Discrete Event Dynamic Systems: Theory and Applications, 12, 335–377, 2002

A General Architecture for Decentralized Supervisory Control of Discrete-Event Systems

T.-S. YOO tyoo@eecs.umich.edu
Department of Electrical Engineering and Computer Science, The University of Michigan, 1301 Beal Avenue, Ann Arbor, MI 48109–2122, U.S.A.

STÉPHANE LAFORTUNE stephane@eecs.umich.edu
Department of Electrical Engineering and Computer Science, The University of Michigan, 1301 Beal Avenue, Ann Arbor, MI 48109–2122, U.S.A.

Abstract. We consider a generalized form of the conventional decentralized control architecture for discrete-event systems where the control actions of a set of supervisors can be ''fused'' using both union and intersection of enabled events. Namely, the supervisors agree *a priori* on choosing ''fusion by union'' for certain controllable events and ''fusion by intersection'' for certain other controllable events. We show that under this architecture, a larger class of languages can be achieved than before since a relaxed version of the notion of co-observability appears in the necessary and sufficient conditions for the existence of supervisors. The computational complexity of verifying these new conditions is studied. A method of partitioning the controllable events between ''fusion by union'' and ''fusion by intersection'' is presented. The algebraic properties of co-observability in the context of this architecture are presented. We show that appropriate combinations of fusion rules with corresponding decoupled local decision rules guarantee the safety of the closed-loop behavior with respect to a given specification that is not co-observable. We characterize an ''optimal'' combination of fusion rules among those combinations guaranteeing the safety of the closed-loop behavior. In addition, a simple supervisor synthesis technique generating the infimal prefix-closed controllable and co-observable superlanguage is presented.

Keywords: supervisory control, decentralized architectures, decision fusion, computational complexity, supervisor synthesis

1. Introduction

We consider control problems for discrete-event systems where a set of ''local'' supervisors, each with its own sensing and actuation capabilities, cooperate in order to achieve a given desired controlled behavior. Such decentralized control architectures are of considerable interest as they arise in a large variety of networked systems. Mobile *ad hoc* communication networks, integrated sensor networks, networked control systems, and automated vehicular systems are all examples of networked systems. Networked systems are informationally-decentralized and event-driven dynamic systems where groups of individual ''agents'' (i.e., local supervisors) interact in order to accomplish a common set of objectives.

Our control problems for networked systems are posed in the framework of the theory of supervisory control of discrete-event systems (cf. Ramadge and Wonham, 1989) and Chapter 3 in Cassandras and Lafortune, 1999). In the conventional decentralized control architecture studied in supervisory control (Cieslak et al., 1988; Rudie and Wonham, 1992), the control actions of the local supervisors are fused using intersection of locally enabled events. We will refer to this architecture as the conjunctive architecture.

Most of the results on decentralized supervisory control are based on the conjunctive architecture (Barrett, 1999; Cieslak et al., 1988; Jiang and Kumar, 2000; Kozak and Wonham, 1995; Kumar and Shayman, 1997; 1998; Lin and Wonham, 1988; Ricker, 1999; Rudie and Willems, 1995; Rudie and Wonham, 1992; Takai, 1998; Takai and Kodama, 1994; Willner and Heymann, 1991). A notable exception is the work in Prosser (1996), where decentralized supervision with different fusion rules is considered. In that work, new event fusion rules are introduced and a synthesis technique guaranteeing the safety of the supervised language under various event fusion rules is developed.

In this paper, we go beyond the approach of Prosser (1996) and consider a generalized form of the conjunctive architecture where the control actions of a set of supervisors can be fused using both union and intersection of enabled events. Under this general architecture, the local supervisors decide *a priori* that some controllable events are processed by ‘‘fusion by union’’ (of enabled events) and other controllable events are handled by ‘‘fusion by intersection’’ (of enabled events).

The contributions of this paper are as follows.

1. Necessary and sufficient conditions for the existence of a set of local supervisors that achieve a given legal language are given in Section 3. These conditions characterize the class of languages achievable under the general architecture and introduce a generalized version of the notion of co-observability of Rudie and Wonham (1992).

2. In Section 4, we compare the classes of languages achievable under the conjunctive, disjunctive, and general architectures. (When the decision fusion rule is ‘‘fusion by union’’ (of enabled events), the resulting architecture is called ‘‘disjunctive’’.) The class of languages achievable under the general architecture strictly includes those of the conjunctive and disjunctive architectures.

3. Section 5 presents computational complexity results. We show that the existence conditions of Section 3 can be verified in polynomial time. Moreover, a polynomial time technique to partition the set of controllable events in the general architecture is given.

4. The algebraic properties of co-observability (as defined in Section 3) are presented in Section 6. These properties show that the supremal and infimal co-observable elements of a class of languages need not exist, in general.

5. A simple decentralized supervisor synthesis technique decoupling the synthesis of local supervisors is developed under the general architecture. Because of the intentional separation of the design of the local decision rules, this technique circumvents the difficulties caused by the dependency of local decision rules in the design of decentralized supervisors.

6. Equipped with the above synthesis technique, we present in Section 7.1 a rule for partitioning the set of controllable events (between ‘‘fusion by union’’ and ‘‘fusion by

intersection'') that guarantees the safety of the closed-loop behavior with respect to a given specification that is not co-observable. We also characterize an ''optimal'' partitioning rule of the set of controllable events among the partitions guaranteeing the safety of the closed-loop behavior.

7. Several properties of the closed-loop behavior generated by the above synthesis technique are presented in Section 7.2.

8. We also present a simple supervisor synthesis procedure that results in the infimal prefix-closed controllable and co-observable (in the sense of Rudie and Wonham, 1992) superlanguage in Section 7.2. The local supervisors are separately synthesized in a direct manner that avoids the explicit computation of the infimal prefix-closed controllable and co-observable superlanguage.

General knowledge of supervisory control and its most common notations is assumed. For introductory material, the reader is directed to Chapter 3 of Cassandras and Lafortune (1999).

2. Problem Formulation

2.1. The Conjunctive Architecture

The problem of decentralized supervision is formalized as follows. Each local decision site (supervisor) has its own sensors ($\Sigma_{o,i}$) and controllers ($\Sigma_{c,i}$). Collectively, the sites can observe $\Sigma_o = \Sigma_{o,1} \cup \ldots \cup \Sigma_{o,n}$ and control $\Sigma_c = \Sigma_{c,1} \cup \ldots \cup \Sigma_{c,n}$. We denote by $\Sigma_{uo} = \Sigma \setminus \Sigma_o$ and $\Sigma_{uc} = \Sigma \setminus \Sigma_c$, the unobservable and uncontrollable event sets, respectively. *A priori* information available to each local site includes the uncontrolled behavior, the desired behavior, and the decision fusion rule to form a global decision. The conventional decentralized architecture shown here in Figure 1 employs the conjunctive fusion rule for enabled events, thus requiring unanimous enabling of events in local

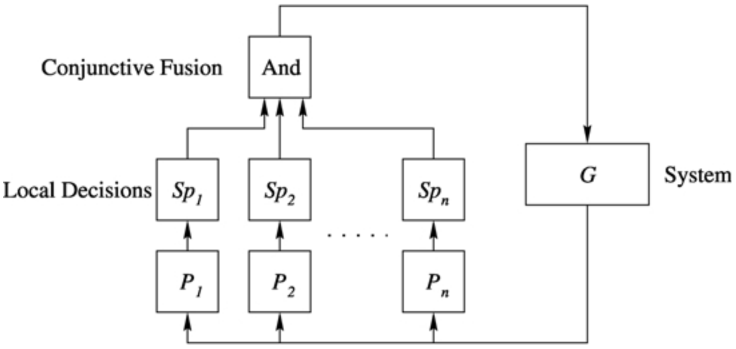


Figure 1. The conjunctive architecture.

decisions for global enablement; we also call this rule ‘‘fusion by intersection’’ (of enabled events). Formally, a local decision rule is a function

$$S_{P_i} : P_i(\Sigma^*) \to \Gamma := \{\gamma \in 2^{\Sigma} : \Sigma_{uc} \subseteq \gamma\}$$

where P_i is the usual projection mapping : $P_i : \Sigma \to \Sigma_{o,i}$. For the conjunctive architecture, a local decision rule enables $\Sigma_c \setminus \Sigma_{c,i}$ by default:

$$S_{P_i} : P_i(\Sigma^*) \to \Gamma := \{\gamma \in 2^{\Sigma} : \Sigma \setminus \Sigma_{c,i} \subseteq \gamma\}$$

The conjunctive supervisor, denoted by $\wedge_i S_{P_i}$, is defined as follows:

$$\wedge_i S_{P_i}(s) := \cap_{i=1}^{n} S_{P_i}(P_i(s))$$

The prefix-closed language generated by the conjunctive supervisor can be expressed as follows:

1. $\varepsilon \in \mathscr{L}(\wedge_i S_{P_i}/G)$;

2. $[s \in \mathscr{L}(\wedge_i S_{P_i}/G)] \wedge [s\sigma \in \mathscr{L}(G)] \wedge [\forall i, \sigma \in S_{P_i}(P_i(s))] \Leftrightarrow s\sigma \in \mathscr{L}(\wedge_i S_{P_i}/G)$

Moreover, the marked language is defined as follows:

$$\mathscr{L}_m(\wedge_i S_{P_i}/G) = \mathscr{L}(\wedge_i S_{P_i}/G) \cap \mathscr{L}_m(G)$$

In prior work on the conjunctive architecture, the default control action for a supervisor under insufficient information is to ‘‘enable’’ an event; this is termed the ‘‘pass the buck’’ policy in Rudie and Wonham (1992). We will refer to this default policy as the ‘‘permissive’’ local decision rule. The permissive local decision rule over Σ_c is formally described as follows: For all $i \in \{1, \ldots, n\}$,

$$S_{P_i}(P_i(s)) = \{\sigma \in \Sigma_{c,i} : P_i^{-1} P_i(s)\sigma \cap \overline{K} \neq \emptyset\} \cup \Sigma \setminus \Sigma_{c,i} \tag{1}$$

where K is the given desired language.

In the conjunctive architecture, co-observability is the key property for the existence of a set of local supervisors controlling a given desired language. This property was introduced in Cieslak et al. (1988) and Rudie and Wonham (1992). Since we shall present a relaxed version of ‘‘co-observability’’, we rename the conventional notion of co-observability as C&P co-observability. C refers to the ‘‘conjunctive’’ fusion rule for controllable events, while P refers to the ‘‘permissive’’ local decision rule. For two sites, the definition of C&P co-observability is as follows.

DEFINITION 1 *A language $K \subseteq M = \overline{M}$ is said to be C&P co-observable w.r.t. M, $\Sigma_{o,1}$, $\Sigma_{c,1}$, $\Sigma_{o,2}$, $\Sigma_{c,2}$, if*

$$
\begin{array}{ll}
 & (\forall s, s', s'' \in \overline{K})[[P_1(s) = P_1(s')] \wedge [P_2(s) = P_2(s'')]] \Rightarrow \\
\textit{Conjunct 1}: & [\sigma \in \Sigma_{c,1} \cap \Sigma_{c,2}] \wedge [s\sigma \in M] \wedge [s'\sigma \in \overline{K}] \wedge [s''\sigma \in \overline{K}] \\
 & \Rightarrow s\sigma \in \overline{K} \\
\textit{Conjunct 2}: & [\sigma \in \Sigma_{c,1} \setminus \Sigma_{c,2}] \wedge [s\sigma \in M] \wedge [s'\sigma \in \overline{K}] \Rightarrow s\sigma \in \overline{K} \\
\textit{Conjunct 3}: & [\sigma \in \Sigma_{c,2} \setminus \Sigma_{c,1}] \wedge [s\sigma \in M] \wedge [s''\sigma \in \overline{K}] \Rightarrow s\sigma \in \overline{K}
\end{array}
$$

We also recall an equivalent definition that is presented in Barrett (1999) and Cassandras and Lafortune (1999) for further arguments. We state this definition for n local sites.

DEFINITION 2 *A language $K \subseteq M = \overline{M}$ is said to be C&P co-observable w.r.t. M, $\Sigma_{o,1}$, $\Sigma_{c,1}, \ldots, \Sigma_{o,n}$, $\Sigma_{c,n}$, if $\forall s \in \overline{K}$ and $\forall \sigma \in \Sigma_c = \cup_{i=1}^{n} \Sigma_{c,i}$ s.t. $s\sigma \in M \setminus \overline{K}$,*

$$\big(\exists i \in \{1, \ldots, n\}\big)[[P_i^{-1} P_i(s)\sigma \cap \overline{K} = \emptyset] \wedge [\sigma \in \Sigma_{c,i}]]$$

2.2. *The Disjunctive Architecture*

Consider an architecture where ''fusion by union'' (of enabled events) is employed, as depicted in Figure 2; let us call the resulting (global) supervisor a ''disjunctive supervisor''. For the disjunctive architecture, a local decision rule disables by default $\Sigma_c \setminus \Sigma_{c,i}$ that is controllable by other local supervisors:

$$S_{P_i} : P_i(\Sigma^*) \rightarrow \Gamma := \{\gamma \in 2^{\Sigma} : (\Sigma_c \setminus \Sigma_{c,i}) \cap \gamma = \emptyset, \Sigma_{uc} \subseteq \gamma\}$$

The disjunctive supervisor $\vee_i S_{P_i}$ is defined as follows:

$$\vee_i S_{P_i}(s) := \cup_{i=1}^{n} S_{P_i}(P_i(s))$$

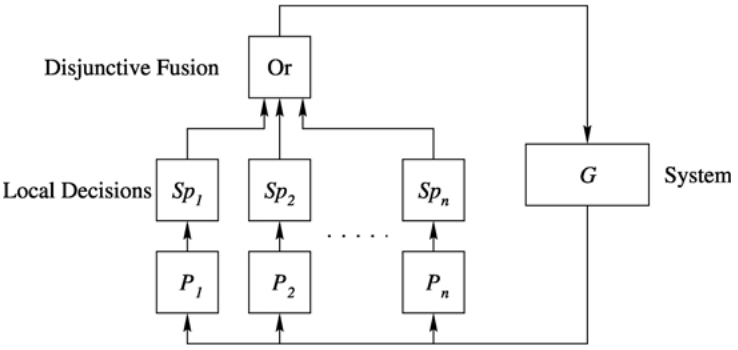


Figure 2. The disjunctive architecture.

The prefix-closed language generated by the disjunctive fusion is described recursively as follows:

1. $\varepsilon \in \mathscr{L}(\vee_i S_{P_i}/G)$;

2. $[s \in \mathscr{L}(\vee_i S_{P_i}/G)] \wedge [s\sigma \in \mathscr{L}(G)] \wedge [\exists i, \sigma \in S_{P_i}(P_i(s))] \Leftrightarrow s\sigma \in \mathscr{L}(\vee_i S_{P_i}/G)$

Analogously, the marked language is $\mathscr{L}_m(\vee_i S_{P_i}/G) = \mathscr{L}(\vee_i S_{P_i}/G) \cap \mathscr{L}_m(G)$.

It is easy to see that the definitions of $\mathscr{L}(\wedge_i S_{P_i}/G)$ and $\mathscr{L}(\vee_i S_{P_i}/G)$ depend on both the local decision rules and the fusion policy. For the conjunctive architecture, the default for controllable events is enablement and local supervisors directly disable the locally controllable events. The disablement default for controllable events should be employed for the disjunctive architecture as local supervisors directly enable the locally controllable events.

2.3. The General Architecture

The general architecture investigated in this paper is now described. The set of controllable events, Σ_c, is partitioned into $\Sigma_{c,e}$ and $\Sigma_{c,d}$:

$$\Sigma_c = \Sigma_{c,e} \dot{\cup} \Sigma_{c,d}$$

$\Sigma_{c,e}$ is the set of controllable events for which the default setting is enablement while $\Sigma_{c,d}$ is the set of controllable events for which the default setting is disablement. The local decisions over $\Sigma_{c,e}$ are processed by the conjunctive fusion rule while the local decisions over $\Sigma_{c,d}$ are processed by the disjunctive fusion rule. Figure 3 is a conceptual diagram of the general architecture. For the general architecture, a local decision rule disables by

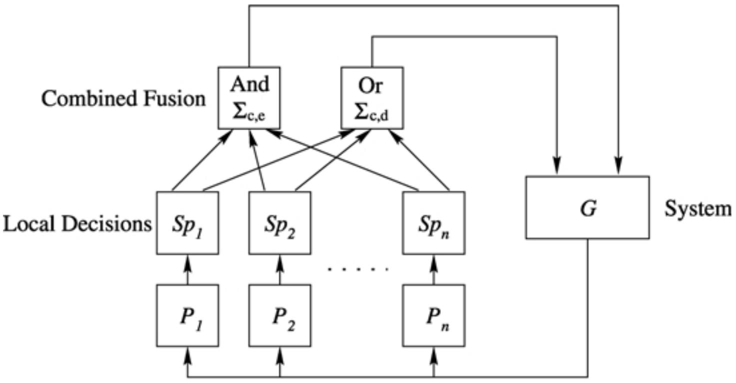


Figure 3. The general architecture.

default $\Sigma_c \setminus \Sigma_{c,i}$ in $\Sigma_{c,d}$ that is controllable by other local supervisors and enables by default $\Sigma_c \setminus \Sigma_{c,i}$ in $\Sigma_{c,e}$:

$$S_{P_i} : P_i(\Sigma^*) \rightarrow \Gamma := \{\gamma \in 2^{\Sigma} : (\Sigma_{c,d} \setminus \Sigma_{c,i}) \cap \gamma = \emptyset, \Sigma_{c,e} \setminus \Sigma_{c,i} \subseteq \gamma, \Sigma_{uc} \subseteq \gamma\}$$

Let us define a generalized decentralized supervisor (generalized supervisor for the sake of brevity) with combined fusion rule, denoted by S_{gdec}, as

$$S_{gdec}(s) := P_{\Sigma_{c,e}}[\wedge_i S_{P_i}(s)] \cup P_{\Sigma_{c,d}}[\vee_i S_{P_i}(s)] \cup \Sigma_{uc}$$

where $P_{\Sigma_{c,e}}$ and $P_{\Sigma_{c,d}}$ are projection mappings: $P_{\Sigma_{c,e}} : \Sigma \rightarrow \Sigma_{c,e}$ and $P_{\Sigma_{c,d}} : \Sigma \rightarrow \Sigma_{c,d}$. The prefix-closed language $\mathscr{L}(S_{gdec}/G)$ generated in the context of the general architecture is defined in the usual manner:

1. $\varepsilon \in \mathscr{L}(S_{gdec}/G)$;

2. $[s \in \mathscr{L}(S_{gdec}/G)] \wedge [s\sigma \in \mathscr{L}(G)] \wedge [\sigma \in S_{gdec}(s)] \Leftrightarrow s\sigma \in \mathscr{L}(S_{gdec}/G)$

The marked language is defined as usual: $\mathscr{L}_m(S_{gdec}/G) = \mathscr{L}(S_{gdec}/G) \cap \mathscr{L}_m(G)$.

It should be clear that if the sets $\Sigma_{c,i}$ are mutually disjoint, then the three architectures (general, disjunctive, conjunctive) are one and the same. This is because each controllable event is controlled by only one supervisor, and therefore the fusion rule is of no consequence: the controllable event is enabled iff the corresponding supervisor enables it. The interest of the general architecture arises when there are controllable events that can be controlled by more than one supervisors. We therefore assume in the sequel the presence of such controllable events.

2.4. Problem Statement

The first fundamental question to consider for the previously-described general architecture is that of the existence of a generalized supervisor achieving a given legal language. Unlike the conjunctive and disjunctive architectures, the general architecture requires partitioning Σ_c. This requirement is reflected in the existence problem (**P**):

(**P**) Given automaton G, modeling the uncontrolled behavior, automaton H, representing the desired behavior, and local supervisors equipped with $\Sigma_{c,i}$, $\Sigma_{o,i}$, $i \in \{1, \ldots, n\}$, respectively, find necessary and sufficient conditions for the existence of a partition of $\Sigma_c = \Sigma_{c,e} \dot{\cup} \Sigma_{c,d}$ and of a nonblocking generalized supervisor controlling $\mathscr{L}_m(H)$ under the partition, that is,

$$\mathscr{L}_m(S_{gdec}/G) = \mathscr{L}_m(H) \text{ and } \mathscr{L}(S_{gdec}/G) = \overline{\mathscr{L}_m(H)}$$

3. Existence Result for Fixed Partition of Σ_c

In this section, we fix the partition of Σ_c and investigate the problem of the existence of a generalized supervisor. This partly answers the question posed in (**P**). In Section 5, we explore the issue of partitioning Σ_c into $\Sigma_{c,e}$ and $\Sigma_{c,d}$ and complete the answer for (**P**). Unlike the conjunctive architecture, the conjunctive and disjunctive fusion rules coexist in the general architecture. Let us focus on the disjunctive fusion rule at first. We define D&A co-observability for the disjunctive architecture, the analogue of C&P co-observability for the conjunctive architecture.

DEFINITION 3 *A language* $K \subseteq M = \overline{M}$ *is said to be D&A co-observable w.r.t. M,* $\Sigma_{o,1}$, $\Sigma_{c,1}, \ldots, \Sigma_{o,n}, \Sigma_{c,n}$, *if* $\forall s \in \overline{K}$ *and* $\forall \sigma \in \Sigma_c = \cup_{i=1}^{n} \Sigma_{c,i}$ *s.t.* $s\sigma \in \overline{K}$,

$$\big(\exists i \in \{1, \ldots, n\}\big)[[(P_i^{-1}P_i(s) \cap \overline{K})\sigma \cap M \subseteq \overline{K}] \wedge [\sigma \in \Sigma_{c,i}]]$$

The term ''D&A'' stands for disjunctive and antipermissive. The reason for this term is that this property is for the disjunctive architecture and the antipermissive decision strategy should be applied at local supervisors. We say that the decision strategy at a local supervisor is ''antipermissive'' if the default control action for controllable events is disablement. The antipermissive local decision rule over Σ_c is formally described as follows: For all $i \in \{1, \ldots, n\}$,

$$S_{P_i}(P_i(s)) = \{\sigma \in \Sigma_{c,i} : (P_i^{-1}P_i(s) \cap \overline{K})\sigma \cap M \subseteq \overline{K}\} \cup \Sigma_{uc} \tag{2}$$

The intuitive meaning of the antipermissive rule is to permit the occurrence of a controllable continuation after observable trace $P_i(s)$ only if the local supervisor has sufficient information to determine with certainty, namely based on its ''estimate'' $P_i^{-1}P_i(s) \cap \overline{K}$, that enabling the controllable event will not cause any violation of legality. Before the existence result is stated, we present an equivalent statement of D&A co-observability for the convenience of further arguments. We restrict the attention to two local supervisors without loss of generality.

PROPOSITION 1 *A language* $K \subseteq M = \overline{M}$ *is D&A co-observable w.r.t. M,* $\Sigma_{o,1}, \Sigma_{c,1}, \Sigma_{o,2}, \Sigma_{c,2}$, *iff*

$$(\forall s, s', s'' \in \overline{K})[[P_1(s) = P_1(s')] \wedge [P_2(s) = P_2(s'')]] \Rightarrow$$

$$\begin{aligned}
&\textit{Conjunct 1}: && [\sigma \in \Sigma_{c,1} \cap \Sigma_{c,2}] \wedge [s\sigma \in M] \wedge [s'\sigma \in M \setminus \overline{K}] \wedge [s'\sigma \in M \setminus \overline{K}] \\
& && \Rightarrow s\sigma \in M \setminus \overline{K} \\
&\textit{Conjunct 2}: && [\sigma \in \Sigma_{c,1} \setminus \Sigma_{c,2}] \wedge [s\sigma \in M] \wedge [s'\sigma \in M \setminus \overline{K}] \Rightarrow s\sigma \in M \setminus \overline{K} \\
&\textit{Conjunct 3}: && [\sigma \in \Sigma_{c,2} \setminus \Sigma_{c,1}] \wedge [s\sigma \in M] \wedge [s''\sigma \in M \setminus \overline{K}] \Rightarrow s\sigma \in M \setminus \overline{K}
\end{aligned}$$

Proof: The proof can be found in Yoo (2002). ■

The centralized architecture is a special case of the decentralized architectures. Under the centralized architecture, the two notions of co-observabilities can be related as follows:

PROPOSITION 2 *C&P co-observability is equivalent to D&A co-observability under the centralized architecture.*

Proof: We restrict the attention to two local supervisors without loss of generality. Since we are considering the centralized architecture, C&P co-observability is reduced to observability. That is, for all $s, s' \in \overline{K}$ such that $P(s) = P(s')$,

$$\begin{aligned}
&[\sigma \in \Sigma_c] \wedge [s'\sigma \in M] \wedge [s\sigma \in \overline{K}] \Rightarrow [s'\sigma \in \overline{K}] \\
\Leftrightarrow\ &[\sigma \notin \Sigma_c] \vee [s\sigma \notin \overline{K}] \vee [s'\sigma \notin M] \vee [s'\sigma \in \overline{K}] \\
\Leftrightarrow\ &[\sigma \notin \Sigma_c] \vee [s\sigma \notin \overline{K}] \vee [s'\sigma \notin M \setminus \overline{K}] \\
\Leftrightarrow\ &[\sigma \notin \Sigma_c] \vee [s\sigma \notin M] \vee [s'\sigma \notin M \setminus \overline{K}] \vee [s\sigma \in M \setminus \overline{K}] \\
\Leftrightarrow\ &[\sigma \in \Sigma_c] \wedge [s\sigma \in M] \wedge [s'\sigma \in M \setminus \overline{K}] \Rightarrow [s\sigma \in M \setminus \overline{K}]
\end{aligned}$$

which is the statement of D&A co-observability in the centralized architecture. ■

Let us define the following sets of events: For $i \in \{1, \ldots, n\}$,

$$\Sigma_{c,e,i} := \Sigma_{c,i} \cap \Sigma_{c,e} \tag{3}$$

$$\Sigma_{c,d,i} := \Sigma_{c,i} \cap \Sigma_{c,d} \tag{4}$$

$\Sigma_{c,e,i}$ is the set of locally controllable events whose default setting is enablement while $\Sigma_{c,d,i}$ is the set of locally controllable events whose default setting is disablement. We generalize C&P and D&A co-observability to embrace the partition of Σ_c; we call this generalized notion ‘‘co-observability’’ for the sake of simplicity.

DEFINITION 4 *A language $K \subseteq M = \overline{M}$ is said to be co-observable w.r.t. M, $\Sigma_{o,1}$, $\Sigma_{c,d,1}$, $\Sigma_{c,e,1}$, $\Sigma_{o,2}$, $\Sigma_{c,d,2}$, $\Sigma_{c,e,2}$, $\ldots$, $\Sigma_{o,n}$, $\Sigma_{c,d,n}$, $\Sigma_{c,e,n}$, if*

1. *K is C&P co-observable w.r.t. $M, \Sigma_{o,1}, \Sigma_{c,e,1}, \ldots, \Sigma_{o,n}, \Sigma_{c,e,n}$,*

2. *K is D&A co-observable w.r.t. $M, \Sigma_{o,1}, \Sigma_{c,d,1}, \ldots, \Sigma_{o,n}, \Sigma_{c,d,n}$.*

With this generalized notion of co-observability, the existence result of the general architecture can be presented.

THEOREM 1 *Consider a language $K \subseteq \mathscr{L}_m(G)$ where $K \neq \emptyset$ and consider a fixed partition of Σ_c such that $\Sigma_c = \Sigma_{c,d} \dot{\cup} \Sigma_{c,e}$. There exists a nonblocking generalized supervisor S_{gdec} such that $\mathscr{L}_m(S_{gdec}/G) = K$ and $\mathscr{L}(S_{gdec}/G) = \overline{K}$ iff the three following conditions hold:*

1. *K is controllable w.r.t. $\mathscr{L}(G)$ and Σ_{uc}.*

2. *K is co-observable w.r.t. $\mathscr{L}(G), \Sigma_{o,1}, \Sigma_{c,d,1}, \Sigma_{c,e,1}, \ldots, \Sigma_{o,n}, \Sigma_{c,d,n}, \Sigma_{c,e,n}$.*

3. *K is $\mathscr{L}_m(G)$-closed.*

Proof: ($\Rightarrow$) Suppose that there exists a generalized supervisor such that $\mathscr{L}_m(S_{gdec}/G) = K$ and $\mathscr{L}(S_{gdec}/G) = \overline{K}$.

($\mathscr{L}_m(G)$-closure): Then, by the definition of $\mathscr{L}_m(S_{gd}/G)$,

$$\begin{aligned} &\mathscr{L}_m(S_{gdec}/G) = \mathscr{L}(S_{gdec}/G) \cap \mathscr{L}_m(G) \\ \Rightarrow &K = \overline{K} \cap \mathscr{L}_m(G) \end{aligned}$$

which is the $\mathscr{L}_m(G)$-closed condition.

(Controllability): Let $s \in \overline{K}$, $\sigma \in \Sigma_{uc}$, and $s\sigma \in \mathscr{L}(G)$. Then, $\sigma \in S_{gdec}(s)$ by the definition of $S_{gdec}(s)$. Now, by the definition of $\mathscr{L}(S_{gdec}/G)$, we have that

$$\begin{aligned} &[s \in \overline{K} = \mathscr{L}(S_{gdec}/G)] \wedge [s\sigma \in \mathscr{L}(G)] \wedge [\sigma \in S_{gdec}(s)] \\ \Rightarrow &[s\sigma \in \mathscr{L}(S_{gdec}/G) = \overline{K}] \end{aligned}$$

or, in terms of languages,

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

which is controllability condition.

(Co-observability): Assume that K is not co-observable w.r.t. $\mathscr{L}(G)$, $\Sigma_{o,1}, \Sigma_{c,d,1}, \Sigma_{c,e,1}, \ldots, \Sigma_{o,n}, \Sigma_{c,d,n}, \Sigma_{c,e,n}$. This implies that K is not C&P co-observable w.r.t. $\mathscr{L}(G), \Sigma_{o,1}, \Sigma_{c,e,1}, \ldots, \Sigma_{o,n}, \Sigma_{c,e,n}$, or K is not D&A co-observable w.r.t. $\mathscr{L}(G), \Sigma_{o,1}, \Sigma_{c,d,1}, \ldots, \Sigma_{o,n}, \Sigma_{c,d,n}$. Assume that K is not C&P co-observable w.r.t. $\mathscr{L}(G), \Sigma_{o,1}, \Sigma_{c,e,1}, \ldots, \Sigma_{o,n}, \Sigma_{c,e,n}$. This implies that there exist $s \in \overline{K}$ and $\sigma \in \Sigma_{c,e}$ such that $s\sigma \in \mathscr{L}(G) \setminus \overline{K}$ and, for all $i \in \{1, \ldots, n\}$,

$$[P_i^{-1}P_i(s)\sigma \cap \overline{K} \neq \emptyset] \vee [\sigma \notin \Sigma_{c,e,i}] \tag{5}$$

For the sake of contradiction, let us assume that $\sigma \notin S_{gdec}(s)$. Since $\sigma \in \Sigma_{c,e}$, we get $\sigma \notin P_{\Sigma_{c,e}}[\wedge_i S_{P_i}(s)]$. This implies that there exists $j \in \{1, \ldots, n\}$ such that $\sigma \notin S_{P_j}(P_j(s))$ and $\sigma \in \Sigma_{c,e,j}$. From (5), we know that there exists $s' \in \overline{K}$ such that

$$[P_j(s') = P_j(s)] \wedge [s'\sigma \in \overline{K}]$$

which implies that $\sigma \in S_{gdec}(s')$ and $\sigma \in \Sigma_{c,e,j}$. This implies that $\sigma \in \Sigma_{c,e}$ and

$\sigma \in P_{\Sigma_{c,e}}[\wedge_i S_{P_i}(s')]$. Then we get $\sigma \in S_{P_j}(P_j(s')) = S_{P_j}(P_j(s))$, which is a contradiction. Therefore, we have

$$[\sigma \in S_{gdec}(s)] \wedge [s \in \overline{K}] \wedge [s\sigma \in \mathscr{L}(G) \setminus \overline{K}]$$

This contradicts the hypothesis that there exists a generalized supervisor such that $\mathscr{L}_m(S_{gdec}/G) = K$ and $\mathscr{L}(S_{gdec}/G) = \overline{K}$.

Now assume that K is not D&A co-observable w.r.t. $\mathscr{L}(G), \Sigma_{o,1}, \Sigma_{c,d,1}, \ldots, \Sigma_{o,n}, \Sigma_{c,d,n}$. This implies that there exist $s \in \overline{K}$ and $\sigma \in \Sigma_{c,d}$ such that $s\sigma \in \overline{K}$ and, for all $i \in \{1, \ldots, n\}$,

$$[(P_i^{-1}P_i(s) \cap \overline{K})\sigma \cap \mathscr{L}(G) \not\subseteq \overline{K}] \vee [\sigma \notin \Sigma_{c,d,i}] \tag{6}$$

For the sake of contradiction, let us assume that $\sigma \in S_{gdec}(s)$. Since $\sigma \in \Sigma_{c,d}$, we get $\sigma \in P_{\Sigma_{c,d}}[\vee_i S_{P_i}(s)]$. This implies that there exists $j \in \{1, \ldots, n\}$ such that $\sigma \in S_{P_j}(P_j(s))$ and $\sigma \in \Sigma_{c,d,j}$. From (6), we know that there exists $s' \in \overline{K}$ such that

$$[P_j(s') = P_j(s)] \wedge [s'\sigma \in \mathscr{L}(G) \setminus \overline{K}]$$

which implies that $\sigma \notin S_{gdec}(s')$ and $\sigma \notin \Sigma_{c,d,j}$. This implies that $\sigma \notin \Sigma_{c,d}$ and $\sigma \notin P_{\Sigma_{c,d}}[\vee_i S_{P_i}(s')]$. Then we get $\sigma \notin, S_{P_j}(P_j(s')) = S_{P_j}(P_j(s))$, which is a contradiction. Therefore, we have

$$[\sigma \notin S_{gdec}(s)] \wedge [s \in \overline{K}] \wedge [s\sigma \in \overline{K}]$$

This contradicts the hypothesis that there exists a generalized supervisor such that $\mathscr{L}_m(S_{gdec}/G) = K$ and $\mathscr{L}(S_{gdec}/G) = \overline{K}$.

($\Leftarrow$) For $s \in \Sigma^*$, define the local decision rules as follows: For all $i \in \{1, \ldots, n\}$,

$$\begin{aligned} S_{P_i}(P_i(s)) = &\{\sigma \in \Sigma_{c,d,i} : (P_i^{-1}P_i(s) \cap \overline{K})\sigma \cap \mathscr{L}(G) \subseteq \overline{K}\} \\ &\cup \{\sigma \in \Sigma_{c,e,i} : (P_i^{-1}P_i(s))\sigma \cap \overline{K} \neq \emptyset\} \\ &\cup \Sigma_{c,e} \setminus \Sigma_{c,e,i} \cup \Sigma_{uc} \end{aligned} \tag{7}$$

We now prove that, with these S_{P_i}, $\mathscr{L}(S_{gdec}/G) = \overline{K}$. Then, the $\mathscr{L}_m(G)$-closure condition will imply that $\mathscr{L}_m(S_{gdec}/G) = K$.

The proof is done by induction on the length of the traces in the two languages $\overline{K}$ and $\mathscr{L}(S_{gdec}/G)$.

(Base of induction): The base case is for $\varepsilon \in \Sigma^*$. By definition of $\mathscr{L}(S_{gdec}/G)$, $\varepsilon \in \mathscr{L}(S_{gdec}/G)$. Since $K \neq \emptyset$ by the assumption, $\varepsilon \in \overline{K}$. Thus the base case holds.

(Induction hypothesis): Assume for all traces such that $|s| \leq n$,

$$s \in \mathscr{L}(S_{gdec}/G) \quad \text{iff} \quad s \in \overline{K}$$

(Induction step): We now prove that for all $\sigma \in \Sigma$,

$$s\sigma \in \mathscr{L}(S_{gdec}/G) \quad \text{iff} \quad s\sigma \in \overline{K} \quad \text{where} \quad |s| = n$$

Let $s\sigma \in \mathscr{L}(S_{gdec}/G)$. By the definition of $\mathscr{L}(S_{gdec}/G)$, this implies that

$$[s \in \mathscr{L}(S_{gdec}/G)] \wedge [\sigma \in S_{gdec}(s)] \wedge [s\sigma \in \mathscr{L}(G)]$$

which in turn implies that

$$[s \in \overline{K}] \wedge [\sigma \in S_{gdec}(s)] \wedge [s\sigma \in \mathscr{L}(G)] \tag{8}$$

using the induction hypothesis. We examine the three following cases.

Case 1: ($\sigma \in \Sigma_{uc}$) Controllability immediately yields $s\sigma \in \overline{K}$ from (8).

Case 2: ($\sigma \in \Sigma_{c,e}$) Assume $s\sigma \notin \overline{K}$. This implies that $s\sigma \in \mathscr{L}(G) \setminus \overline{K}$. By C&P co-observability of K w.r.t. $\mathscr{L}(G)$, $\Sigma_{o,1}$, $\Sigma_{c,e,1}, \ldots, \Sigma_{o,n}$, $\Sigma_{c,e,n}$, there exists $i \in \{1, \ldots, n\}$ such that

$$[(P_i^{-1}P_i(s))\sigma \cap \overline{K} = \emptyset] \wedge [\sigma \in \Sigma_{c,e,i}]$$

By the local decision rule (7), we get $\sigma \notin S_{P_i}(P_i(s))$. This implies that $\sigma \notin \wedge S_{P_i}(s)$. Hence, we get $\sigma \notin S_{gdec}(s)$. It is a contradiction.

Case 3: ($\sigma \in \Sigma_{c,d}$) Then, (8) becomes

$$[s \in \overline{K}] \wedge [\sigma \in \vee S_{P_i}(s)] \wedge [s\sigma \in \mathscr{L}(G)] \tag{9}$$

Using (7) and (9), we get

$$\begin{aligned} &[s \in \overline{K}] \wedge [(\exists i \in \{1, \ldots, n\})[(P_i^{-1}P_i(s) \cap \overline{K})\sigma \cap \mathscr{L}(G) \subseteq \overline{K}] \wedge [s\sigma \in \mathscr{L}(G)] \\ &\Rightarrow s\sigma \in \overline{K} \end{aligned}$$

This completes the proof that $s\sigma \in \overline{K}$.

Let $s\sigma \in \overline{K}$. Then, $s\sigma \in \mathscr{L}(G)$ since $\overline{K} \subseteq \overline{\mathscr{L}_m(G)} \subseteq \mathscr{L}(G)$ by the assumption. Similarly, we examine three cases.

Case 1: ($\sigma \in \Sigma_{uc}$) $\sigma \in S_{gdec}(s)$ by definition. Then it immediately yields $s\sigma \in \mathscr{L}(S_{gdec}/G)$.

Case 2: ($\sigma \in \Sigma_{c,e}$) Assume that $s\sigma \notin \mathscr{L}(S_{gdec}/G)$. This implies

$$\begin{aligned}
&\sigma \notin S_{gdec}(s)\\
&\Rightarrow \sigma \notin \wedge S_{P_i}(s)\\
&\Rightarrow (\exists i \in \{1, \dots, n\})[\sigma \notin S_{P_i}(P_i(s))]\\
&\Rightarrow (\exists i \in \{1, \dots, n\})[(P_i^{-1}P_i(s))\sigma \cap \overline{K} = \emptyset]\\
&\Rightarrow s\sigma \notin \overline{K}
\end{aligned}$$

It is a contradiction.

Case 3: $(\sigma \in \Sigma_{c,d})$ By the definition of S_{P_i} and D&A co-observability,

$$\begin{aligned}
s\sigma \in \overline{K} &\Rightarrow (\exists i \in \{1, \dots, n\})[(P_i^{-1}P_i(s) \cap \overline{K})\sigma \cap \mathscr{L}(G) \subseteq \overline{K}]\\
&\Rightarrow (\exists i \in \{1, \dots, n\})[\sigma \in S_{P_i}(P_i(s))]\\
&\Rightarrow \sigma \in \vee S_{P_i}(s)\\
&\Rightarrow \sigma \in S_{gdec}(s)
\end{aligned}$$

Overall, we have that

$$[s \in \overline{K}] \wedge [\sigma \in S_{gdec}(s)] \wedge [s\sigma \in \mathscr{L}(G)]$$

which in turns implies that

$$[s \in \mathscr{L}(S_{gdec}/G)] \wedge [\sigma \in S_{gdec}(s)] \wedge [s\sigma \in \mathscr{L}(G)]$$

using the induction hypothesis. It then immediately follows that $s\sigma \in \mathscr{L}(S_{gdec}/G)$. This completes the proof of the induction step and $\overline{K} = \mathscr{L}(S_{gdec}/G)$. ■

We note that there are four parts in the decision rule (7). The first part is for the locally controllable events processed disjunctively. The antipermissive rule is applied to this set of events. For the second part, the permissive rule is employed for the locally controllable events forwarded to the conjunctive fusion rule. It is interesting to note the difference of the default control actions of the permissive and antipermissive rules for $\Sigma_c \setminus \Sigma_{c,i}$. For the antipermissive rule with disjunctive fusion, the default control action over $\Sigma_c \setminus \Sigma_{c,i}$ in $\Sigma_{c,d}$ is disablement. Under disjunctive fusion, not locally controllable implies that the local supervisor cannot enable $\Sigma_c \setminus \Sigma_{c,i}$ globally. Therefore, $\Sigma_c \setminus \Sigma_{c,i}$ which are to be processed by disjunctive fusion should not appear in the decision rule (7).

Equipped with the above theorem, we have an immediate corollary revealing the solvability condition of the disjunctive architecture.

COROLLARY 1 *Consider a language $K \subseteq \mathscr{L}_m(G)$ where $K \neq \emptyset$. There exist a nonblocking disjunctive supervisor such that $\mathscr{L}_m(\vee_i S_{P_i}/G) = K$ and $\mathscr{L}(\vee_i S_{P_i}/G) = \overline{K}$ iff the three following conditions hold:*

1. *K is controllable w.r.t. $\mathscr{L}(G)$ and Σ_{uc}.*

2. *K is D&A co-observable w.r.t $\mathscr{L}(G), \Sigma_{o,1}, \Sigma_{c,1}, \ldots, \Sigma_{o,n}, \Sigma_{c,n}$.*

3. *K is $\mathscr{L}_m(G)$-closed.*

3.1. Realization of Supervisors

Let us recall the local decision rule (7). One can see that the local decision rules are decoupled from each other even though they work together in the context of the generalized architecture of Figure 3. In addition to the decoupling of the local decision rules, the information state $(P_i^{-1}P_i(s) \cap \overline{K})$ used in (7) is also independent of the decision rule (7). These observations lead us to propose the following approach to supervisor synthesis. Assume that the automaton describing the desired language is a strict subautomaton of the automaton generating the uncontrolled language. When the desired language is achievable (namely, it satisfies the conditions in Theorem 1), it is possible to design the estimator and control actions sequentially. That is, we can:

1. Build the local observers (estimators) of the automaton corresponding to the desired language.

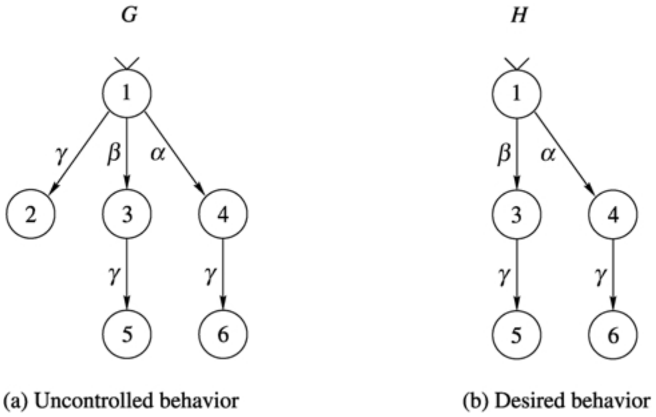
2. Find the local control action for each local observer state according to the decision rule (7).

Moreover, supervision can be conducted in an on-line manner, if so desired. The next local observer state of the desired language can be found on-line upon the occurrence of a locally observable event and the local decision for this local observer state can be calculated once the local observer state is known. Note that the computation of the new observer state only requires the previous observer state and the current control action can be computed based on the current observer state only. For more details on the realization of supervisors, the reader is directed to Chapter 3 of Cassandras and Lafortune (1999).

4. Properties of the Architectures

Let us define the following classes of languages where M is assumed to be prefix-closed:

$$\begin{aligned}
\mathscr{L}_{cen}(K) &= \{L \subseteq K : L \text{ is observable w.r.t. } M, \Sigma_o, \Sigma_c\}\\
\mathscr{L}_{DA}(K) &= \{L \subseteq K : L \text{ is D\&A co-observable w.r.t. } M, \Sigma_{o,1}, \Sigma_{c,1}, \Sigma_{o,2}, \Sigma_{c,2}\}\\
\mathscr{L}_{CP}(K) &= \{L \subseteq K : L \text{ is C\&P co-observable w.r.t. } M, \Sigma_{o,1}, \Sigma_{c,1}, \Sigma_{o,2}, \Sigma_{c,2}\}\\
\mathscr{L}_{gdec}(K) &= \{L \subseteq K : \exists \Sigma_{c,d} \text{ and } \Sigma_{c,e} \text{ s.t. } \Sigma_{c,d} \cap \Sigma_{c,e} = \emptyset, \Sigma_{c,d} \cup \Sigma_{c,e} = \Sigma_c,\\
&\qquad L \text{ is co-observable w.r.t. } M, \Sigma_{o,1}, \Sigma_{c,d,1}, \Sigma_{c,e,1}, \Sigma_{o,2}, \Sigma_{c,d,2}, \Sigma_{c,e,2}\}
\end{aligned}$$



(a) Uncontrolled behavior

(b) Desired behavior

Figure 4. System model and specification.

Since the controllability of the desired language is a common required condition for the existence of supervisors among all architectures, the classes of languages defined above determine the performance (the class of achievable languages) of the architectures.

First, we claim that the conjunctive and the disjunctive architectures are incomparable.

PROPOSITION 3 *In general, $\mathscr{L}_{DA}(K) \not\subset \mathscr{L}_{CP}(K)$ and $\mathscr{L}_{CP}(K) \not\subset \mathscr{L}_{DA}(K)$.*

Proof: We prove this by finding elements in $\mathscr{L}_{DA}(K) \setminus \mathscr{L}_{CP}(K)$ and $\mathscr{L}_{CP}(K) \setminus \mathscr{L}_{CA}(K)$. First we present an element in $\mathscr{L}_{DA}(K) \setminus \mathscr{L}_{CP}(K)$. Figure 4 depicts the specification and system model where $K = \mathscr{L}_m(H) = \mathscr{L}(H)$ and $M = \mathscr{L}_m(G) = \mathscr{L}(G)$ (marking is omitted for all states). We set $\Sigma_{o,1} = \{\alpha, \gamma\}$, $\Sigma_{o,2} = \{\beta, \gamma\}$, $\Sigma_{c,1} = \Sigma_{c,2} = \{\gamma\}$. Then it is easy to see that, for $\gamma \in \mathscr{L}(G) \setminus \mathscr{L}(H)$,

$$\{\beta\gamma\} = P_1^{-1}(P_1(\varepsilon))\gamma \cap \mathscr{L}(H) \text{ and } \{\alpha\gamma\} = P_2^{-1}(P_2(\varepsilon))\gamma \cap \mathscr{L}(H)$$

Since $\varepsilon \in \mathscr{L}(H)$, we conclude that $\mathscr{L}(H)$ is not C&P co-observable w.r.t. $\mathscr{L}(G)$, $\Sigma_{o,1}$, $\Sigma_{c,1}$, $\Sigma_{o,2}$, $\Sigma_{c,2}$. This implies that both supervisors cannot prevent γ from occurring without sacrificing some legal behavior. Hence, there does not exist a set of local supervisors generating the desired language $\mathscr{L}(H)$ conjunctively.

The only two controllable legal traces of $\mathscr{L}(H)$ are $\beta\gamma$ and $\alpha\gamma$. Moreover, it is easy to verify that

$$(P_1^{-1}(P_1(\alpha)) \cap \mathscr{L}(H))\gamma \cap \mathscr{L}(G) = \{\alpha\gamma\} \subset \mathscr{L}(H)$$
$$(P_2^{-1}(P_2(\beta)) \cap \mathscr{L}(H))\gamma \cap \mathscr{L}(G) = \{\beta\gamma\} \subset \mathscr{L}(H)$$

Therefore, $\mathscr{L}(H)$ is D&A co-observable w.r.t. $\mathscr{L}(G)$, $\Sigma_{o,1}$, $\Sigma_{c,1}$, $\Sigma_{o,2}$, $\Sigma_{c,2}$. This proves that $\mathscr{L}_{DA}(K) \not\subset \mathscr{L}_{CP}(K)$.

Moreover, it is easy to verify the controllability (the illegal continuation is with the

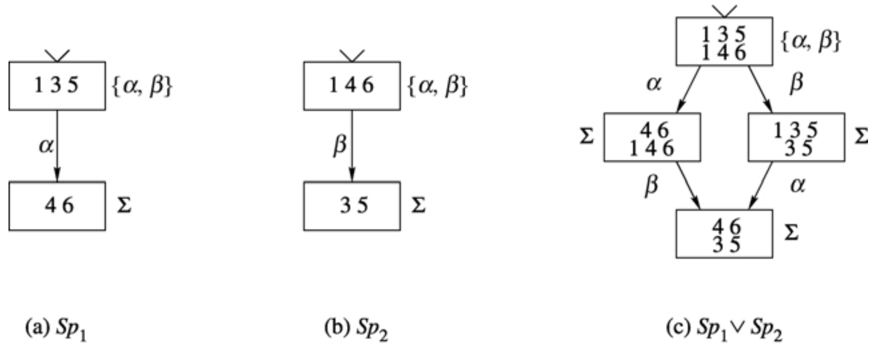


Figure 5. Disjunctive supervisor.

controllable event γ) and $\mathscr{L}_m(G)$-closure (language specifications are prefix-closed) of $\mathscr{L}(H)$. By following the antipermissive local decision rule in (2), we can construct local supervisors and their disjunction (global control action is the union of each local supervisor's decision) as in Figure 5 by following the method presented in Section 3.1. With this disjunctive supervisor, the desired language $\mathscr{L}(H)$ can be achieved. We may verify that $\mathscr{L}(S_{P_1} \vee S_{P_2}/G) = \mathscr{L}(H)$ by adding self-loops for enabled unobservable events at each disjunctive supervisor state and executing the product the disjunctive supervisor with G.

Next, we present an example of an element in $\mathscr{L}_{CP}(K) \setminus \mathscr{L}_{DA}(K)$. Let us change the desired behavior as shown in Figure 6. It is easy to see that, for $\gamma \in \mathscr{L}(H)$,

$$(P_1^{-1}(P_1(\varepsilon)) \cap \mathscr{L}(H))\gamma \cap \mathscr{L}(G) = \{\beta\gamma\} \nsubseteq \mathscr{L}(H)$$
$$(P_2^{-1}(P_2(\varepsilon)) \cap \mathscr{L}(H))\gamma \cap \mathscr{L}(G) = \{\alpha\gamma\} \nsubseteq \mathscr{L}(H)$$

This implies that $\mathscr{L}(H)$ is not D&A co-observable w.r.t. $\mathscr{L}(G)$, $\Sigma_{o,1}$, $\Sigma_{c,1}$, $\Sigma_{o,2}$, $\Sigma_{c,2}$. Therefore, there is no disjunctive supervisor that can achieve $\mathscr{L}(H)$. However,

$$P_1^{-1}(P_1(\alpha))\gamma \cap \mathscr{L}(H) = \emptyset \text{ and } P_2^{-1}(P_2(\beta))\gamma \cap \mathscr{L}(H) = \emptyset$$

implying that $\mathscr{L}(H)$ is C&P co-observable w.r.t. $\mathscr{L}(G)$, $\Sigma_{o,1}$, $\Sigma_{c,1}$, $\Sigma_{o,2}$, $\Sigma_{c,2}$.

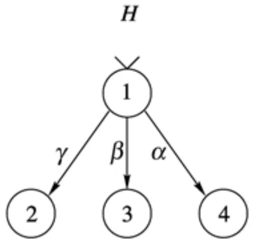


Figure 6. Desired behavior.

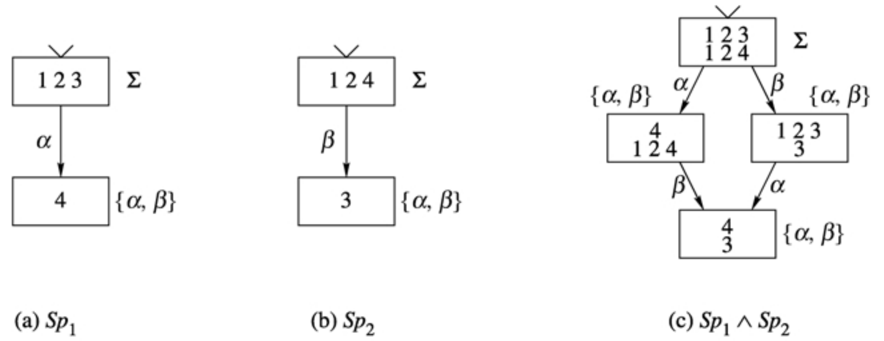


Figure 7. Conjunctive supervisor.

Controllability and $\mathscr{L}_m(G)$-closure hold as before. Therefore, the conjunctive supervisor can be constructed by following the permissive local decision rule in (1). Figure 7 is the realization of local supervisors and their conjunction. We may verify that $\mathscr{L}(S_{P_1} \wedge S_{P_2}/G) = \mathscr{L}(H)$ as well. ■

PROPOSITION 4 *In general,* $\mathscr{L}_{DA}(K) \cup \mathscr{L}_{CP}(K) \subseteq \mathscr{L}_{gdec}(K)$.

Proof: It is easy to see that the general architecture is reduced to the conjunctive (disjunctive) architecture when $\Sigma_{c,d} = \emptyset(\Sigma_{c,e} = \emptyset)$. This implies that, with the freedom of selection of $\Sigma_{c,d}$ and $\Sigma_{c,e}$, the class of languages supervisable under the general architecture includes those of the conjunctive and the disjunctive architectures. There are instances where inclusion is proper. We present an example that demonstrates the existence of an element in $\mathscr{L}_{gdec}(K) \setminus (\mathscr{L}_{DA}(K) \cup \mathscr{L}_{CP}(K))$. Consider the automata G and H shown in Figure 8(a) and 8(b), respectively. We set $\Sigma_{o,1} = \{\alpha, \gamma, \delta\}$, $\Sigma_{o,2} = \{\beta, \gamma, \delta\}$, $\Sigma_{c,1} = \Sigma_{c,2} = \{\gamma, \delta\}$. Observe that $K = \mathscr{L}_m(H) = \mathscr{L}(H)$ and $M = \mathscr{L}_m(G) = \mathscr{L}(G)$ (i.e., marking is omitted for all states). It is easy to see that, for $\gamma \in \mathscr{L}(G) \setminus \mathscr{L}(H)$,

$$\{\beta\gamma\} \subset P_1^{-1}(P_1(\varepsilon))\gamma \cap \mathscr{L}(H) \text{ and } \{\alpha\gamma\} \subset P_2^{-1}(P_2(\varepsilon))\gamma \cap \mathscr{L}(H)$$

Since $\varepsilon \in \mathscr{L}(H)$, we conclude that $\mathscr{L}(H)$ is not C&P co-observable w.r.t. $\mathscr{L}(G)$, $\Sigma_{o,1}$, $\Sigma_{c,1}$, $\Sigma_{o,2}$, $\Sigma_{c,2}$.

We can also see that for $\beta\gamma\delta \in \mathscr{L}(H)$,

$$\{\beta\gamma\beta\delta\} \subset (P_1^{-1}(P_1(\beta\gamma)) \cap \mathscr{L}(H))\delta \cap \mathscr{L}(G), \{\beta\gamma\beta\delta\} \not\subseteq \mathscr{L}(H)$$
$$\{\beta\gamma\alpha\delta\} \subset (P_2^{-1}(P_2(\beta\gamma)) \cap \mathscr{L}(H))\delta \cap \mathscr{L}(G), \{\beta\gamma\alpha\delta\} \not\subseteq \mathscr{L}(H)$$

This implies that $\mathscr{L}(H)$ is not D&A co-observable w.r.t. $\mathscr{L}(G)$, $\Sigma_{o,1}$, $\Sigma_{c,1}$, $\Sigma_{o,2}$, $\Sigma_{c,2}$.

Let us set $\Sigma_{c,e} = \{\delta\}$ and $\Sigma_{c,d} = \{\gamma\}$. The controllable legal traces of $\mathscr{L}(H)$ terminating

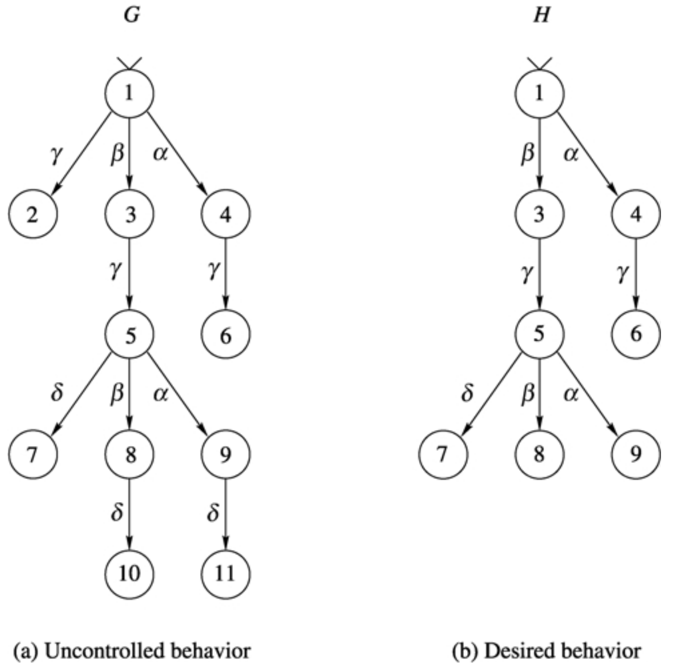


Figure 8. System model and specification.

with the event in $\Sigma_{c,d}$ are $\beta\gamma$ and $\alpha\gamma$. By (3), we have $\Sigma_{c,d,1} = \Sigma_{c,d,2} = \{\gamma\}$ as well. The following inclusions can be shown:

$$(P_1^{-1}(P_1(\alpha)) \cap \mathscr{L}(H))\gamma \cap \mathscr{L}(G) = \{\alpha\gamma\} \subset \mathscr{L}(H)$$
$$(P_2^{-1}(P_2(\beta)) \cap \mathscr{L}(H))\gamma \cap \mathscr{L}(G) = \{\beta\gamma\} \subset \mathscr{L}(H)$$

Therefore, we conclude that $\mathscr{L}(H)$ is D&A co-observable w.r.t. $\mathscr{L}(G)$, $\Sigma_{o,1}$, $\Sigma_{c,d,1}$, $\Sigma_{o,2}$, $\Sigma_{c,d,2}$.

On the other hand, $\beta\gamma\beta\delta$ and $\beta\gamma\alpha\delta$ are the only illegal controllable continuations terminating with the event in $\Sigma_{c,e}$. Furthermore, $\Sigma_{c,e,1} = \Sigma_{c,e,2} = \{\delta\}$ by (3). The following equalities can be verified:

$$P_1^{-1}(P_1(\beta\gamma\alpha))\delta \cap \mathscr{L}(H) = \emptyset, P_2^{-1}(P_2(\beta\gamma\beta))\delta \cap \mathscr{L}(H) = \emptyset$$

This implies that $\mathscr{L}(H)$ is C&P co-observable w.r.t. $\mathscr{L}(G)$, $\Sigma_{o,1}$, $\Sigma_{c,e,1}$, $\Sigma_{o,2}$, $\Sigma_{c,e,2}$. Collectively, $\mathscr{L}(H)$ is co-observable w.r.t. $\mathscr{L}(G)$, $\Sigma_{o,1}$, $\Sigma_{o,2}$, $\Sigma_{c,d,1}$, $\Sigma_{c,d,2}$, $\Sigma_{c,e,1}$, $\Sigma_{c,e,2}$. Moreover, the controllability of $\mathscr{L}(H)$ is easily verified (the illegal continuations are with the controllable events γ and δ) and $\mathscr{L}_m(G)$-closure is trivial. Therefore, we can construct a generalized supervisor controlling $\mathscr{L}_m(H)$. By following the combined local decision rule in (7), local supervisors S_{P_1}, S_{P_2} (Figure 9), and their fusion S_{gdec} (Figure 10) are

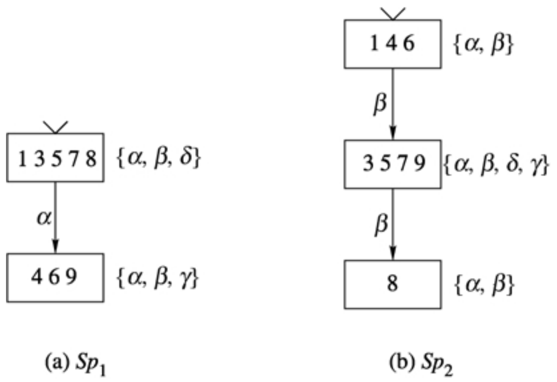


Figure 9. Local supervisors.

constructed. Note that $\Sigma_{c,e} = \{\delta\}$ is fused by the intersection of local decisions while $\Sigma_{c,d} = \{\gamma\}$ is fused by the union of local decisions. More detailed construction procedure is described in Section 3.1. With this generalized supervisor, the desired language $\mathscr{L}(H)$ can be achieved. ■

It is well known that the centralized architecture outperforms the conjunctive architecture; there are observable languages that are not C&P co-observable. It turns out that the centralized architecture also outperforms the general architecture: there are observable languages that are not co-observable.

Proposition 2 reveals that the centralized version of D&A co-observability is another equivalent statement of observability. With this, it is straightforward to see that observability w.r.t. $\mathscr{L}(G)$, Σ_o, Σ_c implies D&A co-observability w.r.t. $\mathscr{L}(G)$, $\Sigma_{o,1}$, $\Sigma_{c,1}$, ..., $\Sigma_{o,n}$, $\Sigma_{c,n}$. This leads us to the following result.

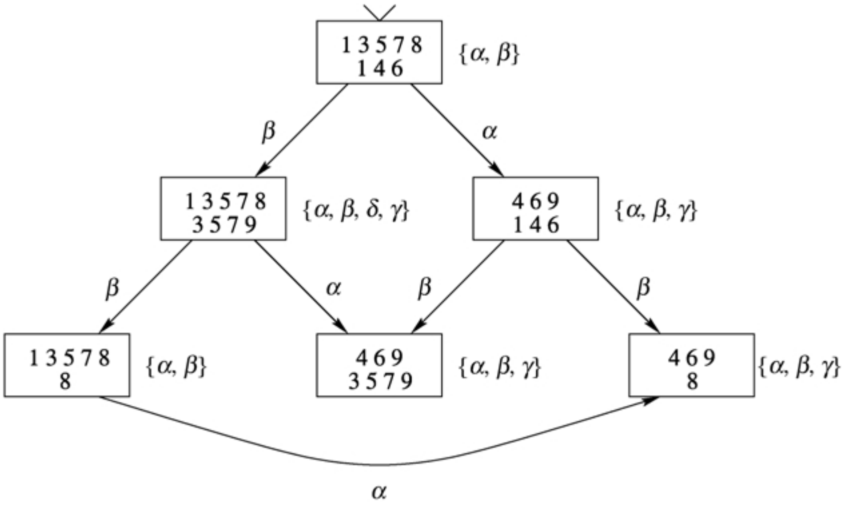


Figure 10. Generalized supervisor.

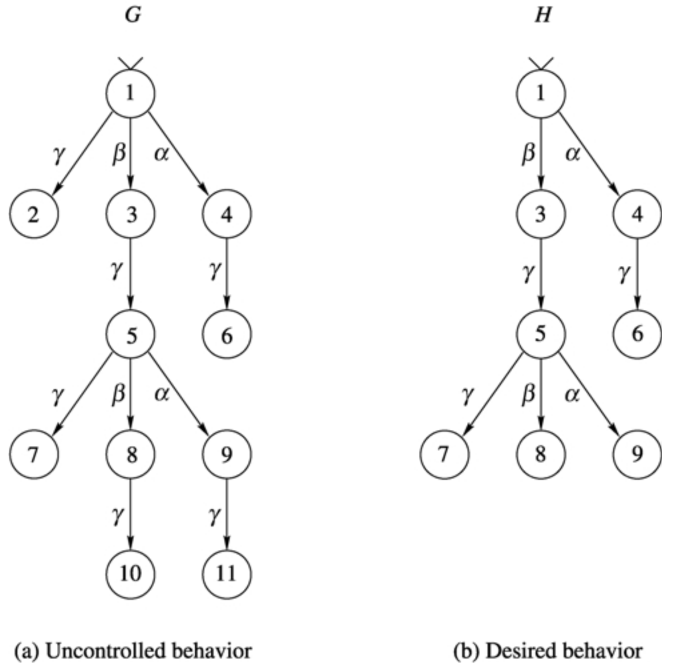


Figure 11. System model and specification.

PROPOSITION 5 $\mathscr{L}_{gdec}(K) \subseteq \mathscr{L}_{cen}(K)$.

Proof: Observability w.r.t. M, Σ_o, Σ_c implies C&P and D&A co-observability w.r.t. M, $\Sigma_{o,1}, \Sigma_{c,1}, \ldots, \Sigma_{o,n}, \Sigma_{c,n}$. C&P and D&A co-observability w.r.t. $M, \Sigma_{o,1}, \Sigma_{c,1}, \ldots, \Sigma_{o,n}, \Sigma_{c,n}$ implies co-observability w.r.t. $M, \Sigma_{o,1}, \Sigma_{c,d,1}, \Sigma_{c,e,1}, \ldots, \Sigma_{o,n}, \Sigma_{c,d,n}, \Sigma_{c,e,n}$, for any partition of Σ_c. Therefore, we get $\mathscr{L}_{gdec}(K) \subseteq \mathscr{L}_{cen}(K)$. Inclusion is proper, in general. We present an example to demonstrate this. Consider the uncontrolled and the desired behaviors in Figure 11(a) and Figure 11(b), respectively. With $\Sigma_{o,1} = \{\alpha, \gamma\}$, $\Sigma_{c,1} = \{\gamma\}$, $\Sigma_{o,2} = \{\beta, \gamma\}$, and $\Sigma_{c,2} = \{\gamma\}$, it is straightforward to show that $K = \mathscr{L}(H)$ is observable w.r.t. $M = \mathscr{L}(G)$, $\Sigma_o = \Sigma_{o,1} \cup \Sigma_{o,2}$, $\Sigma_c = \Sigma_{c,1} \cup \Sigma_{c,2}$. Since $\Sigma_c = \{\gamma\}$, two partitions, $\Sigma_{c,e} = \{\gamma\}$, $\Sigma_{c,d} = \emptyset$ and $\Sigma_{c,e} = \emptyset$, $\Sigma_{c,d} = \{\gamma\}$ are possible. It can be shown that neither partition satisfies the co-observability of $\mathscr{L}(H)$. Details are straightforward and omitted. ■

The relation between the classes of languages defined at the beginning of this section are summarized in Figure 12.

5. Polynomial Test for Co-observability

In this section, we investigate the computational complexity of testing co-observability. First, we present a polynomial time algorithm for testing D&A co-observability. One

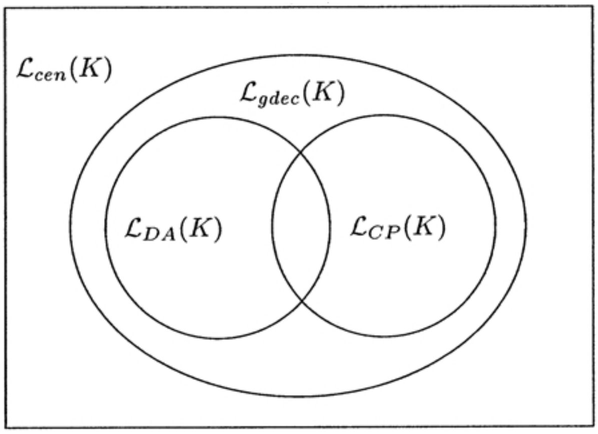


Figure 12. Performance comparison.

obvious way of testing D&A co-observability is to construct local supervisors with the antipermissive rule and see if the disjunctively generated language is the given desired language or not.[1] However, the computational complexity of constructing a partial-observation supervisor is exponential in the worst case (Tsitsiklis, 1989). Following the method of Tsitsiklis (1989) (for observability) and Rudie and Willems (1995) (for C&P co-observability), we present two ways of verifying D&A co-observability. These results indicate that testing the solvability of the disjunctive architecture can be done in polynomial time. The results are stated for two local supervisors. However, the extension to any finite number of local supervisors is straightforward.

First, we present a language formula for D&A co-observability that can be verified in polynomial time.

THEOREM 2 *A language $K \subseteq M = \overline{M}$ is D&A co-observable w.r.t. M, $\Sigma_{o,1}$, $\Sigma_{c,1}$, $\Sigma_{o,2}$, $\Sigma_{c,2}$ iff*

$$\begin{aligned}
&\Bigg[\bigcup_{\sigma\in\Sigma_{c,1}\cap\Sigma_{c,2}}\left\{\bigcap_{i=1}^{2}(P_i^{-1}[P_i\{\overline{K}\sigma\cap(M\setminus\overline{K})\}]\cap\overline{K}\sigma)\right\}\\
&\cup\bigcup_{\sigma\in\Sigma_{c,1}\setminus\Sigma_{c,2}}(P_1^{-1}[P_1\{\overline{K}\sigma\cap(M\setminus\overline{K})\}]\cap\overline{K}\sigma)\\
&\cup\bigcup_{\sigma\in\Sigma_{c,2}\setminus\Sigma_{c,1}}(P_2^{-1}[P_2\{\overline{K}\sigma\cap(M\setminus\overline{K})\}]\cap\overline{K}\sigma)\Bigg]\\
&\cap M\subseteq M\setminus\overline{K}
\end{aligned}\qquad(10)$$

Proof: The proof can be found in Yoo (2002). ■

The following facts from Hopcroft (1979) are stated for the justification of the polynomial complexity of the verification of D&A co-observability.

FACT 1 *Language intersection can be done in polynomial time w.r.t. the product of the sizes of the state sets of the automata involved (these automata can be non-deterministic).*

FACT 2 *Language containment $K \subseteq M$ can be tested in polynomial time if the automaton recognizing M is deterministic.*

FACT 3 *If the automaton recognizing M is deterministic, its complement can be computed in polynomial time.*

FACT 4 *Since $\mathscr{L}(G) \setminus \mathscr{L}(H) = \mathscr{L}(G) \cap (\mathscr{L}(H))^c$, Facts 1 and 3 imply the polynomial computational complexity of set difference if H is deterministic. Also, the resulting automaton is deterministic if all the involved automata are deterministic.*

The projection operation can be realized by replacing unobservable transitions in the finite state machine with the null string ε, thus creating a non-deterministic automaton. Exponential complexity would arise if we converted this non-deterministic automaton into a deterministic one. However, this conversion is not needed to perform the operations in the set inclusion (10). Using this observation and the above facts, the following result holds.

THEOREM 3 *Given two (deterministic) automata H and G, D&A co-observability of $\mathscr{L}_m(H)$ can be checked in polynomial time w.r.t. $|Q^G|$ and $|Q^H|$, where $|Q^G|$ and $|Q^H|$ are the sizes of the state sets of automata G and H, respectively.*

Another approach for proving that D&A co-observability is verifiable in polynomial time is to construct a special nondeterministic automaton, denoted by $M_d(\Sigma_c)$. This approach is based on the construction of the automaton $M(\Sigma_c)$ in Rudie and Willems (1995) for the property of C&P co-observability. Automaton $M_d(\Sigma_c)$ is built as follows:

$$M_d(\Sigma_c) = \left(Q^{M_d(\Sigma_c)}, \Sigma, \delta^{M_d(\Sigma_c)}, q_0^{M_d(\Sigma_c)}, Q_m^{M_d(\Sigma_c)}\right)$$

where

$$\begin{aligned} Q^{M_d(\Sigma_c)} &:= Q^G \times Q^H \times Q^G \times Q^H \times Q^H \cup \{d\} \\ q_0^{M_d(\Sigma_c)} &:= (q_0^G, q_0^H, q_0^G, q_0^H, q_0^H), \text{ and} \\ Q_m^{M_d(\Sigma_c)} &:= \{d\} \end{aligned}$$

Before the nondeterministic transition rule for $M_d(\Sigma_c)$ is defined, let us define the following set of conditions.

For $\sigma \in \Sigma_c$,

$$\left.\begin{array}{ll}
\delta^G(q_1, \sigma) & \text{is defined if } \sigma \in \Sigma_{c,1} \\
\delta^H(q_2, \sigma) & \text{is not defined if } \sigma \in \Sigma_{c,1} \\
\delta^G(q_3, \sigma) & \text{is defined if } \sigma \in \Sigma_{c,2} \\
\delta^H(q_4, \sigma) & \text{is not defined if } \sigma \in \Sigma_{c,2} \\
\delta^H(q_5, \sigma) & \text{is defined}
\end{array}\right\} (*)$$

Following the approach in Rudie and Willems (1995), it can be verified that conditions $(*)$ imply a violation of D&A co-observability. The transition relation $\delta^{M_d(\Sigma_c)}$ is defined as follows.

For $\sigma \notin \Sigma_{o,1}$ and $\sigma \notin \Sigma_{o,2}$,

$$\delta^{M_d(\Sigma_c)}((q_1, q_2, q_3, q_4, q_5), \sigma) = \begin{cases} (\delta^G(q_1, \sigma), \delta^H(q_2, \sigma), q_3, q_4, q_5) \\ (q_1, q_2, \delta^G(q_3, \sigma), \delta^H(q_4, \sigma), q_5) \\ (q_1, q_2, q_3, q_4, \delta^H(q_5, \sigma)) \\ (\delta^G(q_1, \sigma), \delta^H(q_2, \sigma), \delta^G(q_3, \sigma), \delta^H(q_4, \sigma), \delta^H(q_5, \sigma)) \\ d \quad \text{if } (*) \end{cases}$$

For $\sigma \notin \Sigma_{0,1}$ and $\sigma \in \Sigma_{0,2}$,

$$\delta^{M_d(\Sigma_c)}((q_1, q_2, q_3, q_4, q_5), \sigma) = \begin{cases} (\delta^G(q_1, \sigma), \delta^H(q_2, \sigma), q_3, q_4, q_5) \\ (q_1, q_2, \delta^G(q_3, \sigma), \delta^H(q_4, \sigma), \delta^H(q_5, \sigma)) \\ (\delta^G(q_1, \sigma), \delta^H(q_2, \sigma), \delta^G(q_3, \sigma), \delta^H(q_4, \sigma), \delta^H(q_5, \sigma)) \\ d \quad \text{if } (*) \end{cases}$$

For $\sigma \in \Sigma_{o,1}$ and $\sigma \notin \Sigma_{o,2}$,

$$\delta^{M_d(\Sigma_c)}((q_1, q_2, q_3, q_4, q_5), \sigma) = \begin{cases} (\delta^G(q_1, \sigma), \delta^H(q_2, \sigma), q_3, q_4, \delta^H(q_5, \sigma)) \\ (q_1, q_2, \delta^G(q_3, \sigma), \delta^H(q_4, \sigma), q_5) \\ (\delta^G(q_1, \sigma), \delta^H(q_2, \sigma), \delta^G(q_3, \sigma), \delta^H(q_4, \sigma), \delta^H(q_5, \sigma)) \\ d \quad \text{if } (*) \end{cases}$$

For $\sigma \in \Sigma_{o,1}$ and $\sigma \in \Sigma_{o,2}$,

$$\delta^{M_d(\Sigma_c)}((q_1, q_2, q_3, q_4, q_5), \sigma) = \begin{cases} (\delta^G(q_1, \sigma), \delta^H(q_2, \sigma), \delta^G(q_3, \sigma), \delta^H(q_4, \sigma), \delta^H(q_5, \sigma)) \\ d \quad \text{if } (*) \end{cases}$$

For $\sigma \in \Sigma$, $\delta^{M_d(\Sigma_c)}(d, \sigma)$ is undefined.

The following two results show that D&A co-observability is verifiable in polynomial time.

THEOREM 4 *Given two deterministic automata H and G, the construction of* $M_d(\Sigma_c)$ *takes polynomial time in* $|Q^G|$ *and* $|Q^H|$.

THEOREM 5 *Given two deterministic automata H and G,* $\mathscr{L}_m(H)$ *is not D&A co-observable iff* $\mathscr{L}_m(M_d(\Sigma_c)) \neq \emptyset$.

The proofs of Theorems 4 and 5 are similar to those of Proposition 3.1 and Proposition 3.2 in Rudie and Willems (1993), respectively. Roughly, in that reference authors construct the automaton $M(\Sigma_c)$ tracking the traces $s, s', s'' \in \mathscr{L}(H)$ such that

$$P_1(s) = P_1(s'), P_2(s) = P_2(s''), s'\sigma \in \mathscr{L}(H), s''\sigma \in \mathscr{L}(H), s\sigma \in \mathscr{L}(G) \setminus \mathscr{L}(H)$$

where $\sigma \in \Sigma_c$. These traces and event σ characterize the violation of C&P co-observability. Similarly, the automaton $M_d(\Sigma_c)$ tracks $s, s', s'' \in \mathscr{L}(H)$ such that

$$P_1(s) = P_1(s'), P_2(s) = P_2(s''), s'\sigma \in \mathscr{L}(G) \setminus \mathscr{L}(H), s''\sigma \in \mathscr{L}(G) \setminus \mathscr{L}(H), s\sigma \in \mathscr{L}(H)$$

where $\sigma \in \Sigma_c$. These traces and event σ characterize the violation of D&A co-observability. Note the difference between $M_d(\Sigma_c)$ and $M(\Sigma_c)$.[2] The state space of $M_d(\Sigma_c)$ is $Q^G \times Q^H \times Q^G \times Q^H \times Q^H \cup \{d\}$ instead of $Q^H \times Q^H \times Q^H \times Q^G \cup \{d\}$ for $M(\Sigma_c)$. The state spaces $Q^H \times Q^H \times Q^H \times Q^G$ for M and $Q^G \times Q^H \times Q^G \times Q^H \times Q^H$ for $M_d(\Sigma_c)$ are merely for tracking all traces $s, s', s'' \in \mathscr{L}(H)$ such that

$$P_1(s) = P_1(s') \text{ and } P_2(s) = P_2(s'')$$

Automaton $M_d(\Sigma_c)$ requires the product space of five automata because the characterization of the violation of D&A co-observability demands to track two illegal traces (this is done by two pairs of G and H) and one legal trace (the last product of H tracks this). Analogous reasoning can be applied for the automaton $M(\Sigma_c)$ tracking two legal traces and one illegal trace. The violation condition $(*)$ of $M_d(\Sigma_c)$ characterizes the failures of D&A co-observability and results in the marking of those continuations. We present an illustrative example of the verification of D&A co-observability using the automaton $M_d(\Sigma_c)$.

Example 1: We showed that $\mathscr{L}_m(H)$ depicted in Figure 6 is not D&A co-observable w.r.t. $\mathscr{L}(G)$, $\Sigma_{o,1}$, $\Sigma_{c,1}$, $\Sigma_{o,2}$, $\Sigma_{c,2}$ defined earlier. We show this result by constructing the automaton $M_d(\Sigma_c)$; the relevant part of $M_d(\Sigma_c)$ is shown in Figure 13. Recall that $\Sigma_{o,1} = \{\alpha, \gamma\}$, $\Sigma_{o,2} = \{\beta, \gamma\}$, and $\Sigma_{c,1} = \Sigma_{c,2} = \{\gamma\}$. Let us look at the following traces:

$$\gamma \in \mathscr{L}(H), \beta\gamma \in \mathscr{L}(G) \setminus \mathscr{L}(H), \alpha\gamma \in \mathscr{L}(G) \setminus \mathscr{L}(H) \tag{11}$$

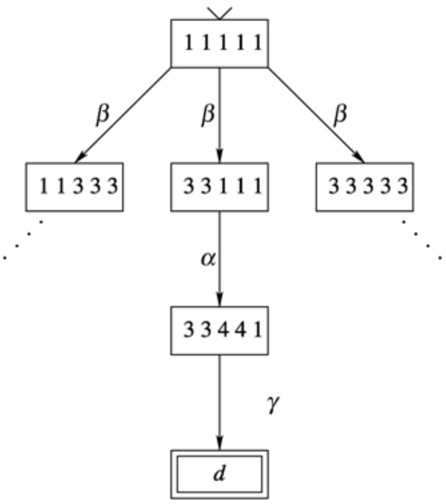


Figure 13. $M_d(\Sigma_c)$.

Since

$$P_1(\varepsilon) = P_1(\beta), P_2(\varepsilon) = P_2(\alpha), \quad \text{and} \quad \gamma \in \Sigma_{c,1} \cap \Sigma_{c,2}$$

the traces in (11) prevent $\mathscr{L}(H)$ from being D&A co-observable w.r.t. $\mathscr{L}(G)$, $\Sigma_{o,1}$, $\Sigma_{c,1}$, $\Sigma_{o,2}$, $\Sigma_{c,2}$. Now we construct the automaton $M_d(\Sigma_c)$. From the initial state, all events are active transitions. Let us follow the transition β. Since $\beta \in \Sigma_{o,2} \setminus \Sigma_{o,1}$, there are up to four possible transitions, but only three of them can be executed since condition $(*)$ does not hold:

$$\delta^{M_d(\Sigma_c)}((1,1,1,1,1),\beta) = \begin{cases} (3,3,1,1,1) \\ (1,1,3,3,3) \\ (3,3,3,3,3) \end{cases}$$

These states are reached by the following set of traces;

$$\begin{aligned} &(3,3,1,1,1) \leftarrow s = \varepsilon,\ s' = \beta,\ s'' = \varepsilon, \\ &(1,1,3,3,3) \leftarrow s = \beta,\ s' = \varepsilon,\ s'' = \beta, \\ &(3,3,3,3,3) \leftarrow s = \beta,\ s' = \beta,\ s'' = \beta \end{aligned}$$

Next, we do transition α from state $(3,3,1,1,1)$. Since $\alpha \in \Sigma_{o,1} \setminus \Sigma_{o,2}$, up to four transitions are possible. However, we only have one executable transition, namely

$$\delta^{M_d(\Sigma_c)}((3,3,1,1,1),\alpha) = (3,3,4,4,1)$$

and the corresponding traces are:

$$(3,3,4,4,1) \leftarrow s = \varepsilon, s' = \beta, s'' = \alpha$$

Finally, we execute the transition γ from $(3,3,4,4,1)$. Since $\gamma \in \Sigma_{o,1} \cap \Sigma_{o,2}$, up to two transitions are possible. However, $(3,3,4,4,1)$ has only one executable transition, namely

$$\delta^{M_d(\Sigma_c)}((3,3,4,4,1),\gamma) = d$$

The corresponding traces are

$$d \leftarrow s = \gamma, s' = \beta\gamma, s'' = \alpha\gamma$$

We can see that marked state d is reached by the set of traces that violate D&A co-observability.

In Tsitsiklis (1989), it is shown that the computational complexity of constructing a partial observation supervisor is exponential in the worst case even though solvability (namely, controllability and observability) can be checked in polynomial time. This is caused by the fact that supervisor construction has the same computational complexity as that of observer construction, which is exponential in the worst case. This negative result is inherited in decentralized architectures.

To verify the existence of a nonblocking supervisor such that $\mathscr{L}_m(S_{gdec}/G) = \mathscr{L}_m(H)$ for the general architecture, we need to determine the existence of a partition of Σ_c into $\Sigma_{c,d}$ and $\Sigma_{c,e}$ that satisfies co-observability. Given a partition of Σ_c, namely $\Sigma_{c,e}$ and $\Sigma_{c,d}$, it is straightforward to see (from the definition of co-observability) that co-observability with respect to this partition can be verified by constructing $M_d(\Sigma_{c,d})$ and $M(\Sigma_{c,e})$ and checking $\mathscr{L}_m(M_d(\Sigma_{c,d})) = \mathscr{L}_m(M(\Sigma_{c,e})) = \emptyset$. A brute force way of finding $\Sigma_{c,d}$ and $\Sigma_{c,e}$ that satisfies co-observability is to test C&P and D&A co-observability (by constructing $M_d(\Sigma_{c,d})$ and $M(\Sigma_{c,e})$ and checking $\mathscr{L}_m(M_d(\Sigma_{c,d})) = \mathscr{L}_m(M(\Sigma_{c,e})) = \emptyset$) for all combinations of $\Sigma_{c,d}$ and $\Sigma_{c,e}$. The number of combinations is exponential w.r.t. m, the number of controllable events. Here we propose a polynomial time algorithm to find a partition satisfying co-observability. This algorithm exploits the properties of the automata $M(\Sigma_c)$ and $M_d(\Sigma_c)$. We define the set of terminal events:

$$\Sigma_{ter}(K) := \{\sigma : s\sigma \in K\}$$

$\Sigma_{ter}(K)$ collects the events terminating the traces in K. The following theorem provides an algorithm to search for a partition of Σ_c satisfying co-observability.

THEOREM 6 *There exist $\Sigma_{c,e}$ and $\Sigma_{c,d}$, a partition of Σ_c, such that $\mathscr{L}_m(H)$ is co-observable w.r.t. $\mathscr{L}(G)$, $\Sigma_{o,1}$, $\Sigma_{c,d,1}$, $\Sigma_{c,e,1}$, $\Sigma_{o,2}$, $\Sigma_{c,d,2}$, $\Sigma_{c,e,2}$, iff*

$$\Sigma_{ter}(\mathscr{L}_m(M(\Sigma_c))) \cap \Sigma_{ter}(\mathscr{L}_m(M_d(\Sigma_c))) = \emptyset$$

Proof: ($\Rightarrow$) Let there exist $\Sigma_{c,e}$ and $\Sigma_{c,d}$, a partition of Σ_c, such that $\mathscr{L}_m(H)$ is co-observable w.r.t. $\mathscr{L}(G)$, $\Sigma_{o,1}$, $\Sigma_{c,d,1}$, $\Sigma_{c,e,1}$, $\Sigma_{o,2}$, $\Sigma_{c,d,2}$, $\Sigma_{c,e,2}$. This implies that

$$[\forall\sigma\in\Sigma_{c,e}, \sigma \notin \Sigma_{ter}(\mathscr{L}_m(M(\Sigma_c)))] \wedge [\forall\sigma\in\Sigma_{c,d}, \sigma \notin \Sigma_{ter}(\mathscr{L}_m(M_d(\Sigma_c)))]$$

Since $\Sigma_{ter}(\mathscr{L}_m(M(\Sigma_c)))$, $\Sigma_{ter}(\mathscr{L}_m(M_d(\Sigma_c))) \subseteq \Sigma_c$, we get the following set inclusions:

$$\begin{aligned}&\Sigma_{ter}(\mathscr{L}_m(M(\Sigma_c))) \subseteq \Sigma_c \setminus \Sigma_{c,e} = \Sigma_{c,d}\\&\Sigma_{ter}(\mathscr{L}_m(M_d(\Sigma_c))) \subseteq \Sigma_c \setminus \Sigma_{c,d} = \Sigma_{c,e}\end{aligned} \tag{12}$$

Since $\Sigma_{c,e}$ and $\Sigma_{c,d}$ are a partition of Σ_c, with (12), we get

$$\begin{aligned}&\Sigma_{ter}(\mathscr{L}_m(M(\Sigma_c))) \cap \Sigma_{ter}(\mathscr{L}_m(M_d(\Sigma_c))) \subseteq \Sigma_{c,d} \cap \Sigma_{c,e} = \emptyset\\&\Rightarrow \Sigma_{ter}(\mathscr{L}_m(M(\Sigma_c))) \cap \Sigma_{ter}(\mathscr{L}_m(M_d(\Sigma_c))) = \emptyset\end{aligned}$$

($\Leftarrow$) Suppose that $\Sigma_{ter}(\mathscr{L}_m(M(\Sigma_c))) \cap \Sigma_{ter}(\mathscr{L}_m(M_d(\Sigma_c))) = \emptyset$. We set

$$\Sigma_{c,d} = \Sigma_{ter}(\mathscr{L}_m(M(\Sigma_c))) \text{ and } \Sigma_{c,e} = \Sigma_c \setminus \Sigma_{c,d} \tag{13}$$

Then

$$\Sigma_{c,d} \cup \Sigma_{c,e} = \Sigma_c \text{ and } \Sigma_{c,d} \cap \Sigma_{c,e} = \emptyset$$

Moreover,

$$\Sigma_{ter}(\mathscr{L}_m(M(\Sigma_c))) \cap \Sigma_{c,e} = \emptyset \text{ and } \Sigma_{ter}(\mathscr{L}_m(M_d(\Sigma_c))) \cap \Sigma_{c,d} = \emptyset \tag{14}$$

This implies that for all $\sigma\in\Sigma_{c,e} \subseteq \Sigma_c$ ($\sigma\in\Sigma_{c,d} \subseteq \Sigma_c$), no violation of C&P (D&A) co-observability occurs. Therefore, we can conclude that $\mathscr{L}_m(H)$ is co-observable w.r.t. $\mathscr{L}(G)$, $\Sigma_{o,1}$, $\Sigma_{c,d,1}$, $\Sigma_{c,e,1}$, $\Sigma_{o,2}$, $\Sigma_{c,d,2}$, $\Sigma_{c,e,2}$. ■

The setting utilized in (13) is one of the partitions satisfying co-observability. Since $\Sigma_{ter}(\mathscr{L}_m(M(\Sigma_c)))$ and $\Sigma_{ter}(\mathscr{L}_m(M_d(\Sigma_c)))$ can be determined in polynomial time, this provides a polynomial time algorithm for finding a partition meeting the requirements of co-observability. Even though Theorem 6 is stated for two local supervisors, $M(\Sigma_c)$ and $M_d(\Sigma_c)$ can be straightforwardly extended to any finite number of local supervisors while still keeping polynomial complexity.[3] Provided with these $M(\Sigma_c)$ and $M_d(\Sigma_c)$, Theorem 6 is generalized to any finite number of local supervisors as well. Consequently, the following theorem can be stated.

THEOREM 7 *Given two deterministic automata H and G, the existence of a partition of* Σ_c, $\Sigma_{c,e}$ *and* $\Sigma_{c,d}$, *satisfying co-observability of* $\mathscr{L}_m(H)$ *w.r.t.* $\mathscr{L}(G)$, $\Sigma_{o,1}$, $\Sigma_{c,d,1}$, $\Sigma_{c,e,1}, \ldots, \Sigma_{o,n}$, $\Sigma_{c,d,n}$, $\Sigma_{c,e,n}$ *can be verified in polynomial time with respect to* $|Q^H|$ *and* $|Q^G|$, *and if such a partition exists, it can be found in polynomial time with respect to* $|Q^H|$ *and* $|Q^G|$ *as well.*

Now we state the result that completes the answer to problem (**P**) posed in Section 2.4.

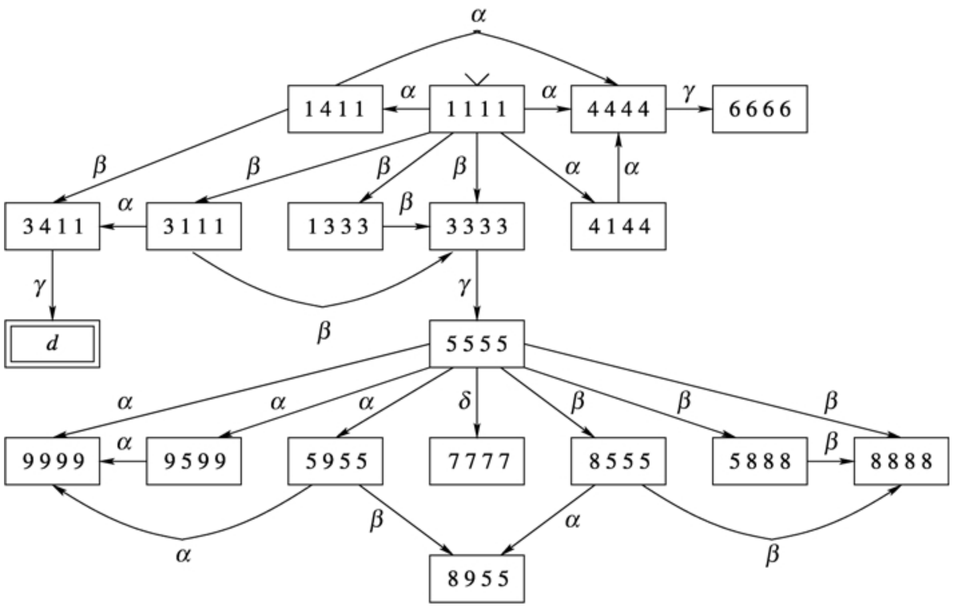


Figure 14. $M(\Sigma_c)$.

THEOREM 8 *Let K be controllable and* $\mathscr{L}_m(G)$*-closed. There exist a partition of* Σ_c *and a corresponding nonblocking generalized supervisor* S_{gdec} *such that* $\mathscr{L}(S_{gdec}/G) = \overline{K}$ *and* $\mathscr{L}_m(S_{gdec}/G) = K$ *iff*

$$\Sigma_{ter}(\mathscr{L}_m(M(\Sigma_c))) \cap \Sigma_{ter}(\mathscr{L}_m(M_d(\Sigma_c))) = \emptyset$$

Proof: Direct consequence of Theorem 1 and Theorem 6. ■

For a language K satisfying the conditions stated in Theorem 8, the synthesis of local supervisors achieving K can be done by following the partition rule (13) and the local decision rule (7). However, it is easy to observe that the partition of Σ_c may not be unique. All partitions satisfying (14) meet the co-observability condition. We present an example to illustrate the partitioning algorithm.

Example 2: We showed that $\mathscr{L}_m(H)$ *depicted in Figure 8(b) is not C&P and D&A co-observable w.r.t.* $\mathscr{L}(G)$, $\Sigma_{o,1}$, $\Sigma_{c,1}$, $\Sigma_{o,2}$, $\Sigma_{c,2}$ *but co-observable w.r.t.* $\mathscr{L}(G)$, $\Sigma_{o,1}$, $\Sigma_{c,d,1}$, $\Sigma_{c,e,1}$, $\Sigma_{o,2}$, $\Sigma_{c,d,2}$, $\Sigma_{c,e,2}$, *where* $\Sigma_{c,d} = \{\gamma\}$ *and* $\Sigma_{c,e} = \{\delta\}$. *The two corresponding automata* $M(\Sigma_c)$ *and* $M_d(\Sigma_c)$ *are shown in Figure 14 and Figure 15, respectively. We can see that* $\Sigma_{ter}(\mathscr{L}_m(M(\Sigma_c))) = \{\gamma\}$ *and* $\Sigma_{ter}(\mathscr{L}_m(M_d(\Sigma_c))) = \{\delta\}$. *Since* $\Sigma_{ter}(\mathscr{L}_m(M(\Sigma_c))) \cap \Sigma_{ter}(\mathscr{L}_m(M_d(\Sigma_c))) = \emptyset$, *there exists a partition of* Σ_c *satisfying co-observability by Theorem 6. By following the partition rule (13), we get*

$$\Sigma_{c,d} = \Sigma_{ter}(\mathscr{L}_m(M(\Sigma_c))) = \{\gamma\} \text{ and } \Sigma_{c,e} = \Sigma_c \setminus \Sigma_{c,d} = \{\delta\}$$

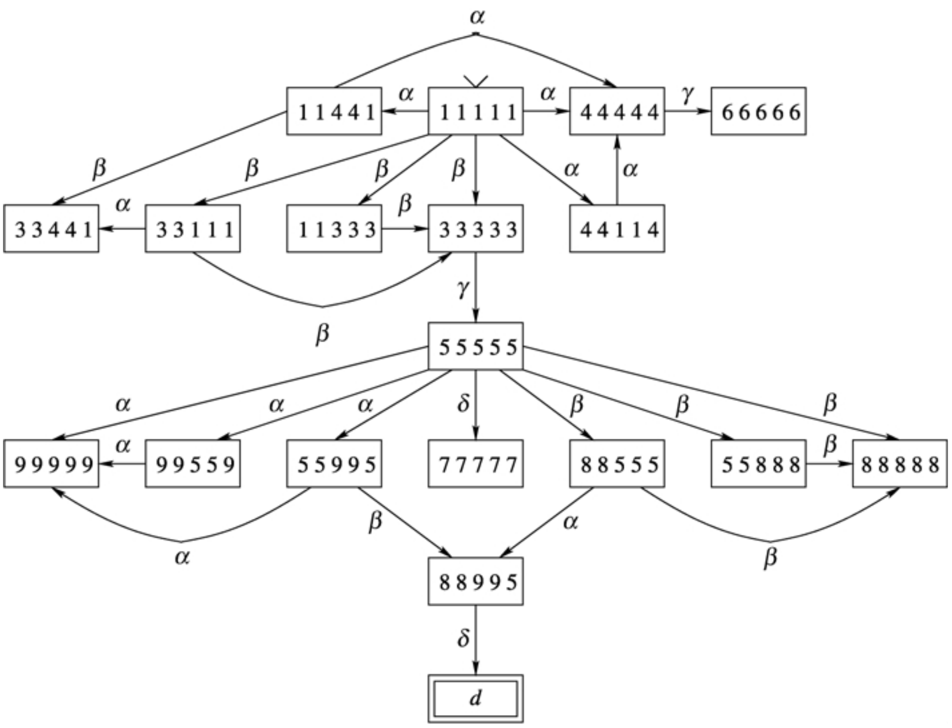


Figure 15. $M_d(\Sigma_c)$.

With this configuration, $\mathscr{L}_m(H)$ is co-observable w.r.t. $\mathscr{L}(G)$, $\Sigma_{o,1}$, $\Sigma_{c,d,1}$, $\Sigma_{c,e,1}$, $\Sigma_{o,2}$, $\Sigma_{c,d,2}$, $\Sigma_{c,e,2}$.

The objective of the following sections is to investigate how to deal with the situation where co-observability fails to hold. For this objective, we examine the algebraic properties of co-observability. These properties show that the supremal and infimal elements of the class of co-observable languages need not exist, in general. Based on these results, some supervisor design issues under the general architecture are then discussed in Section 7.

6. Algebraic Properties of Co-observability

Supremal sublanguages (with respect to a given property) play a key role in supervisory control problems, since they provide the least restrictive solutions when the desired languages are not achievable under control (because they do not possess the property under consideration). Infimal superlanguages are also important (see, e.g., the range problem in Lin and Wonham, 1988; Rudie and Wonham, 1992). The existence of supremal and infimal languages can be verified by checking the closure under algebraic operations (union and intersection) of classes of languages. We present several algebraic properties of co-observability that are relevant in this regard.

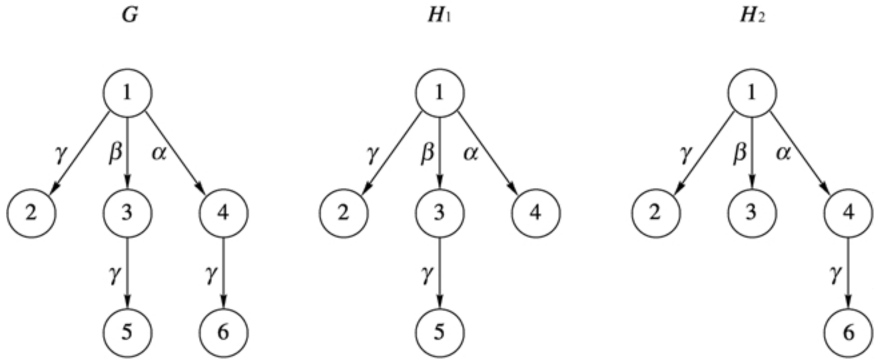


Figure 16. Co-observability example.

PROPOSITION 6 *D&A co-observability and co-observability are not preserved under union of languages.*

Proof: It is proved in Proposition 2 that C&P co-observability, D&A co-observability and observability are all equivalent in the context of the centralized control architecture. Given the fact that observability is not preserved under union, the result is immediate. ■

It is shown in Rudie and Wonham (1992) that C&P co-observability is preserved under intersection when prefix-closed languages are considered. However, the following proposition shows that this is not true for D&A co-observability.

PROPOSITION 7 *D&A co-observability and co-observability are not preserved under intersection of languages.*

Proof: Let $\Sigma_{o,1} = \{\alpha\}$, $\Sigma_{o,2} = \{\beta\}$, and $\Sigma_c = \Sigma_{c,1} = \Sigma_{c,2} = \{\gamma\}$ for the languages shown in Figure 16. $\mathscr{L}(G)$ is the uncontrolled language and $\mathscr{L}(H_1)$ and $\mathscr{L}(H_2)$ are the desired languages. The only traces terminated with a controllable event in $\mathscr{L}(H_1)$ are γ and $\beta\gamma$. For these traces, it is easy to see that

$$(P_1^{-1}P_1(\varepsilon) \cap \mathscr{L}(H_1))\gamma \cap \mathscr{L}(G) = \{\gamma, \beta\gamma\} \subset \mathscr{L}(H_1)$$
$$(P_2^{-1}P_2(\beta) \cap \mathscr{L}(H_1))\gamma \cap \mathscr{L}(G) = \{\beta\gamma\} \subset \mathscr{L}(H_1)$$

That is, $\mathscr{L}(H_1)$ is D&A co-observable w.r.t. $\mathscr{L}(G)$, $\Sigma_{o,1}$, $\Sigma_{c,1}$, $\Sigma_{o,2}$, $\Sigma_{c,2}$. Similarly, $\mathscr{L}(H_2)$ is D&A co-observable w.r.t. $\mathscr{L}(G)$, $\Sigma_{o,1}$, $\Sigma_{c,1}$, $\Sigma_{o,2}$, $\Sigma_{c,2}$. However, it is shown in Proposition 3 that $\mathscr{L}(H_1) \cap \mathscr{L}(H_2)$ is not D&A co-observable w.r.t. $\mathscr{L}(G)$, $\Sigma_{o,1}$, $\Sigma_{c,1}$, $\Sigma_{o,2}$, $\Sigma_{c,2}$. Since D&A co-observability is a special case of co-observability, co-observability is not preserved under intersection of languages as well. ■

Finally, we focus on the class of languages $\mathscr{L}_{DA}(K) \cap \mathscr{L}_{CP}(K)$. Let us pick two elements in $\mathscr{L}_{DA}(K) \cap \mathscr{L}_{CP}(K)$. Does the intersection of these two elements preserve C&P and D&A co-observability?

PROPOSITION 8 *(i) $\mathscr{L}_{DA}(K) \cap \mathscr{L}_{CP}(K)$ is not closed under intersection of its elements. (ii) $\mathscr{L}_{DA}(K) \cap \mathscr{L}_{CP}(K)$ is not closed under union of its elements.*

Proof: Let us return to Figure 16. We showed in the proof of Proposition 7 that $\mathscr{L}(H_1)$ and $\mathscr{L}(H_2)$ are D&A co-observable. Since

$$P_1^{-1}(P_1(\alpha))\gamma \cap \mathscr{L}(H_1) = \emptyset \text{ and } P_2^{-1}(P_2(\beta))\gamma \cap \mathscr{L}(H_2) = \emptyset$$

$\mathscr{L}(H_1)$ and $\mathscr{L}(H_2)$ are C&P co-observable as well. Since $\mathscr{L}(H_1), \mathscr{L}(H_2) \subseteq \mathscr{L}(G)$, these are elements of $\mathscr{L}_{DA}(\mathscr{L}(G)) \cap \mathscr{L}_{CP}(\mathscr{L}(G))$. But as was seen earlier, $\mathscr{L}(H_1) \cap \mathscr{L}(H_2)$ is not D&A co-observable. Therefore, $\mathscr{L}(H_1) \cap \mathscr{L}(H_2) \notin \mathscr{L}_{DA}(\mathscr{L}(G)) \cap \mathscr{L}_{CP}(\mathscr{L}(G))$. Hence, we conclude that $\mathscr{L}_{DA}(\mathscr{L}(G)) \cap \mathscr{L}_{CP}(\mathscr{L}(G))$ is not closed under intersection of languages.

By Proposition 2, observability is equivalent to C&P and D&A co-observability in the context of a centralized architecture. Moreover, observability is not preserved under union of languages. Therefore we conclude that $\mathscr{L}_{DA}(K) \cap \mathscr{L}_{CP}(K)$ is not closed under union, in general. ■

Properties 6, 7, and 8 are disappointing in the sense that the notion of co-observability, which is key to the generalized architecture of Figure 3, does not enjoy any of the algebraic properties that would be useful to address supervisor synthesis problems for desired behaviors that are not co-observable. For this reason, a different approach is necessary to tackle such problems. The next section presents new results in this regard.

7. Synthesis Results

7.1. Design of Safe Generalized Supervisors

In supervisory control problem, the desired language is *not* achievable, one may want to synthesize a safe supervisor that guarantees that the closed-loop behavior stays within the desired language. We call this the inclusion problem and define it as follows:

Inclusion Problem Given uncontrolled system G over the set of events Σ and legal language K, find a nontrivial supervisor S such that $\mathscr{L}(S/G) \subseteq \overline{K}$.

Note that this Inclusion Problem does not consider blocking/nonblocking of solutions. It has recently been shown in Lamouchi and Thistle (2000) and Tripakis (2001) that the inclusion problem for safe and nonblocking decentralized supervisors is undecidable when the specification is expressed in terms of $\mathscr{L}_m(S/G) \subseteq K$.

For the perfect observation case ($\Sigma = \Sigma_o$), the supremal controllable sublanguage of the desired language K, $(\overline{K})^{\uparrow(C)}$, is computable and provides the least restrictive solution to the inclusion problem. Due to the lack of existence of supremal observable sublanguages, several approaches have been developed for control under imperfect observation. For centralized architectures, the property of normality has been suggested in order to compute a ''suboptimal'' solution to the inclusion problem (Cho and Marcus, 1989a; Lin and Wonham, 1988). To improve upon this solution, other safe supervisor synthesis techniques were developed. Most of the efforts (Hadj-Alouane et al., 1996; Heymann and Lin, 1994) were devoted to the centralized architecture. There are very few results on the synthesis of safe supervisors in the context of decentralized architectures. One of the obstacles to the design of safe decentralized supervisors may be the mutual dependency of local decisions. To circumvent the dependency of local decisions,an intentional ''decoupling'' of the design of the local decision rules was suggested in Prosser and Kam (1997) and Prosser (1996). The idea is to design local supervisors separately by following the antipermissive rule and fuse them through various fusion rules. Even though the performance of the supervisor may be degraded due to the separation of local supervisor design, the simplicity of this approach circumvents the mutual dependency of local decision rules. However, a drawback of the approach in Prosser and Kam (1997) and Prosser (1996) is that local supervisors do not exploit the structure of the fusion rule that is *a priori* known to each local supervisor. In the approach that we propose, we also decouple the design of local supervisors. However, the fusion rule and the local decision rules are accounted for in this design in order to enrich the closed-loop behavior.

Assume that the desired language K is controllable.[4] Let us denote by S_{gdec} the supervisor obtained following the decision rule given in (7) with a given partition $\Sigma_{c,e}$ and $\Sigma_{c,d}$. In some sense, this means that S_{gdec} pretends that K is controllable and co-observable. We also build the automaton $M(\Sigma_c)$ and determine $\Sigma_{ter}(\mathscr{L}_m(M(\Sigma_c)))$. Since the controllable events in $\Sigma_{ter}(\mathscr{L}_m(M(\Sigma_c)))$ may cause a violation of safety (illegal continuation) if we follow the permissive rule for $\Sigma_{ter}(\mathscr{L}_m(M(\Sigma_c)))$, we should use the antipermissive rule for these events.

The nature of the permissive rule is to enable events when there is insufficient information. This rule can cause a violation of safety unless other local supervisors disable the events that would lead to illegal behavior. In contrast, the antipermissive rule disables events when there is insufficient information. This conservative approach prevents the closed-loop behavior from being illegal. We have the following theorem providing a procedure for the synthesis of a safe supervisor under the general architecture.

THEOREM 9 *If* $\Sigma_{ter}(\mathscr{L}_m(M(\Sigma_c))) \subseteq \Sigma_{c,d}$, *then synthesizing* S_{gdec} *according to (7) leads to* $\mathscr{L}(S_{gdec}/G) \subseteq \overline{K}$.

Proof: The proof is done by induction on the length of traces.

(Base of the induction) Since K is nonempty, $\varepsilon \in \mathscr{L}(S_{gdec}/G)$ and $\varepsilon \in \overline{K}$.

(Induction hypothesis) Suppose that for all $s \in \mathscr{L}(S_{gdec}/G)$ such that $|s| = n$, $s \in \overline{K}$.

(Inductive step) Assume that there exist $s \in \mathscr{L}(S_{gdec}/G)$ and $\sigma \in \Sigma$ such that $|s| = n$, $s\sigma \in \mathscr{L}(S_{gdec}/G)$, and $s\sigma \notin \overline{K}$. Since $\mathscr{L}(S_{gdec}/G) \subseteq \mathscr{L}(G)$, we get $s\sigma \in \mathscr{L}(G)$. By the controllability of $\overline{K}$,

$$\begin{aligned} & [s\sigma \in \mathscr{L}(G)] \wedge [s\sigma \notin \overline{K}] \wedge [s \in \overline{K}] \\ \Rightarrow & [s\sigma \in \mathscr{L}(G) \setminus \overline{K}] \wedge [\sigma \in \Sigma_c] \end{aligned}$$

Case 1: $(\sigma \in \Sigma_{c,e,1} \cap \Sigma_{c,e,2})$ Since $\Sigma_{ter}(\mathscr{L}_m(M(\Sigma_c))) \subseteq \Sigma_{c,d}$, this implies that $\Sigma_{ter}(\mathscr{L}_m(M(\Sigma_c))) \cap \Sigma_{c,e} = \emptyset$. Therefore, $\overline{K}$ is C&P co-observable w.r.t. $\mathscr{L}(G)$, $\Sigma_{o,1}$, $\Sigma_{o,2}$, $\Sigma_{c,e,1}$, $\Sigma_{c,e,2}$ and we have the following:

$$\begin{aligned}[s \in \overline{K}] \wedge [s\sigma \notin \overline{K}] &\Rightarrow [(\exists i \in \{1,2\})[(P_i^{-1}P_i(s))\sigma \cap \overline{K} = \emptyset]]\\ &\Rightarrow [\sigma \notin S_{gdec}(s)]\\ &\Rightarrow [s\sigma \notin \mathscr{L}(S_{gdec}/G)]\end{aligned}$$

This contradicts the assumption that $s\sigma \in \mathscr{L}(S_{gdec}/G)$.

The Cases 2 and 3 where $\sigma \in \Sigma_{c,e,1} \setminus \Sigma_{c,e,2}$ and $\sigma \in \Sigma_{c,e,2} \setminus \Sigma_{c,e,1}$ can be argued similarly with appropriate modifications.

Case 4: $(\sigma \in \Sigma_{c,d,1} \cap \Sigma_{c,d,2})$ Since $s \in \overline{K}$ and $s\sigma \notin \overline{K}$,

$$\begin{aligned}&[(\forall i \in \{1,2\})[(P_i^{-1}P_i(s) \cap \overline{K})\sigma \cap \mathscr{L}(G) \nsubseteq \overline{K}]]\\ \Rightarrow\ &[\sigma \notin S_{gdec}]\\ \Rightarrow\ &[s\sigma \notin \mathscr{L}(S_{gdec}/G)]\end{aligned}$$

This contradicts the assumption that $s\sigma \in \mathscr{L}(S_{gdec}/G)$.

The Cases 5 and 6 where $\sigma \in \Sigma_{c,d,1} \setminus \Sigma_{c,d,2}$ and $\sigma \in \Sigma_{c,d,2} \setminus \Sigma_{c,d,1}$ can be argued similarly with appropriate modifications. ■

Theorem 9 implies that every partition of the set of controllable events satisfying $\Sigma_{ter}(\mathscr{L}_m(M(\Sigma_c))) \subseteq \Sigma_{c,d}$, together with the corresponding local decision rules given in (7), guarantees a safe closed-loop behavior when S_{gdec} controls G, even if K is not co-observable.

In view of Theorem 9, one may wish to compare the closed-loop behaviors corresponding to different partitions. Let $\Sigma_{c,d}^1$, $\Sigma_{c,d}^2$, $\Sigma_{c,e}^1$, $\Sigma_{c,e}^2 \in 2^{\Sigma_c}$. Suppose that $\Sigma_{ter}(\mathscr{L}_m(M(\Sigma_c))) \subseteq \Sigma_{c,d}^1, \Sigma_{c,d}^2$ and the following partition conditions hold:

$$\Sigma_{c,d}^1 \,\dot{\cup}\, \Sigma_{c,e}^1 = \Sigma_c \quad \text{and} \quad \Sigma_{c,d}^2 \,\dot{\cup}\, \Sigma_{c,e}^2 = \Sigma_c$$

Let us denote the generalized supervisors following the supervision rule (7) with the above two partitions by S_{gdec}^1 and S_{gdec}^2, respectively. Then we have the following theorem demonstrating the monotonicity of the closed-loop behaviors w.r.t. partitions.

THEOREM 10 *If* $\Sigma_{ter}(\mathscr{L}_m(M(\Sigma_c))) \subseteq \Sigma_{c,d}^2 \subseteq \Sigma_{c,d}^1$, *then* $\mathscr{L}(S_{gdec}^1/G) \subseteq \mathscr{L}(S_{gdec}^2/G)$.

Proof: We argue this with induction.

(Base of the induction) Since K is nonempty, $\varepsilon \in \mathscr{L}(S_{gdec}^1/G)$ and $\varepsilon \in \mathscr{L}(S_{gdec}^2/G)$.

(Induction hypothesis) Suppose that for all $s \in \mathscr{L}(S_{gdec}^1/G)$ such that $|s| = n$, $s \in \mathscr{L}(S_{gdec}^2/G)$.

(Inductive step) We now prove that for all $\sigma \in \Sigma_c$ and $s \in \mathscr{L}(S^1_{gdec}/G)$ such that $|s| = n$,

$$s\sigma \in \mathscr{L}(S^1_{gdec}/G) \Rightarrow s\sigma \in \mathscr{L}(S^2_{gdec}/G)$$

Case 1: $(\sigma \in \Sigma_{uc})$

$$\begin{aligned} & [\sigma \in \Sigma_{uc}] \wedge [s\sigma \in \mathscr{L}(S^1_{gdec}/G)] \wedge [s \in \mathscr{L}(S^2_{gdec}/G)] \\ \Rightarrow & [s \in \mathscr{L}(S^2_{gdec}/G)] \wedge [s\sigma \in \mathscr{L}(G)] \wedge [\sigma \in S^2_{gdec}(s)] \\ \Rightarrow & [s\sigma \in \mathscr{L}(S^2_{gdec}/G)] \end{aligned}$$

Case 2: $(\sigma \in \Sigma^1_{c,e})$ Since $\Sigma^1_{c,e} \subseteq \Sigma^2_{c,e}$, it is obvious that for all $s \in \Sigma^*$,

$$\sigma \in S^1_{gdec}(s) \Rightarrow \sigma \in S^2_{gdec}(s)$$

Case 3: $(\sigma \in \Sigma^1_{c,d})$ For $\sigma \in \Sigma^1_{c,d} \cap \Sigma^2_{c,d}$,

$$\sigma \in S^1_{gdec}(s) \Rightarrow \sigma \in S^2_{gdec}(s)$$

By Theorem 9, we know that $s \in \overline{K}$. For $\sigma \in \Sigma^1_{c,d} \cap \Sigma^2_{c,e}$,

$$\begin{aligned} & [s \in \overline{K}] \wedge [s\sigma \in \mathscr{L}(S^1_{gdec}/G)] \\ \Rightarrow & [s \in \overline{K}] \wedge [s\sigma \in \mathscr{L}(G)] \wedge [\sigma \in S^1_{gdec}(s)] \\ \Rightarrow & [s \in \overline{K}] \wedge [s\sigma \in \mathscr{L}(G)] \wedge [(\exists i \in \{1,2\})[(P_i^{-1}P_i(s) \cap \overline{K})\sigma \cap \mathscr{L}(G) \subseteq \overline{K}] \wedge [\sigma \in \Sigma_{c,i}]] \\ \Rightarrow & [s \in \overline{K}] \wedge [s\sigma \in \overline{K}] \\ \Rightarrow & [s \in \overline{K}] \wedge [s\sigma \in \overline{K}] \wedge [(\forall i \in \{1,2\})[(P_i^{-1}P_i(s))\sigma \cap \overline{K} \neq \emptyset]] \\ \Rightarrow & [s \in \overline{K}] \wedge [s\sigma \in \mathscr{L}(G)] \wedge [\sigma \in S^2_{gdec}(s)] \\ \Rightarrow & s\sigma \in \mathscr{L}(S^2_{gdec}/G) \end{aligned}$$

■

Combining Theorems 9 and 10 leads to the following result.

COROLLARY 2 *If* $\Sigma_{ter}(\mathscr{L}_m(M(\Sigma_c))) \subseteq \Sigma^2_{c,d} \subseteq \Sigma^1_{c,d}$, *then* $\mathscr{L}(S^1_{gdec}/G) \subseteq \mathscr{L}(S^2_{gdec}/G) \subseteq \overline{K}$.

One may be tempted to infer that the condition $\Sigma_{ter}(\mathscr{L}_m(M(\Sigma_c))) \subseteq \Sigma_{c,d}$ is necessary for safety. However, the following result demonstrates that it is not the case.

PROPOSITION 9 $\mathscr{L}(S_{gdec}/G) \subseteq \overline{K}$ *does not imply* $\Sigma_{ter}(\mathscr{L}_m(M(\Sigma_c))) \subseteq \Sigma_{c,d}$.

Proof: Figure 17(a) and Figure 17(b) represent the uncontrolled behavior and the desired behavior, respectively. Let $\Sigma_c = \Sigma_{c,1} = \Sigma_{c,2} = \Sigma = \{\alpha, \beta\}$ and $\Sigma_o = \Sigma_{o,1} = \Sigma_{o,2} = \emptyset$. The controllability of $\mathscr{L}(H)$ is trivial. Since $\alpha\beta$ and $\beta\alpha$ violate C&P co-

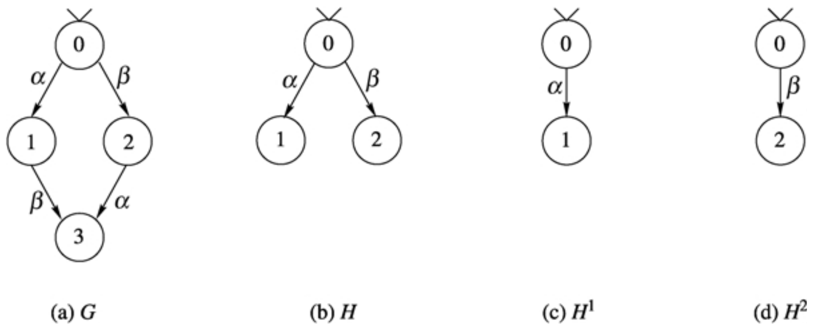


Figure 17. Safety example to illustrate Proposition 9.

observability, the automaton $M(\Sigma_c)$ returns $\Sigma_{ter}(\mathscr{L}_m(M(\Sigma_c))) = \Sigma_c$. However, the settings $\Sigma_{c,d} = \{\beta\}, \Sigma_{c,e} = \{\alpha\}$ and $\Sigma_{c,d} = \{\alpha\}, \Sigma_{c,e} = \{\beta\}$, with the corresponding decoupled local rules given in (7), produce the safe closed-loop languages $\mathscr{L}(H^1)$ and $\mathscr{L}(H^2)$ shown in Figure 17(c) and Figure 17(d), respectively. ■

However, the condition $\Sigma_{ter}(\mathscr{L}_m(M(\Sigma_c))) \subseteq \Sigma_{c,d}$ is critical to guarantee the safety of the closed-loop behavior in the sense that if this condition is violated, the closed-loop behavior may not be safe.

Proposition 10 *If* $\Sigma_{ter}(\mathscr{L}_m(M(\Sigma_c))) \nsubseteq \Sigma_{c,d}$, *then the closed-loop behavior may not be safe.*

Proof: Consider again Figure 17(a) and Figure 17(b) representing the uncontrolled behavior and the desired behavior, respectively, with the same event settings as in the proof of Proposition 9. We know that $\Sigma_{ter}(\mathscr{L}_m(M(\Sigma_c))) = \Sigma_c$. By setting $\Sigma_{c,e} = \Sigma_c$, it is obvious that $\Sigma_{ter}(\mathscr{L}_m(M(\Sigma_c))) \nsubseteq \Sigma_{c,d} = \emptyset$. It is also easy to see that the closed-loop language is $\mathscr{L}(G)$. Therefore, we can say that the closed-loop behavior is not guaranteed to be safe if $\Sigma_{ter}(\mathscr{L}_m(M(\Sigma_c))) \nsubseteq \Sigma_{c,d}$. ■

In view of the above results, we conclude that $\Sigma_{ter}(\mathscr{L}_m(M(\Sigma_c))) = \Sigma_{c,d}$ is the ‘‘optimal’’ partition, in the sense that it generates the largest safe closed-loop behavior among all the partitions guaranteeing the safety of the closed-loop behavior. Roughly speaking, the intuition behind Theorem 10 is that as the local decisions become ‘‘aggressive’’ (even though their fusion rule is conservative),[5] the closed-loop behavior becomes larger. Corollary 2 and Proposition 10 set a limit on ‘‘how aggressive’’ local decisions can be in order to guarantee the safety of the closed-loop behavior.

7.2. Properties of the Synthesized Language

Consider S_{gdec} defined in Section 7.1 with a partition of Σ_c guaranteeing the safety of the closed-loop behavior. Since the local decision rules are decoupled intentionally, it is natural to expect that the closed-loop behavior may not be maximally permissive when S_{gdec} is used. We have the following result.

PROPOSITION 11 *(Non-maximality) In general, $\mathscr{L}(S_{gdec}/G)$ with the ''optimal'' partitioning is not a maximal controllable and co-observable sublanguage of $\overline{K}$.*

Proof: Consider Figure 17(a) and Figure 17(b) representing the uncontrolled behavior and the desired behavior, respectively. Let $\Sigma_c = \Sigma_{c,1} = \Sigma_{c,2} = \Sigma = \{\alpha, \beta\}$ and $\Sigma_o = \Sigma_{o,1} = \Sigma_{o,2} = \emptyset$. The controllability of $\mathscr{L}(H)$ is trivial. Since $\alpha\beta$ and $\beta\alpha$ violate C&P co-observability, the automaton $M(\Sigma_c)$ returns $\Sigma_{ter}(\mathscr{L}_m(M(\Sigma_c))) = \Sigma_c$. By following the ''optimal'' partition strategy $\Sigma_{ter}(\mathscr{L}_m(M(\Sigma_c))) = \Sigma_{c,d}$, we get $\Sigma_{c,d} = \Sigma_c$. With this strategy, the generated language $\mathscr{L}(S_{gdec}/G)$ is $\{\varepsilon\}$. However, there are two maximal controllable and co-observable languages which are generated by the two automata shown in Figure 17(c) and Figure 17(d). Therefore, it can be concluded that $\mathscr{L}(S_{gdec}/G)$ with the ''optimal'' partitioning is not maximal, in general. ■

Since the local decision rule (7) does not consider marking, the following can be demonstrated.

PROPOSITION 12 *(Blocking) In general, $\overline{\mathscr{L}_m(S_{gdec}/G)} \neq \mathscr{L}(S_{gdec}/G)$.*

Proof: Consider the automata G and H shown in Figure 18, representing the uncontrolled behavior and the desired behavior, respectively. Let $\Sigma_c = \Sigma_{c,1} = \Sigma_{c,2} = \{\alpha\}$ and $\Sigma_o = \Sigma_{o,1} = \Sigma_{o,2} = \emptyset$. The controllability of $\mathscr{L}(H)$ is trivial. We can see that $\mathscr{L}_m(H)$ is $\mathscr{L}_m(G)$-closed as well. Since $\gamma\alpha\alpha$ violates C&P co-observability, the automaton $M(\Sigma_c)$ returns $\Sigma_{ter}(\mathscr{L}_m(M(\Sigma_c))) = \{\alpha\}$. Therefore, $\mathscr{L}(S_{gdec}/G) = \{\gamma\}$ with the partition where $\Sigma_{c,d} = \{\alpha\}$. Obviously, $\overline{\mathscr{L}_m(S_{gdec}/G)} \neq \mathscr{L}(S_{gdec}/G)$. ■

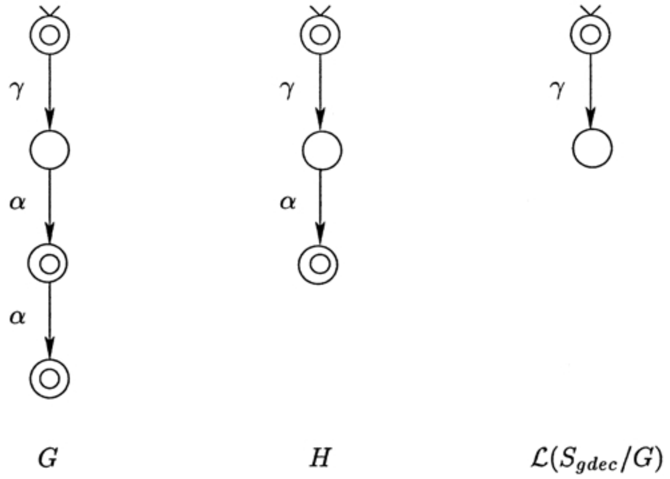


Figure 18. Blocking example.

In the remainder of this section, we consider the properties of two versions of S_{gdec} where the local decision rules are either always antipermissive or always permissive.

A language K is called strongly decomposable if

$$[P_1^{-1}P_1(\overline{K}) \cup P_2^{-1}P_2(\overline{K}) \cup \cdots \cup P_n^{-1}P_n(\overline{K})] \cap \mathscr{L}(G) = \overline{K}$$

This property has been considered as a decentralized version of normality in the sense that it is preserved under union and is a stronger condition than C&P co-observability (Rudie and Wonham, 1992). Since controllability is also preserved under union, the supremal controllable and strongly decomposable sublanguage, denoted by $K^{\uparrow(CD)}$, exists. Let us denote by $K^{\uparrow(CNi)}$ the supremal controllable and normal sublanguage of K, where controllability is w.r.t. $\mathscr{L}(G)$ and Σ_{uc} and normality is w.r.t. $\mathscr{L}(G)$ and $\Sigma_{o,i}$ (hence the superscript "i" in $K^{\uparrow(CNi)}$). We have the following result.

PROPOSITION 13 *For all i,* $K^{\uparrow(CD)} \subseteq K^{\uparrow(CNi)}$.

Proof: Since strong decomposability implies normality w.r.t. $\mathscr{L}(G)$ and $\Sigma_{o,i}$, for all $i, K^{\uparrow(CD)} \subseteq K^{\uparrow(CNi)}$ follows immediately. ■

Let us set $\Sigma_c = \Sigma_{c,d}$ and denote the supervisor following the decision rule (7) with this partition as S_{gdec}^{ap}. This is the most conservative partition according to Corollary 2 and every local decision is based on the antipermissive local decision rule (hence the superscript "ap" in S_{gdec}^{ap}). We have the following inclusion.

PROPOSITION 14 *For all i,* $(\overline{K})^{\uparrow(CNi)} \subseteq \mathscr{L}(S_{gdec}^{ap}/G)$.

Proof: We only prove that $(\overline{K})^{\uparrow(CN1)} \subseteq \mathscr{L}(S_{gdec}^{ap}/G)$ without loss of generality. If $(\overline{K})^{\uparrow(CN1)}$ is empty, inclusion is trivial. Let us suppose that $(\overline{K})^{\uparrow(CN1)}$ is not empty.

(Base of induction) Then $\varepsilon \in (\overline{K})^{\uparrow(CN1)}$ and $\varepsilon \in \mathscr{L}(S_{gdec}^{ap}/G)$ by the definition of $\mathscr{L}(S_{gdec}^{ap}/G)$.

(Induction hypothesis) Suppose that for all $s \in (\overline{K})^{\uparrow(CN1)}$ such that $|s| = n$, $s \in \mathscr{L}(S_{gdec}^{ap}/G)$.

(Induction step) For a contradiction, assume that there exist $s \in (\overline{K})^{\uparrow(CN1)}$ and $\sigma \in \Sigma$ such that $|s| = n$, $s\sigma \in (\overline{K})^{\uparrow(CN1)}$, and $s\sigma \notin \mathscr{L}(S_{gdec}^{ap}/G)$. Since $(\overline{K})^{\uparrow(CN1)} \subseteq \overline{K} \subseteq \mathscr{L}(G)$, we get $s\sigma \in \mathscr{L}(G)$. By the controllability of $\mathscr{L}(S_{gdec}^{ap}/G)$ and the definition of $\mathscr{L}(S_{gdec}^{ap}/G)$,

$$\begin{aligned}
&[s\sigma \in \mathscr{L}(G)] \wedge [s\sigma \notin \mathscr{L}(S_{gdec}^{ap}/G)] \\
&\Rightarrow [\sigma \in \Sigma_c] \wedge [\forall i \in \{1,2\}, \sigma \notin S_{P_i}(P_i(s))] \\
&\Rightarrow [\sigma \in \Sigma_c] \wedge [(\forall i \in \{1,2\})[(P_i^{-1}P_i(s) \cap \overline{K})\sigma \cap \mathscr{L}(G) \not\subseteq \overline{K}]]
\end{aligned}$$

This implies that there exist $s', s'' \in \overline{K}$ such that

$$[P_1(s) = P_1(s'), P_2(s) = P_2(s'')] \wedge [s'\sigma \in \mathscr{L}(G) \setminus \overline{K}] \wedge [s''\sigma \in \mathscr{L}(G) \setminus \overline{K}]$$

However, by normality w.r.t. $\mathscr{L}(G)$ and $\Sigma_{o,1}$, we get

$$s'\sigma \in P_1^{-1}P_1(s\sigma) \cap \mathscr{L}(G) \subseteq (\overline{K})^{\uparrow(CN1)} \subseteq \overline{K}$$

This is the contradiction. ■

In Prosser (1996), the inclusions $(\overline{K})^{\uparrow(CD)} \subseteq \mathscr{L}(S^{ap}_{gdec}/G) \subseteq \overline{K}$ were proved. Note that $(\overline{K})^{\uparrow(CNi)}$ is a suboptimal solution where all control authorities are given to the i-th local supervisor.[6] From Propositions 13 and 14, we see that $(\overline{K})^{\uparrow(CNi)}$ provides a tighter lower bound than $(\overline{K})^{\uparrow(CD)}$. Moreover, Corollary 2 states that $\mathscr{L}(S^{ap}_{gdec}/G)$ is the most conservative language guaranteed to be safe given that the decoupled local control actions following the decision rule (7) are applied.

7.3. Synthesis of the Infimal Controllable and C&P Co-observable Superlanguage

We conclude this section with one last result when the local decision rule is always permissive. A formula for the infimal prefix-closed controllable and C&P co-observable superlanguage is known (Kumar and Shayman, 1998; Rudie and Wonham, 1992). One of the purposes of the computation of this infimal superlanguage, denoted by $(\overline{K})^{\downarrow(CCO)}$, is to realize a supervisor generating this language. We present a simple algorithm synthesizing directly the decoupled local supervisors that result in this language without having to explicitly compute the infimal prefix-closed controllable and C&P co-observable superlanguage. We set $\Sigma_{c,e} = \Sigma_c$ and denote the supervisor following the decision rule (7) with this partition as S^p_{gdec}. This implies the architecture is the conjunctive one and every decision is based on the permissive decision rule. We need following proposition for further argument.

PROPOSITION 15 $\overline{K} \subseteq \mathscr{L}(S^p_{gdec}/G)$.

Proof: (Base of the induction) Since K is a nonempty set, $\varepsilon \in \overline{K}$ and $\varepsilon \in \mathscr{L}(S^p_{gdec}/G)$.
(Induction hypothesis) Suppose that for all $s \in \overline{K}$ such that $|s| = n$, $s \in \mathscr{L}(S^p_{gdec}/G)$.
(Inductive step) We now prove that for all $\sigma \in \Sigma$ and $s \in \overline{K}$ such that $|s| = n$,

$$s\sigma \in \overline{K} \Rightarrow s\sigma \in \mathscr{L}(S^p_{gdec}/G)$$

Case 1: $(\sigma \in \Sigma_{uc})$

$$\begin{aligned}
&[\sigma \in \Sigma_{uc}] \wedge [s\sigma \in \overline{K}] \wedge [s \in \mathscr{L}(S^p_{gdec}/G)]\\
\Rightarrow &[\sigma \in \Sigma_{uc}] \wedge [s\sigma \in \mathscr{L}(G)] \wedge [s \in \mathscr{L}(S^p_{gdec}/G)]\\
\Rightarrow &[(\forall i \in \{1,2\})[\sigma \in S^p_{P_i}(P_i(s))]] \wedge [s\sigma \in \mathscr{L}(G)] \wedge [s \in \mathscr{L}(S^p_{gdec}/G)]\\
\Rightarrow &[s\sigma \in \mathscr{L}(S^p_{gdec}/G)]
\end{aligned}$$

Case 2: $(\sigma \in \Sigma_{c,1} \cap \Sigma_{c,2})$ It is easy to see that

$$\begin{aligned} &[(\forall i \in \{1,2\})[s\sigma \in P_i^{-1}[P_i(s)]\sigma \cap \overline{K}]] \wedge [s \in \mathscr{L}(S_{gdec}^p/G)] \\ \Rightarrow &[(\forall i \in \{1,2\})[\sigma \in S_{P_i}(P_i(s))]] \wedge [s \in \mathscr{L}(S_{gdec}^p/G)] \end{aligned}$$

With $s\sigma \in \overline{K} \subseteq \mathscr{L}(G)$ and the definition of $\mathscr{L}(S_{gdec}^p/G)$, $s\sigma \in \mathscr{L}(S_{gdec}^p/G)$

The cases where $\sigma \in \Sigma_{c,1} \setminus \Sigma_{c,2}$ and $\sigma \in \Sigma_{c,2} \setminus \Sigma_{c,1}$ can be argued similarly with appropriate modifications. ■

We have the following theorem. Note that the controllability assumption on K is not needed for this theorem.

THEOREM 11 $\mathscr{L}(S_{gdec}^p/G) = (\overline{K})^{\downarrow(CCO)}$.

Proof: $(\supseteq)$ Since $\overline{K} \subseteq \mathscr{L}(S_{gdec}^p/G)$ by Proposition 15 and $\mathscr{L}(S_{gdec}^p/G)$ is controllable and C&P co-observable, $(\overline{K})^{\downarrow(CCO)} \subseteq \mathscr{L}(S_{gdec}^p/G)$ by the infimality of $(\overline{K})^{\downarrow(CCO)}$.

$(\subseteq)$ Since $(\overline{K})^{\downarrow(CCO)}$ is controllable and C&P co-observable, there exist S_{P_1} and S_{P_2} such that $\mathscr{L}(S_{P_1} \wedge S_{P_2}/G) = (\overline{K})^{\downarrow(CCO)}$ with local decision rules, for all $i \in \{1,2\}, s \in \Sigma^*$,

$$S_{P_i}(P_i(s)) = \Sigma \setminus \Sigma_{c,i} \cup \{\gamma \in \Sigma_{c,i} : P_i^{-1}P_i(s)\gamma \cap (\overline{K})^{\downarrow(CCO)} \neq \emptyset\}$$

Moreover, by denoting the local decision rule of S_{gdec}^p as $S_{P_i}^p$:

$$S_{P_i}^p(P_i(s)) = \Sigma \setminus \Sigma_{c,i} \cup \{\gamma \in \Sigma_{c,i} : P_i^{-1}P_i(s)\gamma \cap \overline{K} \neq \emptyset\}$$

It is easy to see that for all $i \in \{1,2\}, s \in \Sigma^*$, $\sigma \in \Sigma$,

$$\begin{aligned} &P_i^{-1}P_i(s)\sigma \cap \overline{K} \subseteq P_i^{-1}P_i(s)\sigma \cap (\overline{K})^{\downarrow(CCO)} \\ \Rightarrow &S_{P_i}^p(P_i(s)) \subseteq S_{P_i}(P_i(s)) \\ \Rightarrow &\mathscr{L}(S_{gdec}^p/G) \subseteq \mathscr{L}(S_{P_1} \wedge S_{P_2}/G) = (\overline{K})^{\downarrow(CCO)} \end{aligned}$$

Therefore, $\mathscr{L}(S_{gdec}^p/G) = (\overline{K})^{\downarrow(CCO)}$. ■

Theorem 11 can be interpreted as another characterization of $(\overline{K})^{\downarrow(CCO)}$, namely, as the closed-loop behavior that results from S_{gdec}^p. The discussion in Section 3.1 is directly applicable for the realization of the local supervisors. By adding self-loops for enabled unobservable events at each estimator state, the automata representing the local supervisors can be realized. The automaton generating $(\overline{K})^{\downarrow(CCO)}$ can be constructed by forming the product of these automata with the system model. This method of building a generator of $(\overline{K})^{\downarrow(CCO)}$ is closely related to the modular approach in Kumar and Shayman (1998), since the combination of permissive supervisors by conjunction in the context of

S^p_{gdec} is equivalent to the intersection of languages in Theorem 3 of Kumar and Shayman (1998).

8. Conclusion

In this paper, a generalized form of the conventional decentralized control architecture for discrete-event systems is considered. The general architecture allows combinations between ''fusion by intersection'' and ''fusion by union''. It is shown that a proper combination of fusion rules with corresponding local decision rules (the permissive rule for ''fusion by intersection'' and the antipermissive rule for ''fusion by union'') results in a larger class of achievable languages than those of the conjunctive and disjunctive architectures. A polynomial time verification method of the necessary and sufficient conditions for the existence of supervisors under the general architecture is given. A polynomial time technique for finding an appropriate combination of fusion rules is also presented. Furthermore, we have presented simple ''decoupled'' control policies for the local supervisors and studied their properties. The design of these local supervisors is carried out as if the local supervisors were capable of achieving the desired behavior, namely, as if the desired language were co-observable. Under this technique, we found the ''optimal'' partition of the set of controllable events guaranteeing the safety of the closed-loop behavior. This simple supervisor synthesis technique can also be applied to synthesize local supervisors generating the infimal prefix-closed controllable and C&P co-observable superlanguage.

Appendix

We recall the transition relation of automaton $M(\Sigma_c)$ in Rudie and Wonham (1992).

$$M(\Sigma_c) = \left(Q^{M(\Sigma_c)}, \Sigma, \delta^{M(\Sigma_c)}, q_0^{M(\Sigma_c)}, Q_m^{M(\Sigma_c)}\right)$$

where

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

Let us define the set of conditions implying the violation of C&P co-observability. Note that these conditions are only defined for the controllable events. For $\sigma \in \Sigma_c$,[7]

$$\left.\begin{array}{l}\delta^H(q_1,\sigma) \text{ is defined if } \sigma\in\Sigma_{c,1}\\ \delta^H(q_2,\sigma) \text{ is defined if } \sigma\in\Sigma_{c,2}\\ \delta^H(q_3,\sigma) \text{ is not defined}\\ \delta^G(q_4,\sigma) \text{ is defined}\end{array}\right\}\quad(*)$$

The transition relation $\delta^{M(\Sigma_c)}$ is defined as follows.

For $\sigma\notin\Sigma_{o,1}$ and $\sigma\notin\Sigma_{o,2}$,

$$\delta^{M(\Sigma_c)}((q_1,q_2,q_3,q_4),\sigma)=\begin{cases}(\delta^H(q_1,\sigma),q_2,q_3,q_4)\\ (q_1,\delta^H(q_2,\sigma),q_3,q_4)\\ (q_1,q_2,\delta^H(q_3,\sigma),\delta^G(q_4,\sigma))\\ (\delta^H(q_1,\sigma),\delta^H(q_2,\sigma),\delta^H(q_3,\sigma),\delta^G(q_4,\sigma))\\ d \quad \text{if } (*)\end{cases}$$

For $\sigma\notin\Sigma_{o,1}$ and $\sigma\in\Sigma_{o,2}$,

$$\delta^{M(\Sigma_c)}((q_1,q_2,q_3,q_4),\sigma)=\begin{cases}(\delta^H(q_1,\sigma),q_2,q_3,q_4)\\ (q_1,\delta^H(q_2,\sigma),\delta^H(q_3,\sigma),\delta^G(q_4,\sigma))\\ (\delta^H(q_1,\sigma),\delta^H(q_2,\sigma),\delta^H(q_3,\sigma),\delta^G(q_4,\sigma))\\ d \quad \text{if } (*)\end{cases}$$

For $\sigma\in\Sigma_{o,1}$ and $\sigma\notin\Sigma_{o,2}$,

$$\delta^{M(\Sigma_c)}((q_1,q_2,q_3,q_4),\sigma)=\begin{cases}(q_1,\delta^H(q_2,\sigma),q_3,q_4)\\ (\delta^H(q_1,\sigma),q_2,\delta^H(q_3,\sigma),\delta^G(q_4,\sigma))\\ (\delta^H(q_1,\sigma),\delta^H(q_2,\sigma),\delta^H(q_3,\sigma),\delta^G(q_4,\sigma))\\ d \quad \text{if } (*)\end{cases}$$

For $\sigma\in\Sigma_{o,1}$ and $\sigma\in\Sigma_{o,2}$,

$$\delta^{M(\Sigma_c)}((q_1,q_2,q_3,q_4),\sigma)=\begin{cases}(\delta^H(q_1,\sigma),\delta^H(q_2,\sigma),\delta^H(q_3,\sigma),\delta^G(q_4,\sigma))\\ d \quad \text{if } (*)\end{cases}$$

For $\sigma\in\Sigma$, $\delta^{M(\Sigma_c)}(d,\sigma)$ is undefined.

Acknowledgment

This research is supported in part by NSF grant CCR-0082784 and by the DDR&E MURI on Low Energy Electronics Design for Mobile Platforms and managed by ARO under grant ARO DAAH04-96-1-0377.

Notes

[1] The controllability of $\mathscr{L}_m(H)$ is assumed. This does not affect the polynomial test of solvability since controllability can be verified in polynomial time. We also assume that $\overline{\mathscr{L}_m(H)} = \mathscr{L}(H)$.

[2] The transition relation and the violation condition $(*)$ of the automaton $M(\Sigma_c)$ are recalled, in Appendix.

[3] Given that the number of local supervisors is fixed, the result provides the polynomial test. However, it should be noted that the computational complexity of constructing $M(\Sigma_c)$ and $M_d(\Sigma_c)$ is exponential in the number of local supervisors, n.

[4] This assumption is not restrictive since it is always possible to find the supremal element, $(\overline{K})^{\uparrow(C)}$. Moreover, it is not possible to find any closed-loop language which is in between $(\overline{K})$ and $(\overline{K})^{\uparrow(C)}$ due to the supremality of $(\overline{K})^{\uparrow(C)}$.

[5] Note that ''aggressive'' (permissive) local decision rules are matched with intersection (conservative fusion) and that ''conservative'' (antipermissive) local decision rules are matched with union (aggressive fusion).

[6] Note that the control authorities are distributed throughout the local supervisors under decentralization. The purpose of Proposition 14 is to illustrate the properties of the language $\mathscr{L}(S^{ap}_{gdec}/G)$.

[7] This condition is not mentioned in Rudie and Wonham (1992).

References

Barrett, G. 1999. Modeling, analysis and control of centralized and decentralized logical discrete-event systems. PhD thesis, The University of Michigan.

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

Cho, H., and Marcus, S. I. 1989a. On supremal languages of classes of sublanguages that arise in supervisor synthesis problems with partial observation. *Math. Control Signals Systems* 2: 47–69.

Cieslak, R., Desclaux, C., Fawaz, A. and Varaiya, P. 1988. Supervisory control of discrete event processes with partial observation. *IEEE Trans. on Automat. Contr.* 33(3): 249–260.

Hadj-Alouane, N. B., Lafortune, S., and Lin, F. 1996. Centralized and distributed algorithm for on-line synthesis of maximal control policies under partial observation. *Discrete Event Dynamic Systems: Theory and Applications* 6(41): 379–427.

Heymann, M., and Lin, F. 1994. On-line control of partially observed discrete event systems. *Discrete Event Dynamic Systems: Theory and Applications* 4(3): 221–236.

Hopcroft, J. E. 1979. *Introduction to automata theory, languages, and computation.* Addison-Wesley.

Jiang, S., and Kumar, R. 2000. Decentralized control of discrete event systems with specializations to local control and concurrent systems. *IEEE Transactions on Systems, Man and Cybernetics, Part B* 30(5): 653–660.

Kozak, P., and Wonham, W. M. 1995. Fully decentralized solutions of supervisory control problems. *IEEE Trans. on Automat. Contr.* 40(12): 2094–2097.

Kumar, R., and Shayman, M. A. March 1997. Centralized and decentralized supervisory control of nondeterministic systems under partial observation. *SIAM J. Control Optim.* 35(2): 363–383.

Kumar, R., and Shayman, M. A. 1998. Formulae relating controllability, observability, and co-observability. *Automatica* 34(2): 211–215.

Lamouchi, H., and Thistle, J. 2000. Control of infinite behavior of discrete event systems under partial observations. In *Proc. of CDC 2000, IEEE Conference on Decision and Control* 22–28.

Lin, F., and Wonham, W. M. 1988. Decentralized supervisory control of discrete event systems. *Information Sciences* 44: 199–224.

Lin, F. and Wonham, W. M. 1988. On observability of discrete-event systems. *Information Sciences* 44(3): 173–198.

Prosser, J. H., Kam, M., and Kwatny, H. G. June 1997. Decision fusion and supervisor synthesis in decentralized discrete-event systems. In *Proc. 1997 Ameri. Contr. Conf.* 2251–2255.

Prosser, J. 1996. *Supervisor Synthesis for Partially Observed Discrete-Event Systems*. Ph.D. thesis, Drexel University.

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

Ricker, S. L. 1999. *Knowledge and Communication in Decentralized Discrete-Event Control*. Ph.D. thesis, Queen's University.

Rudie, K., and Willems, J. C. 1995. The computational complexity of decentralized discrete-event control problems. *IEEE Trans. on Automat. Contr.* 40(7): 1313–1318.

Rudie, K., and Willems, W. 1993. IMA preprint series 1105: The computational complexity of decentralized discrete-event control problems. Institute for Mathematics and its Application, www.ima.umn.edu/preprints/MARCH1993/1105.ps.

Rudie, K., and Wonham, W. M. November 1992. Think globally, act locally: Decentralized supervisory control. *IEEE Trans. on Automat. Contr.* 37(11): 1692–1708.

Takai, S. 1998. On the languages generated under fully decentralized supervision. *IEEE Trans. on Automat. Contr.* 43(9): 1253–1256.

Takai, S., and Kodama, S. 1994. Decentralized state feedback control of discrete event systems. *Systems and Control Letters* 22(5): 369–375.

Tripakis, S. 2001. Undecidable problems of decentralized observation and control. In *Proc. of CDC 2001, IEEE Conference on Decision and Control*.

Tsitsiklis, J. N. 1989. On the control of discrete event dynamical systems. *Math. Control Signals Systems* 2(2): 95–107.

Willner, Y., and Heymann, M. 1991. Supervisory control of concurrent discrete-event systems. *International Journal of Control* 54(5): 1143–1169.

Yoo, T.-S. 2002. *Monitoring and Control of Centralized and Decentralized Partially-Observed Discrete-Event Systems*. Ph.D. thesis, The University of Michigan.