenarios $2C$ and $3C$, transition type 5 can occur, i.e., $s^*(\sigma, 5)$ is a legitimate path in $\tilde{M}$ and it leads to state $(q_1, \delta^E(\sigma, q_2), \delta^E(\sigma, q_3), \delta^G(\sigma, q_4))$.

Then we add $(v''(0), 2)(v'(1), 2)...(v''(l), 2)$ following $s^*(\sigma, 5)$, which is still a legitimate path in $\tilde{M}$.

$$\begin{aligned}
F_3(s^*\sigma v'') &= F_3(s^*)F_3(\sigma)F_3(v'') \\
&= sF_3(\sigma)F_3(v'') \quad \text{(by inductive hypothesis)} \\
&= s\sigma F_3(v'') \quad (\sigma \text{ comes from transition type 5}) \\
&= s\sigma \quad \text{(all transitions in } v'' \text{ are type 2 and } F_3 \text{ erases transition type 2)} \\
&= t
\end{aligned}$$

$$\begin{aligned} F_1(s^*\sigma v'') &= F_1(s^*)F_1(\sigma)F_1(v'') \\ &= t'F_1(\sigma)F_1(v'') \quad \text{(by inductive hypothesis)} \\ &= t'F_1(v'') \quad (\sigma \text{ comes from transition type 5}) \\ &= t' \quad \text{(all transitions in } v'' \text{ are type 1 and } F_1 \text{ erases transition type 1)} \end{aligned}$$

$$\begin{aligned} F_2(s^*\sigma v'') &= F_2(s^*)F_2(\sigma)F_2(v'') \\ &= s''F_2(\sigma)F_2(v'') \quad \text{(by inductive hypothesis)} \\ &= s''\sigma F_2(v'') \quad (\sigma \text{ comes from transition type 5}) \\ &= s''\sigma v'' \quad \text{(all transition in } v'' \text{ are type 2 and } F_2 \text{ preserves transition type 2)} \\ &= t'' \end{aligned}$$

Case 4: $(\delta^E(s, q_0^E), \sigma, \delta^E(\sigma, \delta^E(s, q_0^E))) \notin \delta_{1,o} \cup \delta_{2,o}$

Since $(\delta^E(s, q_0^E), \sigma, \delta^E(\sigma, \delta^E(s, q_0^E))) \notin \delta_{1,o}$, $P_{1,G}(t, q_0^E) = P_{1,G}(s, q_0^E)$. Since $P_{1,G}(t, q_0^E) = P_{1,G}(t', q_0^E)$, this means $P_{1,G}(s, q_0^E) = P_{1,G}(t', q_0^E)$.

Since $(\delta^E(s, q_0^E), \sigma, \delta^E(\sigma, \delta^E(s, q_0^E))) \notin \delta_{2,o}$, $P_{2,G}(t, q_0^E) = P_{2,G}(s, q_0^E)$. Since $P_{2,G}(t, q_0^E) = P_{2,G}(t'', q_0^E)$, this means $P_{2,G}(s, q_0^E) = P_{2,G}(t'', q_0^E)$.

So s, t', and t'' satisfy the inductive hypothesis. Therefore, there is a sequence $s^* \in L(\tilde{M}) \backslash L_m(\tilde{M})$, such that

$$F_3(s^*) = s$$

$$F_1(s^*) = t'$$

$$F_2(s^*) = t''$$

Assume s^* ends in (q_1, q_2, q_3, q_4) in $\tilde{M}$. Using reasoning analogous to that for Case 1 and Case 2, we can see $\delta^E(\sigma, q_3)$ and $\delta^G(\sigma, q_4)$ are defined.

Since σ is not observable by either Supervisor 1 or Supervisor 2 after s, it corresponds to $3D$ from Table 4.1. From Table 4.3, in each category with scenario $3D$, transition type 3 can occur, i.e., $s^*(\sigma, 3)$ is a legitimate path in $\tilde{M}$ and it leads to $(q_1, q_2, \delta^E(\sigma, q_3), \delta^G(\sigma, q_4))$.

$$\begin{aligned} F_3(s^*\sigma) &= F_3(s^*)F_3(\sigma) \\ &= sF_3(\sigma) \quad \text{(by inductive hypothesis)} \\ &= s\sigma \quad (\sigma \text{ comes from transition type 3}) \\ &= t \end{aligned}$$

$$\begin{aligned} F_1(s^*\sigma) &= F_1(s^*)F_1(\sigma) \\ &= t'F_1(\sigma) \quad \text{(by inductive hypothesis)} \\ &= t' \quad (\sigma \text{ comes from transition type 3, } F_1 \text{ erases it)} \end{aligned}$$

$$\begin{aligned} F_2(s^*\sigma) &= F_2(s^*)F_2(\sigma) \\ &= t''F_2(\sigma) \quad \text{(by inductive hypothesis)} \\ &= t'' \quad (\sigma \text{ comes from transition type 3, } F_2 \text{ erases it.)} \end{aligned}$$

□

Now we return to the proof of Proposition 1. Since E is not transition-based coobservable w.r.t. G and $P_{i,G}$ $(i = 1, 2)$, one of the three conjuncts of transition-based coobservability would fail. Therefore there exist $s, s', s'' \in \bar{E}$ such that $P_{1,G}(s, q_0) = P_{1,G}(s', q_0)$ and $P_{2,G}(s, q_0) = P_{2,G}(s'', q_0)$ and either one of the following three cases exists:

$$(\exists \sigma \in \Sigma_{1,c} \cap \Sigma_{2,c})\, s \in L(E) \wedge s\sigma \in L(G) \wedge s'\sigma, s''\sigma \in L(E) \text{ but } s\sigma \notin L(E) \quad (4.11)$$

$$(\exists \sigma \in \Sigma_{1,c} \setminus \Sigma_{2,c})\, s \in L(E) \wedge s\sigma \in L(G) \wedge s'\sigma \in L(E) \text{ but } s\sigma \notin L(E) \quad (4.12)$$

$$(\exists \sigma \in \Sigma_{2,c} \setminus \Sigma_{1,c})\, s \in L(E) \wedge s\sigma \in L(G) \wedge s''\sigma \in L(E) \text{ but } s\sigma \notin L(E) \quad (4.13)$$

By Claim 6, there exists $s^* \in L(\tilde{M}) \backslash L_m(\tilde{M})$ such that $F_1(s^*) = s', F_2(s^*) = s''$, and $F_3(s^*) = s$. Suppose s^* leads to (q_1, q_2, q_3, q_4) in $\tilde{M}$. By Claim 1, Claim 2, and Claim 3, s, s', s'' lead to states q_3, q_1, q_2 in E, respectively, and s leads to state q_4 in G.

Suppose that (4.11) is the situation that violates transition-based coobservability. Then $\delta^E(\sigma, q_1)$ is defined, $\delta^E(\sigma, q_2)$ is defined, $\delta^G(\sigma, q_4)$ is defined, but $\delta^E(\sigma, q_3)$ is not defined. By the definition of when $\delta^{\tilde{M}}$ leads to the dump state in $\tilde{M}$, as seen in (4.1), the sequence $s^*\sigma$ leads to the dump state. The cases for (4.12) and (4.13) follow similar reasoning.

□

Chapter 5

Communication in Distributed DES Control

In Chapter 3, we proved that if the legal behavior $L(E)$ is controllable with regard to the plant G, then the condition needed to implement the control by two decentralized supervisors is transition-based coobservability. If a system is not coobservable—which is based on the events each supervisor directly observes, then transition-based coobservability could be used in an approach that incorporates communication between supervisors, as follows. First one would find a set of transitions whose observation would make the system transition-based coobservable. Then, one would develop a communication scheme whereby each supervisor sends the other supervisor information that conveys when transitions in the aforementioned set occur.

The recent model for communication [10] is that two supervisors cooperate to perform a system-level task. It is assumed that we already know the supervisors' structure. In this scenario, the aim of communication between supervisors is making them know where they are at at all times without any control action.

In this chapter, an algorithm is given to exploit a theoretical solution to solve the control and communication problem. Our approach of decentralized control and communication is based on traditional decentralized discrete-event system control [13]. When controllability is satisfied and coobservability is not satisfied, we involve communication between supervisors to ensure transition-based coobservability. We use a modified M-machine to check transition-based coobservability and find a transition to be communicated step by step until transition-based coobservability is finally satisfied. The detailed computing procedure will be explained in Section 5.2. In summary, we are interested in when communication between supervisors can solve decentralized DES control problems, instead of ensuring that each supervisor always knows which state it is at. So far, we have not attempted to optimize our algorithm to make it usable in practical applications and we do not know the scalability of our algorithm.

In this chapter, we first discuss, in detail, some issues related to communication. Then we give an algorithm to find a communication protocol between two supervisors. At the end we use an example to explain how the algorithm works.

5.1 Communication Consistency

Finding a suitable communication protocol is not trivial. Since a supervisor can only observe some of the events occurring, it can not distinguish some of the states in the system. Therefore, it must make the same communications in the states which look the same to it. This is a property called *consistency*.

Take the example in Section 3.3: If Supervisor 2 communicates β at State 1 of the plant G, it also needs to communicate β at State 0, since it does not observe whether α happens in State 0. To Supervisor 2, State 0 and State 1 are indistinguishable.

Therefore, if transition $(1, \beta, 4)$ needs to be communicated to satisfy transition-based coobservability, we also add $(0, \beta, 2)$ to the communication protocol to ensure communication consistency.

In [10], a function $\mathcal{N}_{12}(V_{12}, V_{21})$ is given to compute minimal additional communication needed to make Supervisor 1 consistent given that V_{12} is the set of transitions Supervisor 1 communicates to Supervisor 2 and V_{21} is the set Supervisor 2 communicates to Supervisor 1. Here we use a similar function $\mathcal{N}_{12}$ to represent the same procedure as in [10].

The notation used in the following function is as follows. The parameter

$$V_{21} = \{(q, \sigma, \delta^E(q, \sigma)), ...)\}$$

is a set of transitions in E whose event labels will be communicated from Supervisor 2 to Supervisor 1. Supervisor 1 can see only the events in $\Sigma_{1,o}$ by direct observation and the events occurring in V_{21} by communication. We replace all other unobservable transitions in E by the empty string ϵ. The resulting automaton

$$E_1^\epsilon(V_{21}) = (\Sigma, Q^E, \delta_1^\epsilon, q_0)$$

is an NFA. It is Supervisor 1's view of the legal automaton E. We then transform E_1^ϵ from an NFA to a DFA, an operation we denote by DA. The resulting DFA is denoted by $\tilde{E}_1$:

$$\tilde{E}_1(V_{21}) = DA(E_1^\epsilon(V_{21})) := (\Sigma, \tilde{Q^E}, \tilde{\delta}_1, \tilde{q_0})$$

The automaton $E_2^\epsilon(V_{12})$ and $\tilde{E}_2(V_{12})$ are defined by interchanging 1 and 2 in the above definitions.

Function $\mathcal{N}_{12}(V_{12}, V_{21})$

/*V_{12} is the set of transitions whose event labels are communicated from Supervisor 1 to Supervisor 2. V_{21} is the set of transitions whose event labels are communicated from Supervisor 2 to Supervisor 1.*/

Input: $E = (\Sigma, Q^E, \delta^E, q_0^E)$, V_{12}, V_{21}

Output: N_{12}

/*N_{12} is the set of transitions whose event labels need to be communicated from Supervisor 1 to Supervisor 2 to guarantee consistency for Supervisor 1.*/

1. $N_{12} := \emptyset$
2. Transition (E_1^ϵ):=$\emptyset$
3. For all $(q, \sigma, \delta^E(q, \sigma)) \in$ Transition(E) do

   If $\sigma \in \Sigma \backslash \Sigma_{1,o} \wedge (q, \sigma, \delta^E(q, \sigma) \notin V_{21})$

   Transition(E_1^ϵ) := Transition$(E_1^\epsilon) \cup \{(q, \epsilon, \delta^E(q, \sigma))\}$

   else

   Transition(E_1^ϵ) := Transition$(E_1^\epsilon) \cup \{(q, \sigma, \delta^E(q, \sigma))\}$
4. $E_1^\epsilon := \{\Sigma, Q^E, Transition(E_1^\epsilon), q_0^E\}$
5. $\tilde{E}_1^\epsilon := DA\{E_1^\epsilon\} = \{\Sigma, \tilde{Q}_1, \tilde{\delta}_1, \tilde{q_{1,0}}\}$
6. $W_{12} := \emptyset$
7. For all $\tilde{q} \in \tilde{Q}_1$, do

   If $(\exists q,\ q' \in \tilde{q})(q, \sigma, \delta^E(q, \sigma)) \notin W_{12} \cup V_{12} \wedge (q', \sigma, \delta^E(q', \sigma)) \in W_{12} \cup V_{12}$,

   then $N_{12} := N_{12} \cup \{(q, \sigma, \delta^E(q, \sigma))\}$

8. If $N_{12} \neq W_{12}$, then $W_{12} := N_{12}$, go to 7;
   else return.

The fact that the additional transitions to be communicated, determined by the functions $\mathcal{N}_{12}$ and $\mathcal{N}_{21}$, guarantee each supervisor's consistency is proved in [10], because the legal automaton E we used as input is just a special case of the parameter $R = R_1 \times R_2$ used in [10].

5.2 An Algorithm for Communication

Let us assume that the system is globally observable, i.e., with $\Sigma_o = \Sigma_{1,o} \cup \Sigma_{2,o}$, $L(E)$ is observable with respect to the plant G and projection $P : \Sigma^* \rightarrow \Sigma_o^*$. Therefore, if each supervisor communicates whatever it observes to the other supervisor, the system is surely transition-based coobservable. However, since communications may be costly, we only want communications necessary to ensure transition-based coobservability and consistency, in another words, a minimal communications set. A set of communications is called minimal when it (a) satisfies consistency, (b) provides enough information to solve the control problem and (c) no subset of it satisfies (a) and (b) [10].

An algorithm was proposed in [10] that finds the minimal communication pair between two supervisors so that they each always know at which state they are all the time. The modelling framework is that two supervisors cooperate to perform a system-level job.

In [4], another more general approach for minimal communication in a distributed discrete-event system is proposed. Instead of each supervisor always knowing its

current state, it always "observes" the event labelling of a specific set of transitions. This set is named *essential transitions*. In that paper, the essential transitions are prescribed at the outset and the paper focuses on an algorithm for guaranteeing a consistent minimal communication scheme that ensures all essential transitions are observed.

Now we propose an algorithm to find essential transitions to guarantee transition-based coobservability. The communication protocol found here guarantees consistency. It remains to be investigated whether our algorithm yields a minimal communication scheme.

We denote by com_{12} the communication set of transitions from Supervisor 1 to Supervisor 2; com_{21} is the set of transitions from Supervisor 2 to Supervisor 1.

Informally speaking, the strategy is as follows. Initially, com_{12} and com_{21} are set to be empty, then we start to construct the M-machine $\tilde{M}_0$. (Because there are no communications yet, we check coobservability using the M-machine, which is a special case of the modified M-machine.) If there is a sequence $s^M\sigma$ leading to the dump state, which means E is not coobservable, we stop the construction process. If $\sigma \in \Sigma_{1,c} \setminus \Sigma_{2,c}$, which means the second condition of coobservability is violated, i.e., Supervisor 1 doesn't have enough information to distinguish s and s' which look the same to it but require different control actions. We extract s and s' from s^M. To make Supervisor 1 distinguish s and s', we choose an event of a transition from either s or s' which is observed by Supervisor 2 to be communicated to Supervisor 1, i.e., add the transition to com_{21}. On the other hand, if $\sigma \in \Sigma_{2,c} \setminus \Sigma_{1,c}$, we choose an event of a transition from either s or s'' and add the transition to com_{12}, by the same reasoning. If $\sigma \in \Sigma_{1,c} \cup \Sigma_{2,c}$, then we randomly follow either of the above two

cases. We use functions named $\mathcal{C}_{12}$ and $\mathcal{C}_{21}$ to find the transitions that need to be communicated.

After we update com_{12} or com_{21}, we need to check consistency by using $\mathcal{N}_{21}$ or $\mathcal{N}_{12}$. Then we start constructing a new modified M-machine $\tilde{M}_1$ until we hit the dump state, and follow the same process as above. In the setting of this algorithm, when we construct $\tilde{M}_i$, we don't need to consider all the 64 cases in the transition table (Table 4.2). For example, case $1A, 2A, 3D$ will not happen. The scenario $3D$ means the transition $(\delta^E(s, q_0), \sigma, \delta^E(\sigma, \delta^E(s, q_0)))$ is unobservable by both supervisors, therefore the event σ is unobservable by both supervisors since a supervisor's observation is based only on its direct observation and on communications from the other supervisor. All cases with D as one of the scenarios will not happen except case $1D, 2D, 3D$. By the same reasoning, scenario B and C would not appear in the same case. Consequently, we need only consider 16 of the original 64 cases; it is impossible for the other 48 cases to occur.

The program stops when $\tilde{M}_i$ doesn't have a dump state. The last pair of updated transitions sets (com_{12}, com_{21}) is a consistent communication protocol. However, the communication pair may not be a minimal one, because the algorithm is iterative. A transition added to keep a supervisor's consistency may not be necessary if the supervisor gets more information about the plant in subsequent iterations.

In the function $\mathcal{C}_{21}$ which will be defined below, we borrow a concept from [9], called *a maximal-P pair*, to mark the last place two sequences look alike to a global observer.

Denote by P the projection from Σ^* to $(\Sigma_{1,o} \cup \Sigma_{2,o})^*$. Then a *maximal-P pair* of (s, s') is a pair of sequences (t, t') in Σ^*, where $t \in \bar{s}$, $t' \in \bar{s'}$, $P(t) = P(t')$ and $\nexists \sigma \in \Sigma$

such that $t\sigma \in \bar{s}$ and $P(t\sigma) = P(t')$ or $t'\sigma \in \bar{s'}$ and $P(t) = P(t'\sigma)$.

We also represent s and s' as $s(1)s(2)...$ and $s'(1)s'(2)....$ To simplify the notation of a transition in sequences s, s', we use $\delta^E[s(k)]$ to represent the transition $(\delta^E(s(1)..s(k-1), q_0^E), s(k), \delta^E(s(1)..s(k), q_0^E))$ (if $k \geq 2$) and $(q_0^E, s(1), \delta^E(s(1), q_0^E))$ (if $k = 1$) in the following definition of function $\mathcal{C}_{21}$. We call the k in $\delta^E[s(k)]$ the *index number* of the transition.

Function $\mathcal{C}_{21}(s, s')$

/* s, s' are sequences in $L(E)$ that are indistinguishable to Supervisor 1. */

Input: s, s'

Output: T_{21}

/* T_{21} is a transition whose event label needs to be communicated from Supervisor 2 to Supervisor 1 so that Supervisor 1 can distinguish sequences s and s' after the communication. */

1. /* If (t, t') is the maximal-P pair of (s, s'), then r is the length of t, r' is the length of t'. The index i will be used to mark the first transition observable to Supervisor 1 after t in s , and the index j will be used to mark the first transition observable to Supervisor 1 after t' in s'. */

   $r = 0$, $r' = 0$, $i = 1$, $j = 1$

2. Find the maximal-P pair of (s, s'): (t, t').

   $r = |t|$, $r' = |t'|$

3. In sequence s, find the next transition observable to Supervisor 1 after t, $\delta^E|s(i)|$.

   In sequence s', find the next transition observable to Supervisor 1 after t',

$\delta^E|s(j)|$.

4. Construct two sets of transitions X_1, X_2.

   /* The set X_1 is a set containing all the transitions after t and before transition $\delta^E|s(i)|$ in s. The set X_2 is a set containing all the transitions after t' and before transition $\delta^E|s(j)|$ in s'. */

   $X_1 = \{\delta^E[s(r+1)], \delta^E[s(r+2)], ..., \delta^E[s(i-1)]\}$

   If $r+1 > a-1$, then $X_1 = \emptyset$.

   $X_2 = \{\delta^E[s'(r+1)], \delta^E[s'(r+2)], ..., \delta^E[s(b-1)]\}$

   If $r+1 > b-1$, then $X_2 = \emptyset$.

5. Select a transition x from $X_1 \setminus X_2$ or from $X_2 \setminus X_1$ which is observable by Supervisor 2 and such that no other transition in $X_1 \setminus X_2$ or $X_2 \setminus X_1$ has a higher index number.

   Let $T_{21} = x$.

The function works as follows: The pair of sequences (t, t') we get from step 2 is the maximal-P pair of the sequences (s, s'). Index i marks the first observable transition to Supervisor 1 after t and index j marks the first observable transition to Supervisor 1 after t'. Since the aim of the function is to find a transition in s or s' so that Supervisor 1 can distinguish these two sequences, we select a transition between index r to i in sequence s or between index r' to j in sequence s' which satisfies the following conditions: (1) the transition is observable to Supervisor 2 but not observable to Supervisor 1, (2) if it is a transition from s, it should not be a transition in s' between index r' and j and vise versa, because communicating such

a transition would not help Supervisor 1 to distinguish the two sequences. We select a transition satisfying the above conditions with the highest index. The function $\mathcal{C}_{12}(s, s'')$ is defined in a similar way by interchanging 1 and 2 and using s'' to replace s' everywhere in the above function.

The overall algorithm is shown here. We call it **Main** algorithm.

Main

Input: G, E

/* G is the plant and E is the legal automaton.*/

Output: com_{12}, com_{21}

/*com_{12} is the communication set from Supervisor 1 to Supervisor 2 and com_{21} is the communication set from Supervisor 2 to Supervisor 1./*

1. $com^*_{12} = \emptyset$, $com^*_{21i} = \emptyset$, $i = 0$

2. Start constructing $\tilde{M}_i(G, E, com^*_{12}, com^*_{21})$ from the initial state, add reachable states using a depth-first search or a breadth-first search.

   If the dump state is reached, then stop the construction process.

   If no dump state is reached after the whole construction, $com_{12} = com^*_{12}$ and $com_{21} = com^*_{21}$, the algorithm terminates.

3. Obtain the sequence $s^M\sigma$ leading to the dump state.

   Case 1: $\sigma \in \Sigma_{1,c} \setminus \Sigma_{2,c}$

   extract s, s' from s^M,

   /* T_{21} is a transition communicated from Supervisor 2 to Supervisor 1 so Supervisor 1 can distinguish sequences s and s'. */

   $T_{21} = \mathcal{C}_{21}(s, s')$.

$com^*_{21} = com^*_{21} \cup \{T_{21}\}$

/* check consistency using $\mathcal{N}_{21}$*/

$com^*_{21} = \mathcal{N}_{21}(com^*_{21}, com^*_{12}) \cup com^*_{21}$

$i = i + 1$

go to 2.

Case 2: $\sigma \in \Sigma_{2,c} \setminus \Sigma_{1,c}$

extract s, s'' from s^M,

/* T_{12} is a transition communicated from Supervisor 1 to Supervisor 2 so Supervisor 2 can distinguish sequences s and s''. */

$T_{12} = \mathcal{C}_{12}(s,\, s'')$

$com^*_{12} = com^*_{12} \cup \{T_{12}\}$

/* check consistency using $\mathcal{N}_{12}$ */

$com^*_{12} = \mathcal{N}_{12}(com^*_{12}, com^*_{21}) \cup com^*_{12}$

$i = i + 1$

go to 2.

Case 3: $\sigma \in \Sigma_{2,c} \cup \Sigma_{1,c}$

follow either Case 1 or Case 2.

5.3 An Example

We illustrate the mechanics of the algorithm with an example. Consider the finite-state automata G and E given in Figure 5.1. The dotted lines are the illegal transitions, i.e., the transitions not in E. The alphabet Σ is $\{a, b, c, g\}$. Two supervisors have only partial observation

$$\Sigma_{1,o} = \{a, g\},\ \Sigma_{2,o} = \{b, c\}$$

and partial control

$$\Sigma_{1,c} = \{a, g, \},\ \Sigma_{2,c} = \{b, c\}$$

.

We start the **Main** algorithm by constructing $\tilde{M}_0$ which is the same as the M-machine since there is no communication yet. Suppose the procedure finds a sequence $s^M\sigma$ leading to the dump state of $\tilde{M}_0$ as shown in Figure 5.2.

Since $s^M\sigma = (b,5)(a,6)(b,1)(a,0)$ and $\sigma = a \in \Sigma_{1,c} \setminus \Sigma_{2,c}$, we extract s and s' using $F_i\,(i = 1, 3)$ as in Chapter 4. We get

$$s = ba$$

$$s' = ab$$

By function $\mathcal{C}_{21}(s, s')$, after step 2 finishes, we find the sequences t and t' that form the maximal-P pair of (s, s'): $t = \epsilon$ and $t' = \epsilon$. Therefore, $r = r' = 0$. The next

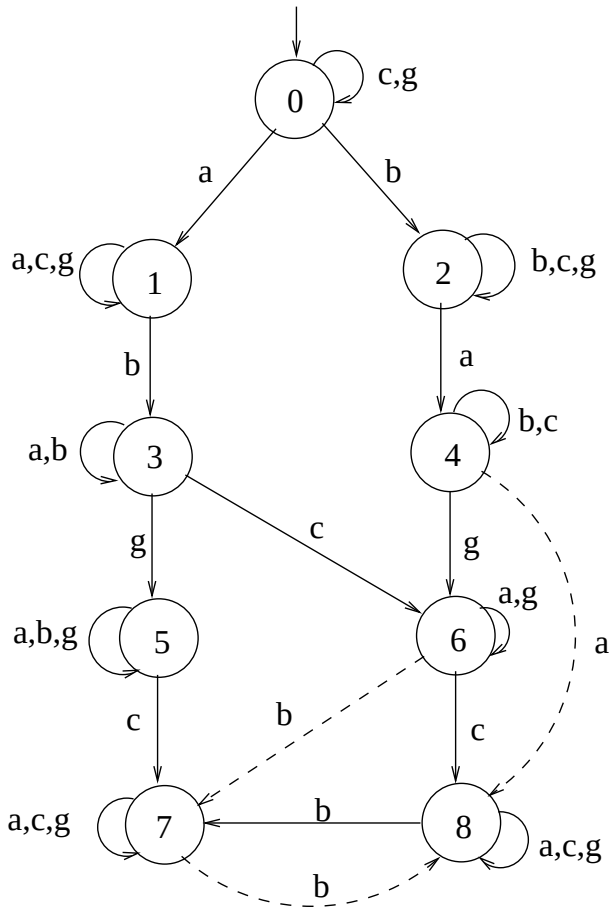

Figure 5.1: Plant and Legal Automaton in Example of Section 5.3

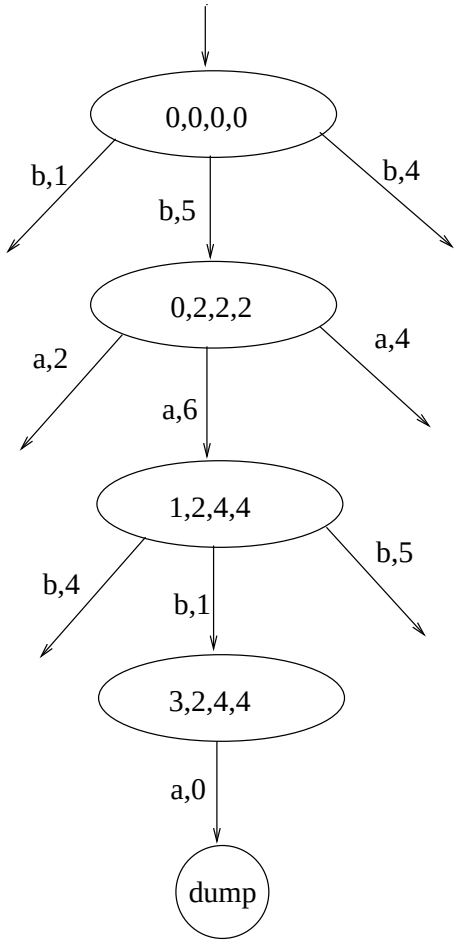

Figure 5.2: A Portion of the Modified M-machine $\tilde{M}_0$

transition observable to Supervisor 1 but not observable to Supervisor 2 in sequence s is $(\delta^E(b, q_0^E), a, \delta^E(ba, q_0^E))$, which is $(2, a, 4)$, i.e., the index $i = 2$. The transition before that which is observable to Supervisor 2 but not observable to Supervisor 1 is $(0, b, 2)$, therefore $X_1 = \{(0, b, 2)\}$. Since in s' the next transition observable to Supervisor 1 but not observable to Supervisor 2 from index $r' = 0$ is the first transition, we set $j = 1$. Because $r' + 1 > j - 1$, $X_2 = \emptyset$. There is only one element in X_1 and $X_2 = \emptyset$. We set $T_{21} = (0, b, 2)$ and add it to com_{21}^*.

With $com_{21}^* = \{(0, b, 2)\}$, $com_{12}^* = \emptyset$, we perform the function $\mathcal{N}_{21}(V_{21}, V_{12})$ where $V_{21} = com_{21}^*$ and $V_{12} = com_{12}^*$. First, we create E_2^ϵ (steps 2–4 of the function) by starting from E and replacing the event labels of the transitions that are not in $\Sigma_{2,o}$ and not in V_{12} with ϵ. This is displayed in Figure 5.3.

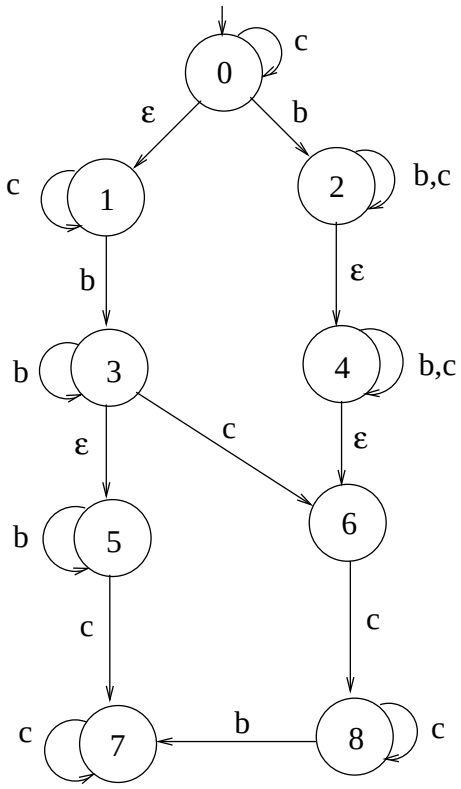

Figure 5.3: NFA of E_2^ϵ, Iteration 1

Then we transform the nondeterministic finite automaton into an equivalent deterministic finite automaton, which is displayed in Figure 5.4.

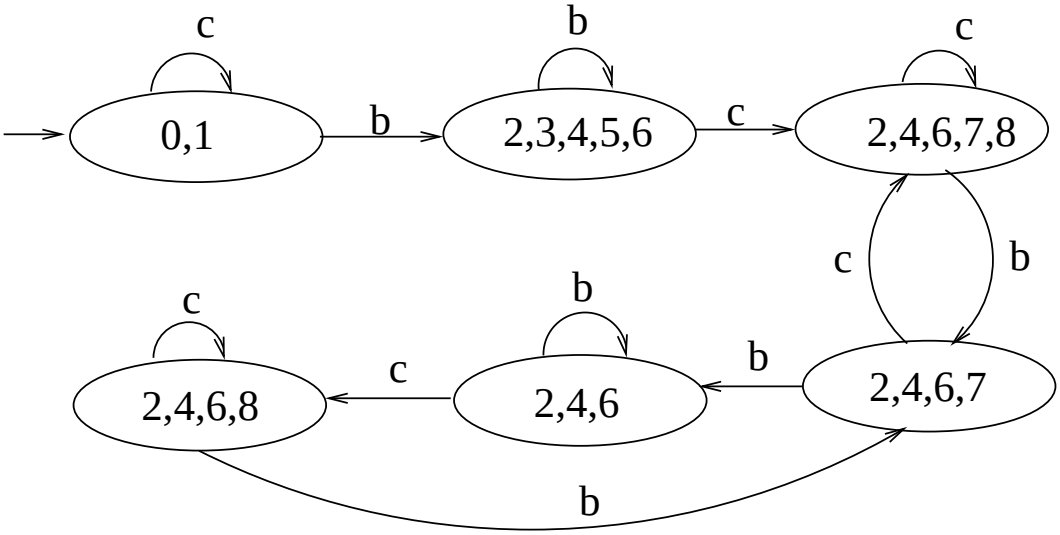

Figure 5.4: DFA of $\tilde{E}_2^{\epsilon}$, Iteration 1

We compute N_{21} in steps 6–8 of Function $\mathcal{N}_{21}$ using $\tilde{E}_2^{\epsilon}$. By searching at all $\tilde{q} \in \tilde{Q}_1$ where 0 is an element of $\tilde{q}$, there is one state: $\{0, 1\}$. Since $(0, b, 2) \in V_{12} \cup W_{12}$, but $(1, b, 3) \notin V_{12} \cup W_{12}$, we add $(1, b, 3)$ to N_{21}. After updating W_{21} in step 8 of function $\mathcal{N}_{21}$, nothing needs to be added to N_{21} for the next iteration of step 7. By now, the communication sets are: $com_{21}^* = \{(0, b, 2), (1, b, 3)\}$, $com_{12}^* = \emptyset$.

With the new communication pair, we start constructing a new modified M-machine $\tilde{M}_1$. Suppose $s^M\sigma = (a, 6)(b, 4)(a, 2)(c, 4)(b, 0)$ is the first sequence in computing $\tilde{M}_1$ which ends at the dump state as shown in Figure 5.5.

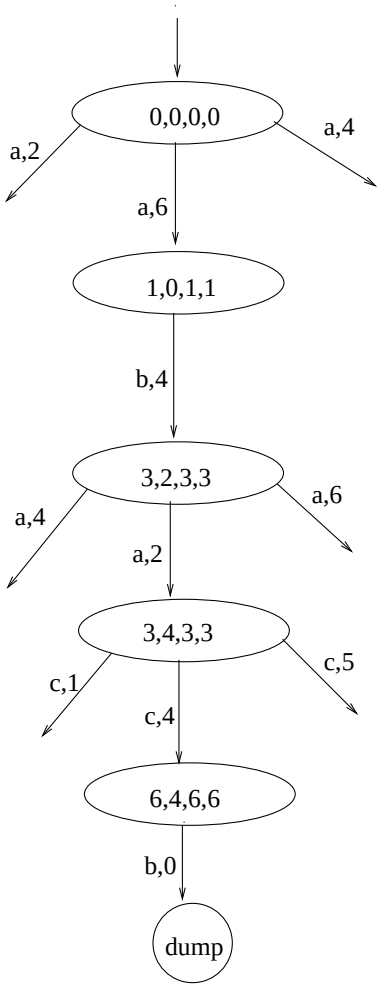

Figure 5.5: A Portion of the Modified M-machine $\tilde{M}_1$

Since $s^M\sigma = (a,6)(b,4)(a,2)(c,4)(b,0)$ and $\sigma = b \in \Sigma_{2,c} \setminus \Sigma_{1,c}$, we extract s and s'' using $F_i\,(i = 2,3)$ as in Chapter 4. We get

$$s = abc$$

$$s'' = bac$$

By function $\mathcal{C}_{12}(s, s'')$, after step 2 finishes, we have $r = r' = 0$. The next transition observable to Supervisor 2 but not observable to Supervisor 1 in sequence s is $(\delta^E(a, q_0^E), b, \delta^E(ab, q_0^E))$, which is $(1, b, 3)$. The transition right before that which is observable to Supervisor 1 but not observable to Supervisor 2 is $(0, a, 1)$, therefore $X_1 = \{(0, a, 1)\}$. Since in s'' the next transition observable to Supervisor 2 but not observable to Supervisor 1 from index $r' = 0$ is the first transition (i.e., $j = 1$), $X_2 = \emptyset$. We set $T_{12} = (0, a, 1)$ and add it to com_{12}^*.

Now we check Supervisor 1's consistency. With $com_{21}^* = \{(0, b, 2), (1, b, 3)\}$, $com_{12}^* = \{(0, a, 1)\}$, we perform the function $\mathcal{N}_{12}(V_{12}, V_{21})$ where $V_{21} = com_{21}^*$ and $V_{12} = com_{12}^*$. First, we create E_1^ϵ by starting from E and replacing the event labels of the transitions that are not in $\Sigma_{1,o}$ and not in V_{21} with ϵ. This is displayed in Figure 5.6.

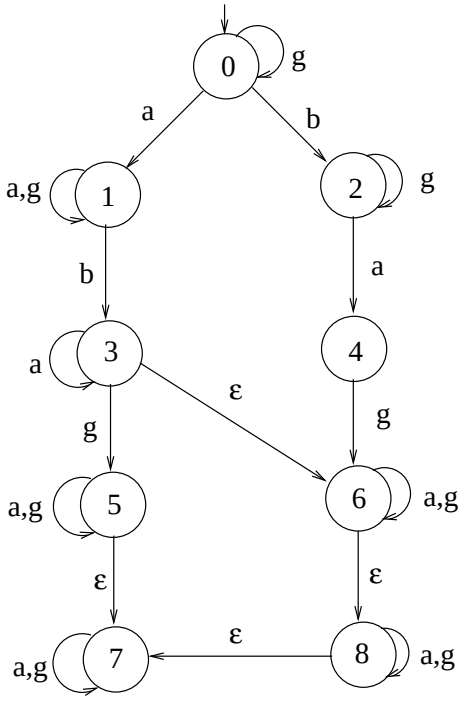

Figure 5.6: NFA of E_1^ϵ, Iteration 2

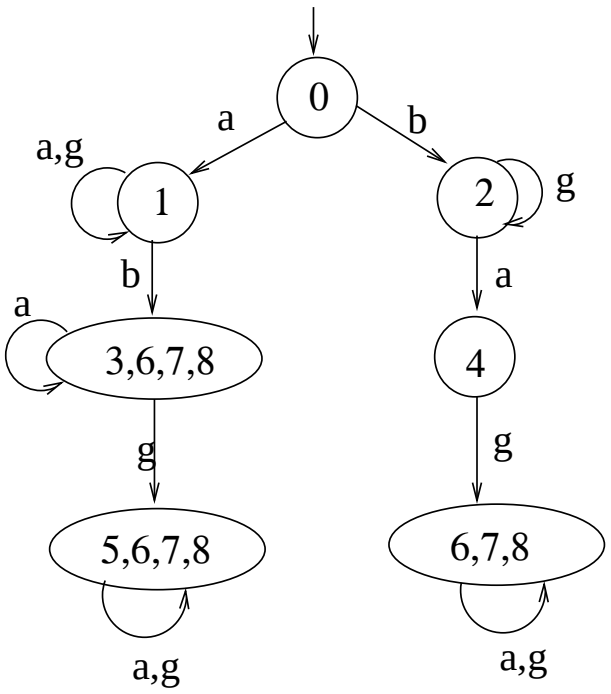

Figure 5.7: DFA of $\tilde{E}_1^{\epsilon}$, Iteration 2

We transform the nondeterministic finite automaton into an equivalent deterministic finite automaton, which is displayed in Figure 5.7.

Then we check consistency for Supervisor 1. Since there is only one state, $\{0\}$, in $\tilde{E}_1^{\epsilon}$ with 0 as an element, and 0 is the only element in that state, we don't need to add any more transitions to com_{12}^*.

With communication pair $com_{12}^* = \{(0, a, 1)\}$ and $com_{21}^* = \{(0, b, 2), (1, b, 3)\}$, we start constructing the modified M-machine $\tilde{M}_2$.

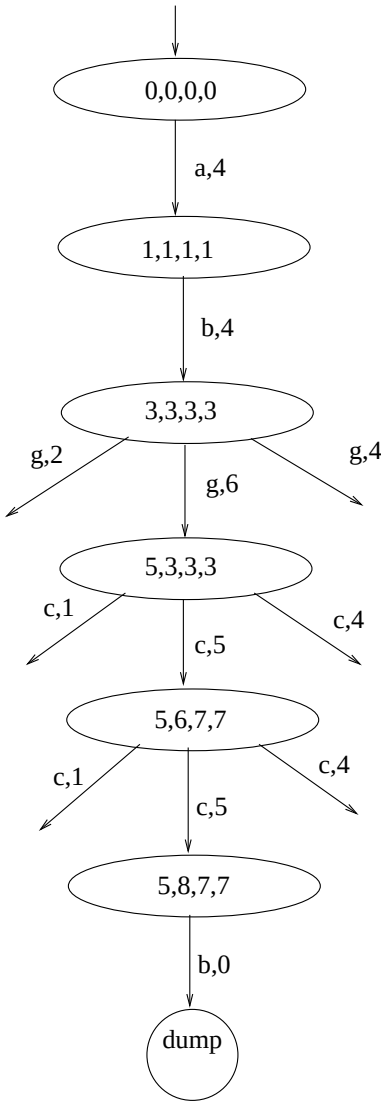

Figure 5.8: A Portion of the Modified M-machine $\tilde{M}_2$

A sequence leading to the dump state of $\tilde{M}_2$ is $s^M\sigma = (a,4)(b,4)(g,6)(c,5)(c,5)(b,0)$ as shown in Figure 5.8. Because $\sigma = b \in \Sigma_{2,c} \setminus \Sigma_{1,c}$, we extract s and s'' using $F_i(i = 2,\ 3)$. We get

$$s = abgcc$$

$$s'' = abcc$$

Following the same reasoning as in earlier iterations, we need to add a transition to com_{12}^* which is $(3, g, 5)$. Since no new transitions are added to com_{21}^*, the NFA of Figure 5.6 and the DFA of Figure 5.7, determined in iteration 2, can be used for E_1^ϵ and $\tilde{E}_1^\epsilon$ (respectively) of iteration 3. After checking consistency, we need to add $(5, g, 5), (6, g, 6)$, $(7, g, 7)$ and $(8, g, 8)$ to keep consistent.

Now the communication pair is

$com_{12}^* = \{(0,a,1),(3,g,5),(5,g,5)(6,g,6),(7,g,7),(8,g,8)\}$ and $com_{21}^* = \{(0,b,2),(1,b,3)\}$.

From Figure 5.9 a sequence leading to the dump state of $\tilde{M}_3$ is

$$s^M\sigma = (b,4)(a,2)(g,2)(c,4)(a,4)(g,4)(b,0).$$

Since $\sigma = b \in \Sigma_{2,c} \setminus \Sigma_{1,c}$, we extract s and s''

$$s = bcag$$

$$s'' = bagcag$$

By function $\mathcal{C}_{12}(s, s'')$, after step 2 finishes, we have $r = r' = 1$. The next transition observable to Supervisor 2 but not observable to Supervisor 1 in sequence s is $(\delta^E(b, q_0^E), c, \delta^E(bc, q_0^E))$, that is, $i = 2$. Since $r + 1 > i - 1$, $X_1 = \emptyset$. For sequence s'', the next transition observable to Supervisor 2 but not observable to Supervisor 1 is $(\delta^E(bag, q_0^E), c, \delta^E(bagc, q_0^E))$, that is, $j = 4$, therefore, X_2 has two elements which are $(2, a, 4)$ (the index is $r'+1 = 2$) and $(4, g, 6)$ (the index is $r'+2 = 3$). Both transitions in X_2 are not in X_1 and both are observable to Supervisor 1 but not observable to Supervisor 2, so we select the transition with higher index which is $T_{12} = (4, g, 6)$ and we add it to com_{12}^*. Also consistency checking does not add more communication.

After the fourth iteration, the communication pair is

$com_{12}^* = \{(0,a,1)(3,g,5)(4,g,6),(5,g,5)(6,g,6),(7,g,7),(8,g,8)\}$ and

$com_{21}^* = \{(0,b,2),(1,b,3)\}$.

We then construct the modified M-machine $\tilde{M}_4$ and there is no dump state in

it. Therefore, with the communication pair $com_{12} = com_{12}^*$ and $com_{21} = com_{21}^*$, this decentralized DES control problem is solved, i.e., two supervisors can be constructed and cooperate to enforce legal behavior from the plant. Moreover, the supervisors are consistent by the nature of the **Main** algorithm. The two supervisors are shown in Figure 5.10 and Figure 5.11. The labels in square blocks are the events labelling transitions that need to be communicated.

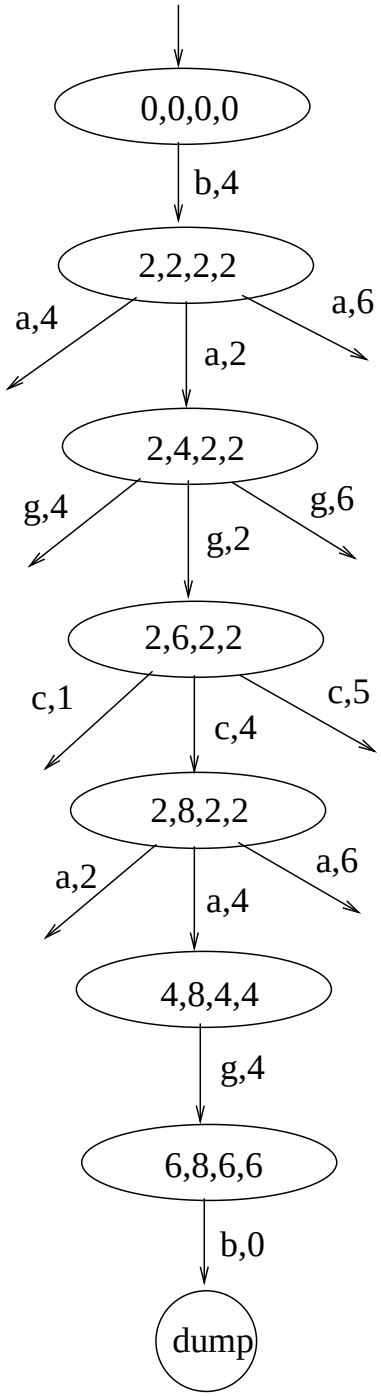

Figure 5.9: A Portion of the Modified M-machine $\tilde{M}_3$

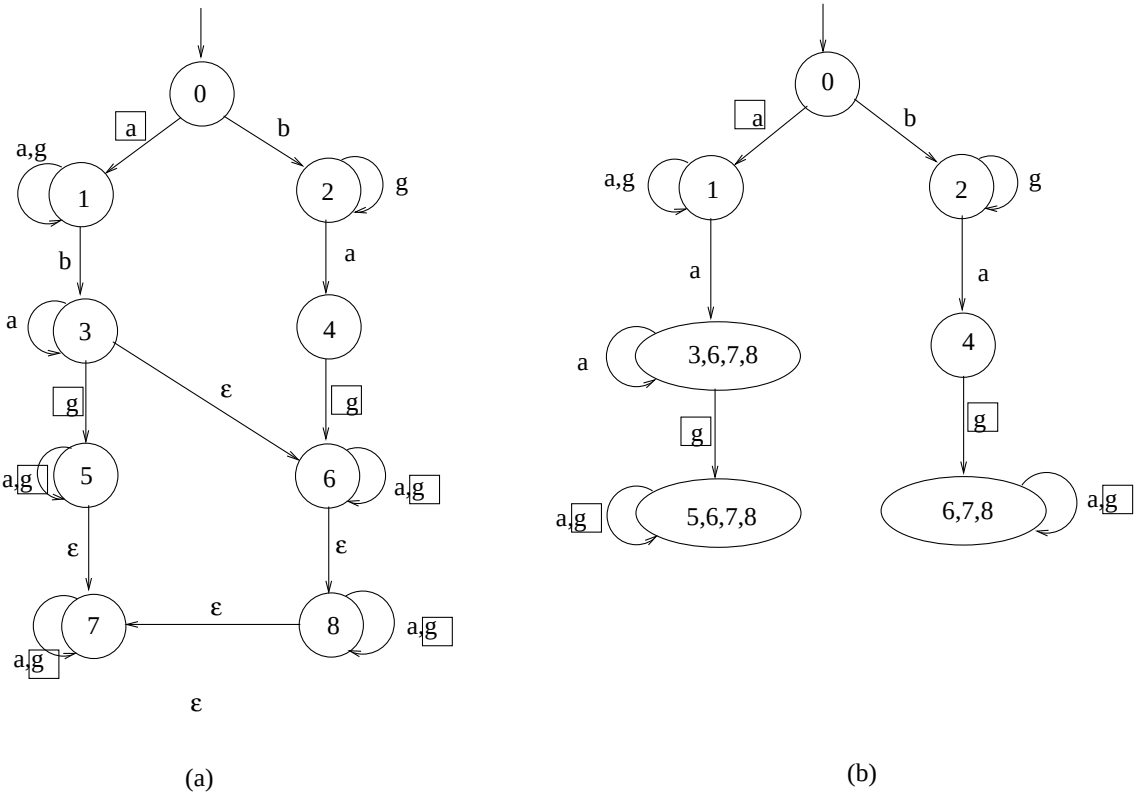

Figure 5.10: Supervisor 1 in Example of Section 5.3 (a): NFA of Supervisor 1 (b): DFA of Supervisor 1

Now, if we look again at the plant and legal automaton of Figure 5.1, we see that the critical thing is for one of the supervisors to know if the plant is at state 4, 6 or 7 since these are the states where an event must be disabled. From Figure 5.10 (b), we see that Supervisor 1 will be able to distinguish state 4 from other states and from Figure 5.11(b), Supervisor 2 will be able to distinguish state 6 and state 7 from other states.

To illustrate how this relates to coobservability, consider sequences

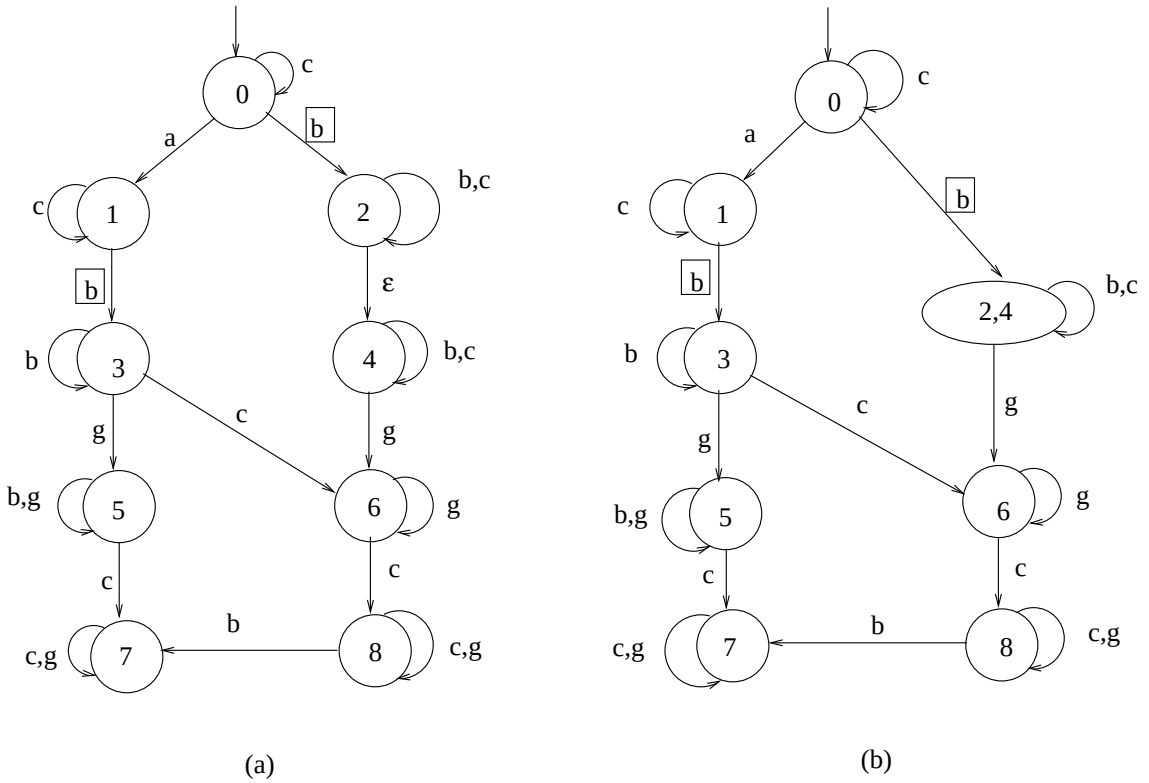

Figure 5.11: Supervisor 2 in Example of Section 5.3 (a): NFA of Supervisor 2 (b): DFA of Supervisor 2

$$s = bcag$$

$$s'' = bagc$$

These sequences violate coobservability since $P_2(s) = P_2(s'')$, $bcag \in L(E)$, $bcagb \in L(G)$ and $bagcb \in L(E)$, but $bcagb \notin L(E)$. However, in Figure 5.11(b), s and s'' lead to two different states, namely 8 and 6, and at state 6 event b is disabled whereas at state 8 event b is enabled.

Chapter 6

Conclusions and Future Work

In this thesis, we claim that when coobservability is not satisfied in decentralized DES control problems, the problems are not always unsolvable. Involving communications between supervisors which makes a property named transition-based coobservability satisfied could solve the problem. We define a new automaton structure, named the modified M-machine, to check transition-based coobservability. With the algorithm described in Chapter 5, we can find a consistent communication pair between two supervisors using the modified M-machine.

Transition-based coobservability can be extended to more than two local supervisors. However, the **Main** algorithm is designed only for cases with two supervisors. Also, work remains to be done to make the algorithm scalable and optimized with respect to computing time and space. As we noted in Chapter 5, the **Main** algorithm does yield a viable communication scheme but it remains as future work to determine if the communication scheme is also minimal. Our work represents an inroad into using the communication of only some occurrences of events to help solve decentralized DES problems. It is our hope that the theory and existence results in this thesis can

be used to develop an algorithm that can be used in practical applications.

Bibliography

[1] C. G. Cassandras and S. Lafortune. *Introduction to Discrete-Event Systems*. Kluwer Academic Publishers, 1999.

[2] R. Cieslak, C. Desclaux, A. S. Fawaz, and P. Varaiya. Supervisory control of discrete-event processes with partial observations. *IEEE Transaction on Automatic Control*, 33(3):249–260, 1988.

[3] J. E. Hopcroft and J. D. Ullman. *Introduction to Automata Theory, Languages, and Computation*. Addison-Wesley, 1979.

[4] F. Lin, K. Rudie, and S. Lafortune. Minimal communication for essential transitions in a distributed discrete-event system. College of Engineering Control Group Report No. CGR-05-02, University of Michigan, Februrary 2005.

[5] F. Lin and W. M. Wonham. On observability of discrete-event systems. *Information Sciences*, 44:173–198, 1988.

[6] P. J. Ramadge and W. M. Wonham. Supervisory control of a class of discrete event processes. *SIAM Journal on Control and Optimization*, 25(1):206–230, 1987.

[7] P. J. Ramadge and W. M. Wonham. Modular supervisory control of discrete-event systems. *Mathematics of Control, Signals, and Systems*, 1:13–30, 1988.

[8] P. J. Ramadge and W. M. Wonham. The control of discrete event systems. *Proceedings of the IEEE*, 77(1):81–98, 1989.

[9] S. L. Ricker. *Knowlege and Communication in Decentralized Discrete-Event Control*. PhD thesis, Department of Computing and Information Science, Queen's University, Kingston, Canada, 1999.

[10] K. Rudie, S. Lafortune, and F. Lin. Minimal communication in a distributed discrete-event system. *IEEE Transactions on Automatic Control*, 48(6):957–975, 2003.

[11] K. Rudie and J. C. Willems. The computational complexity of decentralized discrete-event control problems. IMA Preprint Series 1105, Institute for Mathematics and Its Applications, University of Minnesota, 1993.

[12] K. Rudie and J. C. Willems. The computational complexity of decentralized discrete-event control problems. *IEEE Transactions on Automatic Control*, 40(7):1313–1319, 1995.

[13] K. Rudie and W. M. Wonham. Think globally, act locally: Decentralized supervisory control. *IEEE Tranactions on Automatic Control*, 37(11):1692–1708, 1992.