SIAM J. CONTROL OPTIM.
Vol. 39, No. 2, pp. 492–511

MAXIMAL SOLUTIONS IN DECENTRALIZED SUPERVISORY CONTROL*

ARD OVERKAMP† AND JAN H. VAN SCHUPPEN‡

Abstract. The decentralized supervisory control problem is to construct for a discrete-event system a set of supervisors each observing only part of the system and each controlling only part of the events such that the interconnection of the system and the supervisors meets control objectives of safety and liveness. Definitions are provided of the concepts of a maximal solution, of a Nash equilibrium, and of a strong Nash equilibrium for a set of supervisors with as order relation the inclusion relation on the set of closed-loop languages. The main result is that a set of supervisors is a maximal solution if and only if it is a strong Nash equilibrium. A procedure to determine a Nash equilibrium is described and illustrated by an example. There is no guarantee that the procedure halts in finite time. However, in the case that it halts in finite time, then it is proven that a Nash equilibrium is obtained.

Key words. discrete-event system, decentralized supervisory control, maximal solution, Nash equilibrium

AMS subject classification. 93C30

PII. S0363012997321139

1. Introduction.

The purpose of this paper is to show how the concept of a Nash equilibrium can be used to obtain maximal solutions of supervisors for decentralized control of discrete-event systems.

Decentralized supervisory control problems arise very naturally in protocol design problems for computer and communication networks but also occur in transportation and manufacturing problems. The network may be modeled as a discrete-event system. The physical separation between the sender and the receiver implies that observations of the operation of the network are available only locally. The problem is then to synthesize a set of controllers in a protocol problem—one at the sender end and one at the receiver end of the communication channel. The interconnection of the network with the supervisors has then to meet control objectives of safety and liveness according to a specification.

Decentralized control is a conceptually difficult problem. Results are available mainly for decentralized control of linear systems and of stochastic systems (see the survey [18]). Fundamental results are partly based on the analogy with game, dynamic game, and team problems. The decentralized supervisory control problem was formulated by R. Cieslak et al. (see [2]), in which the alternating bit protocol is used as an example. The authors presented a necessary and sufficient condition for the existence of a controller for which the closed-loop language equals a specified language. A generalization of this result to the closed-loop language was fit between an upper and lower bound, and an analysis of the set of supervisors was derived by K. Rudie and W. M. Wonham (see [14, 17]). These authors also studied the protocol synthesis problem (see [15, 16]). P. Kozak and W. M. Wonham proposed another solution procedure based on projection of the supremal supervisor (see [4]). Recent work on

*Received by the editors May 7, 1997; accepted for publication (in revised form) February 28, 2000; published electronically September 7, 2000.
http://www.siam.org/journals/sicon/39-2/32113.html

†ORTEC Consultants B.V., P.O. Box 490, 2800 AL Gouda, The Netherlands (aoverkamp@ortec.nl).

‡CWI, P.O. Box 94079, 1090 GB Amsterdam, The Netherlands (J.H.van.Schuppen@cwi.nl).

decentralized supervisory control of nondeterministic systems using prioritized synchronization is presented by R. Kumar and M.A. Shayman (see [5]). The synthesis procedures proposed so far do not satisfy the need of engineering decentralized control problems. The performance of the resulting controllers is in general too conservative.

The approach of this paper is based on an analogy with dynamic game problems. The restriction is imposed to consider a model with only two supervisors. The concept of a maximal solution of a pair of supervisors is defined with respect to the inclusion relation on the set of languages of the closed-loop system. Because the set of pairs of supervisors is a large discrete set, there may be many such pairs. The determination of a maximal pair of supervisors is achieved indirectly. The concept of a strong Nash equilibrium of a pair of supervisors is introduced based on analogy with game theory. It is shown that a pair of supervisors that is a strong Nash equilibrium is also a maximal solution and conversely. A procedure is proposed to compute a strong Nash equilibrium of a pair of supervisors. The procedure is illustrated with an example.

A description of the paper by section follows. Section 2 contains a definition of a discrete-event system that differs slightly from the case usually considered in the literature, the formulation of the decentralized supervisory control problem, and the definition of maximal solution and Nash equilibrium. The result, that a pair of supervisors that is a strong Nash equilibrium is also a maximal solution and conversely, is established in section 3. The procedure for a Nash equilibrium is stated in section 4. Section 5 contains conclusions.

The results of this paper were announced in the conference paper [11] and form part of the thesis [10, Chap. 6] of the first author.

2. Problem formulation.

2.1. Framework. A simple framework will be introduced that allows us to concentrate on the decentralized aspects of the control problem.

Throughout this paper denote the global set of events by Σ, the global set of controllable events by Σ_c, the uncontrolled system by G, and the specification by E. The discrete-event system will be modeled as a finite state automaton with the notation $G = (\Sigma, Q, \delta, q_0)$, with Q the discrete state space, $\delta : \Sigma \times Q \to Q$ the transition function, and q_0 the initial state. For a string $s \in \Sigma^*$ denote by $\bar{s} \subseteq \Sigma^*$ the set of prefixes of this string.

DEFINITION 2.1. *A* supervisor *or* discrete-event controller *is defined by a triple*

$$S = (\Sigma(S),\ \Sigma_c(S),\ \gamma(S)),$$

where

$$\Sigma(S) \subseteq \Sigma,\quad \Sigma_c(S) \subseteq \Sigma(S),\quad \gamma(S) : p_s(\mathrm{L}(G)) \to 2^{\Sigma_c(S)},$$

and p_s is the projection from Σ to $\Sigma(S)$.

Define the controlled language of supervisor S with respect to G or, for short, the language *of S as*

$$\mathrm{L}(S/G) = \{s \in \mathrm{L}(G) : \forall v\sigma \in \bar{s}, \sigma \notin \gamma(S, p_s(v))\}.$$

Note that $\mathrm{L}(S/G) \subseteq \Sigma^*$.

Let $C(\Sigma_a)$ denote the set of all supervisors S with event set $\Sigma(S) = \Sigma_a$ and controllable event set $\Sigma_c(S) = \Sigma_c \cap \Sigma_a$. The function $\gamma(S)$ will be called the control law *of supervisor S. Note that $\gamma(S, s)$ is defined for all $s \in p_s(\mathrm{L}(G))$.*

The control law $\gamma(S)$ maps each trace $s \in p_s(\mathrm{L}(G))$ onto the set of disabled events. In the literature, often the set of enabled events is specified [13]. Both approaches are equivalent.

In the definition above the set of controllable events is taken to be contained in the set of events observable by the supervisor. In general it is possible that a supervisor can influence events it cannot observe. In [10, Sect. 5.2] it is shown how in this situation a control problem can be remodeled such that all controllable events are observable. As that reference is not widely available, the approach is briefly sketched. The idea is based on flags. Controllable, unobservable events are usually implemented with flags. If a flag is set, then the event can execute. If the flag is cleared, then the event is disabled. The plant is remodelled such that it includes the events that set and clear the flags. These so-called flag events are observable and controllable. In this remodeled plant the original events are no longer controllable as they are enabled and disabled via the flag events. If the flag events are, via projection, removed from the language of the remodeled plant, then the language of the original plant is obtained.

Attention will be focused on the decentralized aspects of the supervisory control problem. Marking, nondeterminism, or failure semantics will not be considered. The argument for a simple framework also justifies the restriction to only two supervisors. The authors are confident that in the future the results can be extended to more general frameworks and more supervisors.

The basic supervisory control problem needs to be redefined for the new framework. Note that supervisors, as stated in Definition 2.1, can disable only controllable events. So they are always complete. It is not necessary to add a completeness requirement as is done in [13].

DEFINITION 2.2. *Consider a discrete-event system and a legal language* $L(E) \subset \Sigma^*$. *The basic supervisory control problem (BSCP) is to find a supervisor* S, *such that* $\mathrm{L}(S/G) \subseteq \mathrm{L}(E)$.

Ramadge and Wonham showed that there exists a unique supremal solution to this control problem. This supremal can be effectively computed [13]. It is characterized by a language called the supremal controllable sublanguage contained in $\mathrm{L}(G) \cap \mathrm{L}(E)$. As the notion of controllability will not be used any further, we refer the interested reader to the given reference for more information. The only aspect of controllability that will be used in this chapter is that the supremal controllable language can be effectively computed.

DEFINITION 2.3. *Let* $K^\uparrow$ *be the supremal controllable sublanguage contained in* $\mathrm{L}(G) \cap \mathrm{L}(E)$. *The* supremal supervisor, *denoted by* $S^\uparrow$, *is defined by*

$$\gamma(S^\uparrow, s) = \begin{cases} \{\sigma \in \Sigma_c : s\sigma \in \mathrm{L}(G) \text{ and } s\sigma \notin K^\uparrow\}, & \text{if } s \in K^\uparrow, \\ \emptyset, & \text{otherwise.} \end{cases}$$

It is not difficult to show that $\mathrm{L}(S^\uparrow/G) = K^\uparrow$. *As* $S^\uparrow$ *is supremal it holds for all supervisors* S *which solve the given BSCP, that* $\mathrm{L}(S/G) \subseteq \mathrm{L}(S^\uparrow/G)$.

In this paper it will be assumed that the BSCP is already solved and that the supremal supervisor $S^\uparrow$ is given. It is sufficient to find a supervisor that implements $S^\uparrow$, with respect to the implementation relation defined below. Proposition 2.7 shows that this is a valid approach. A supervisor implements the supremal supervisor if and only if the supervisor solves the BSCP.

DEFINITION 2.4. *Let* S_a, S_b *be two supervisors such that* $\Sigma(S_a) = \Sigma(S_b)$. *Supervisor* S_a implements S_b, *denoted by* $S_a \sqsubseteq S_b$, *if*

$$\gamma(S_b, s) \subseteq \gamma(S_a, s) \quad \forall\, s \in p(\mathrm{L}(S_a/G)),$$

where p is the projection on $\Sigma(S_a) = \Sigma(S_b)$.

Supervisor S_a implements S_b if it disables at least as much as S_b.

LEMMA 2.5. *Let S_a, S_b be two supervisors such that $\Sigma(S_a) = \Sigma(S_b)$.*

$$S_a \sqsubseteq S_b \Rightarrow \mathrm{L}(S_a/G) \subseteq \mathrm{L}(S_b/G).$$

The proof of the preceding lemma and that of Proposition 2.7 are simple and may be found in [10, Chap. 6].

The following example will show why the converse of Lemma 2.5 does not hold.

Example 2.6. Let G be the system such that $\mathrm{L}(G) = \{\varepsilon, \mathtt{a}\}$. Define S_a by $\gamma(S_a, \varepsilon) = \emptyset$ and $\gamma(S_a, \mathtt{a}) = \emptyset$. Define S_b by $\gamma(S_b, \varepsilon) = \emptyset$ and $\gamma(S_a, \mathtt{a}) = \{\mathtt{a}\}$. Then $\mathrm{L}(S_a/G) = \{\varepsilon, \mathtt{a}\} = \mathrm{L}(S_b/G)$, but $\gamma(S_b, \mathtt{a}) \not\subseteq \gamma(S_a, \mathtt{a})$. So $S_a \not\sqsubseteq S_b$. □

In [10, Thm. 2.17] it was shown that in the failure-semantics-based framework a supervisor solves the BSCP if and only if it implements the supremal supervisor. Proposition 2.7 states the same result for the framework of this paper. Because the proof is analogous to that of [10, Thm. 2.17], it is omitted.

PROPOSITION 2.7. *Let the uncontrolled system G, the specification E, and the set of controllable events Σ_c be given. Let $S^\uparrow$ be the supremal supervisor of the BSCP.*

$$\forall S \in \mathcal{C}(\Sigma),\ S \sqsubseteq S^\uparrow \iff \mathrm{L}(S/G) \subseteq \mathrm{L}(S^\uparrow/G) \iff \mathrm{L}(S/G) \subseteq \mathrm{L}(E).$$

In the rest of this paper we will consider control problems that place extra constraints on the supervisor besides the ones given in the BSCP. Proposition 2.7 states that we can first solve the BSCP to get the supremal supervisor $S^\uparrow$. Next we can look for supervisors that satisfy the extra constraints and that implement $S^\uparrow$. In this last step we can concentrate on the extra requirement. As we are mainly interested in the extra requirements imposed by the decentralized nature of the control problem, we will assume that the first step is already solved and that the supremal supervisor $S^\uparrow$ is given.

DEFINITION 2.8. *The* basic supervisory synthesis problem *(BSSP) is to find a supervisor $S \in \mathcal{C}(\Sigma(S^\uparrow))$ such that $S \sqsubseteq S^\uparrow$.*

Often in the literature supervisors are defined as languages instead of control maps. We choose to use control maps as they allow us to divide the control problem into two steps. In the first step the supremal supervisor is synthesized. In this step the controllability condition plays an important role. In the second step we can concentrate on the decentralized aspect of the control problem. Proposition 2.7 shows that we do not have to consider the controllability condition in this step. If supervisors are defined as languages, then also the problem can be divided into two parts. The synthesis problem of the second part is then defined as follows: find a supervisor S such that $\mathrm{L}(S/G) \subseteq \mathrm{L}(S^\uparrow/G)$ and $\mathrm{L}(S/G)$ is controllable. It is necessary to check for controllability, as $\mathrm{L}(S/G) \subseteq \mathrm{L}(S^\uparrow/G)$ does not imply that $\mathrm{L}(S/G)$ is controllable. So in the second step we still have to consider controllability. Using control maps, we can forget about controllability in the second step and concentrate on the decentralized aspects of the control problem. It is not too difficult to adapt the results of this paper to a language-based approach.

2.2. Decentralized supervisory synthesis problem. Up until now we have only looked at supervisors that can observe the whole event set and that enable or disable all controllable events. Now we will look at the decentralized control problem where we have two supervisors, each observing a part of the event set, and each controlling only part of the controllable events (see Figure 1). The two supervisors

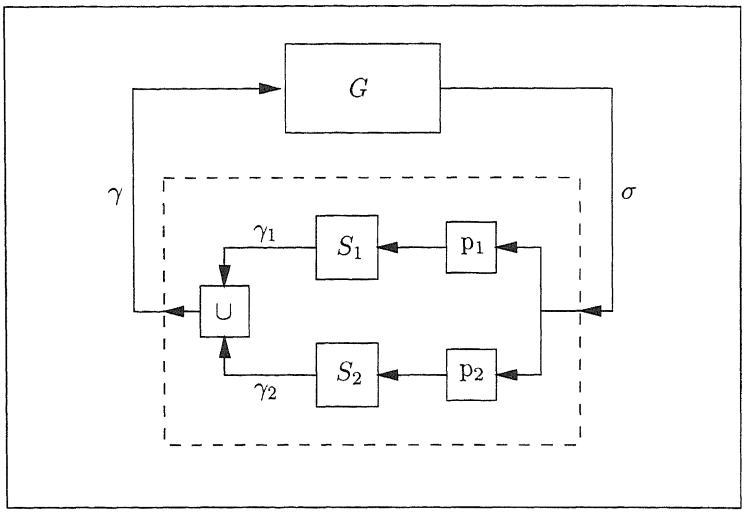


FIG. 1. *The decentralized supervisory control problem.*

together have to control G such that the language of the controlled system is contained in the language of E. Note that the specification is given for the whole controllable system. This is usually referred to as a global specification [14, 17]. If the specification can be decomposed into two local specifications, one for each supervisor, then the decentralized control problem can be reduced to two independent supervisory control problems. In each of these local control problems a single supervisor is synthesized. This control problem has already been solved by F. Lin and W. M. Wonham [7]. In what follows we will assume that the specification is global and cannot be decomposed into local specifications.

As stated before, we will assume the BSCP is already solved and the supremal supervisor $S^{\uparrow}$ is known. By Proposition 2.7 it is sufficient to find a decentralized implementation of $S^{\uparrow}$ to solve the decentralized supervisory control problem.

First it will be defined how two decentralized supervisors co-operate. An event is disabled by the combination of the two supervisors if it is disabled by at least one of them.

DEFINITION 2.9. *Let S_1 and S_2 be two supervisors. The* composition *of S_1 and S_2 is denoted $S_1 \wedge S_2$ and defined by*

$$\Sigma(S_1 \wedge S_2) = \Sigma_1 \cup \Sigma_2,$$
$$\Sigma_c(S_1 \wedge S_2) = \Sigma_c(S_1) \cup \Sigma_c(S_2),$$
$$\gamma(S_1 \wedge S_2, s) = \gamma(S_1, \mathrm{p}_1(s)) \cup \gamma(S_2, \mathrm{p}_2(s)) \qquad \forall\, s \in \mathrm{p}_{1,2}(\mathrm{L}(G)),$$

where p_1 *denotes the projection on* $\Sigma(S_1)$, p_2 *denotes the projection on* $\Sigma(S_2)$, *and* $\mathrm{p}_{1,2}$ *denotes the projection on* $\Sigma(S_1 \wedge S_2)$.

PROPOSITION 2.10. $\mathrm{L}(S_1 \wedge S_2/G) = \mathrm{L}(S_1/G) \cap \mathrm{L}(S_2/G)$.

Proof. The reasoning follows from Definition 2.1.

$$\begin{aligned} & s \in \mathrm{L}(S_1 \wedge S_2/G) \\ \iff & s \in \mathrm{L}(G)\ \forall v\sigma \in \overline{s},\ \sigma \notin \gamma(S_1 \wedge S_2, \mathrm{p}_{1,2}(v)) \\ \iff & s \in \mathrm{L}(G)\ \forall v\sigma \in \overline{s},\ \sigma \notin \gamma(S_1, \mathrm{p}_1(v)),\ \sigma \notin \gamma(S_2, \mathrm{p}_2(v)) \\ \iff & s \in \mathrm{L}(S_1/G),\ s \in \mathrm{L}(S_2/G) \\ \iff & s \in \mathrm{L}(S_1/G) \cap \mathrm{L}(S_2/G). \qquad \square \end{aligned}$$

DEFINITION 2.11. *Consider the discrete-event system specified before and a global specification. Let the supremal supervisor $S^\uparrow$ be given. Let $\Sigma_1, \Sigma_2 \subseteq \Sigma$ be two event sets such that $\Sigma_1 \cup \Sigma_2 = \Sigma$. The decentralized supervisory synthesis problem (DSSP) is to find a pair of supervisors $(S_1, S_2) \in \mathcal{C}(\Sigma_1) \times \mathcal{C}(\Sigma_2)$ such that*

$$S_1 \wedge S_2 \sqsubseteq S^\uparrow.$$

In this definition we made two important assumptions. The one is that $\Sigma_1 \cup \Sigma_2 = \Sigma$. The other is that, according to the definition of $\mathcal{C}(\Sigma_i)$, the set of controllable events of supervisor S_i, $\Sigma_{c,i}$, is equal to $\Sigma_i \cap \Sigma_c$ for $i = 1, 2$.

Consider the case where $\Sigma_1 \cup \Sigma_2 \subsetneqq \Sigma$. If $\Sigma_c \subseteq \Sigma_1 \cup \Sigma_2$, then we can compute the supremal supervisor under partial observation, with observation alphabet $\Sigma_1 \cup \Sigma_2$. See [2, 6] and section [10, Sect. 5.1]. Equivalently to Proposition 2.7, it can be shown that a supervisor implements this supremal supervisor if and only if it solves the control problem under partial observation. We can assume that this control problem is already solved and that the supremal supervisor under partial observation is given. So this control problem can be reduced to the DSSP.

If $\Sigma_c \not\subseteq \Sigma_1 \cup \Sigma_2$, then the control problem can be remodeled in such a way that all controllable events are observable. See [10, Sect. 5.2].

The other assumption is that $\Sigma_{c,i} = \Sigma_i \cap \Sigma_c$, $i = 1, 2$. That is, the controllable events of supervisor S_i are observable by S_i, and an event that is controllable by $S^\uparrow$ and observable by S_i is also controllable by S_i. This is the same constraint as given by Rudie [14, 17] under which decomposability of the closed-loop language is necessary and sufficient for the existence of a decentralized solution. It is argued that in most communication problems these constraints are satisfied. Again, as we want to keep the model simple, we do not consider systems that fail to satisfy this constraint. The authors hope that in the future these constraints can be relaxed.

2.3. Maximal solutions. Traditionally in discrete-event control, supervisors are synthesized that restrict the uncontrolled system as little as possible. A solution is considered optimal if the language of the system controlled by this optimal supervisor is larger than the languages of all other solutions.

DEFINITION 2.12. *Consider the DSSP of Definition* 2.11. *A pair of supervisors $(S_1^\uparrow, S_2^\uparrow) \in \mathcal{C}(\Sigma_1) \times \mathcal{C}(\Sigma_2)$ is called an* optimal decentralized solution *if it is a solution, i.e.,*

$$S_1^\uparrow \wedge S_2^\uparrow \sqsubseteq S^\uparrow, \tag{1}$$

and for all pairs $(S_1, S_2) \in \mathcal{C}(\Sigma_1) \times \mathcal{C}(\Sigma_2)$

$$S_1 \wedge S_2 \sqsubseteq S^\uparrow \Rightarrow \mathrm{L}(S_1 \wedge S_2/G) \subseteq \mathrm{L}(S_1^\uparrow \wedge S_2^\uparrow/G). \tag{2}$$

Recall from [14, 17] the definition of decomposability. A language $K \subseteq \mathrm{L}(G)$ is called *decomposable* if

$$K = \mathrm{p}_1^{-1}(\mathrm{p}_1(K)) \cap \mathrm{p}_2^{-1}(\mathrm{p}_2(K)) \cap \mathrm{L}(G). \tag{3}$$

Rudie showed that, under the given assumptions, there exists a decentralized solution, $(S_1, S_2) \in \mathcal{C}(\Sigma_1) \times \mathcal{C}(\Sigma_2)$, such that the language of the controlled system, $\mathrm{L}(S_1 \wedge S_2/G)$, is equal to a given language $K \subseteq \mathrm{L}(G)$ if and only if K is decomposable. The set of decomposable languages is not closed under arbitrary unions. It is therefore not guaranteed that this set contains a unique supremal element. This implies that

in general the optimal decentralized solution does not exist. There may exist several, mutually incomparable, maximal solutions.

DEFINITION 2.13. *Consider the DSSP of Definition* 2.11. *A pair of supervisors* $(S_1^{\Box}, S_2^{\Box}) \in \mathcal{C}(\Sigma_1) \times \mathcal{C}(\Sigma_2)$ *is called a* maximal decentralized solution *if it is a solution, i.e.,*

$$S_1^{\Box} \wedge S_2^{\Box} \sqsubseteq S^{\uparrow},$$

and there does not exist a pair $(S_1, S_2) \in \mathcal{C}(\Sigma_1) \times \mathcal{C}(\Sigma_2)$ *such that*

$$S_1 \wedge S_2 \sqsubseteq S^{\uparrow} \text{ and } \mathrm{L}(S_1^{\Box} \wedge S_2^{\Box}/G) \subsetneqq \mathrm{L}(S_1 \wedge S_2/G).$$

The set of decomposable languages is closed under arbitrary intersections. It therefore contains a unique infimal element. Rudie posed the following control problem. Given lower bound $\mathrm{L}(A) \subseteq \Sigma^*$ and upper bound $\mathrm{L}(E) \subseteq \Sigma^*$, find a pair $(S_1, S_2) \in \mathcal{C}(\Sigma_1) \times \mathcal{C}(\Sigma_2)$, such that

$$\mathrm{L}(A) \subseteq \mathrm{L}(S_1 \wedge S_2/G) \subseteq \mathrm{L}(E).$$

She showed there exists a solution to this control problem if and only if the infimal decomposable language containing $\mathrm{L}(A)$ is contained in $\mathrm{L}(E)$. Although this infimal is useful to solve the existence question, it often does not give a satisfactory solution. The following example shows that it is in general not trivial to define the lower bound $\mathrm{L}(A)$.

Example 2.14. Consider the alternating bit protocol [14, 17, 19]. This protocol achieves the reliable transmission of messages across an unreliable connection. To achieve this, the sender attaches to each message an extra bit containing either a zero or a one. The protocol can start with either a zero or a one attached to the first message. Consequently, the message with either a one or a zero attached is disabled initially. If the lower bound allows a zero attached to the first message, then the protocol cannot disable this message. It cannot choose the option where a one is attached to the first message. The lower bound $\mathrm{L}(A)$ should allow for both options. Therefore it cannot contain either of the options as this would exclude the other option. The only lower bound that allows both options is the empty language. Unfortunately the infimal decomposable language derived from the empty language does not give a satisfactory solution. See also [10, Sect. 2.5]. □

Another suggestion presented in [14, 17] was to look for the suboptimal solution characterized by the strong decomposability condition. A language $K \subseteq \mathrm{L}(G)$ is called *strongly decomposable* (with respect to Σ_1 and Σ_2) if

$$K = \left(\mathrm{p}_1^{-1}(\mathrm{p}_1(K)) \cup \mathrm{p}_2^{-1}(\mathrm{p}_2(K))\right) \cap \mathrm{L}(G). \tag{4}$$

This condition is closed under arbitrary unions. So the supremal strongly decomposable language exists. Recall from [6] the definition of normality. A language $K \subseteq \mathrm{L}(G)$ is called *normal* (with respect to $\Sigma_o \subseteq \Sigma$) if

$$K = \mathrm{p}_o^{-1}(\mathrm{p}_o(K)) \cap \mathrm{L}(G). \tag{5}$$

Normality of a language K is a sufficient condition for the existence of supervisor that can observe events in Σ_o and that achieves K as language of the controlled system.

PROPOSITION 2.15. *If* $K \subseteq \mathrm{L}(G)$ *is strongly decomposable with respect to* Σ_1 *and* Σ_2, *then* K *is normal with respect to* Σ_1 *and normal with respect to* Σ_2.

Proof. The inclusion $K \subseteq \mathrm{p}_i^{-1}(\mathrm{p}_i(K)) \cap \mathrm{L}(G)$ is satisfied for all languages contained in $\mathrm{L}(G)$. So, it is sufficient to prove $K \supseteq \mathrm{p}_i^{-1}(\mathrm{p}_i(K)) \cap \mathrm{L}(G)$. By the definition of strong decomposability

$$\begin{aligned} K &= \left(\mathrm{p}_1^{-1}(\mathrm{p}_1(K)) \cup \mathrm{p}_2^{-1}(\mathrm{p}_2(K))\right) \cap \mathrm{L}(G) \\ &= \left(\mathrm{p}_1^{-1}(\mathrm{p}_1(K)) \cap \mathrm{L}(G)\right) \cup \left(\mathrm{p}_2^{-1}(\mathrm{p}_2(K)) \cap \mathrm{L}(G)\right) \\ &\supseteq \mathrm{p}_i^{-1}(\mathrm{p}_i(K)) \cap \mathrm{L}(G) \text{ for } i = 1, 2. \qquad \square \end{aligned}$$

The consequence of this proposition is that, if language K is strongly decomposable, then one supervisor, either $S_1 \in \mathcal{C}(\Sigma_1)$ or $S_2 \in \mathcal{C}(\Sigma_2)$, can obtain K as language of the controlled system. The other supervisor is not needed. Obviously, strong decomposability is too strong a restriction for decentralized control problems.

It can be concluded that the existing results for decentralized supervisory control problems do not satisfy the needs from control engineering.

In this paper a characterization of maximal solutions for decentralized control problems will be derived. Is it useful to look for maximal solutions? If a solution is maximal, then this does not imply that it is a good solution. For instance, a maximal solution may allow a lot of unimportant traces and disable all important ones. Another solution which allows less unimportant traces but more important ones may be considered a better solution. However, the authors believe there are some good reasons to investigate the characteristics of maximal solutions. The first and most important reason is that it gives us valuable insight into the fundamental properties of decentralized control problems. This insight may be used to derive algorithms that can synthesize "good" (in whatever sense) solutions, whether they are maximal or not.

Another reason why the authors believe maximality is important is that these "good" solutions will probably be maximal. So, although maximality of a solution does not imply that this solution is useful, a solution that is useful (good in some sense) will most likely be maximal. If a characterization of all maximal solutions can be given, then all "good" solutions will satisfy this characterization. So this characterization limits the class of solutions in which the good ones can be found.

Suppose a solution is given, but it is not fully satisfactory. One can ask the question whether the solution can be extended to obtain a better one. This is possible only if the given solution is not yet maximal. So also in this case a characterization of the maximal solutions will be useful.

2.4. Projections. In [4], Kozak and Wonham propose projections of the supremal supervisor as a solution to the decentralized control or synthesis problem.

DEFINITION 2.16. *The* projection of the supremal supervisor *to event set* $\Sigma_a \subseteq \Sigma(S^\uparrow)$ *is denoted by* $\mathrm{proj}(S^\uparrow, \Sigma_a)$. *It is defined for all* $s_a \in \mathrm{p_a}(\mathrm{L}(G))$ *by*

$$\begin{aligned} &\gamma(\mathrm{proj}(S^\uparrow, \Sigma_a), s_a) \\ &= \left\{\sigma \in \Sigma_c \cap \Sigma_a : \exists s \in \mathrm{p}_a^{-1}(s_a) \cap \mathrm{L}(S^\uparrow/G) \textit{such that } \sigma \in \gamma(S^\uparrow, s)\right\}. \end{aligned}$$

PROPOSITION 2.17 ([4], Lem. 5.1).

$$\mathrm{proj}(S^\uparrow, \Sigma_1) \wedge \mathrm{proj}(S^\uparrow, \Sigma_2) \sqsubseteq S^\uparrow. \tag{6}$$

Kozak and Wonham call $\mathrm{proj}(S^\uparrow, \Sigma_1) \wedge \mathrm{proj}(S^\uparrow, \Sigma_2)$ the *fully decentralized solution.* In general the infimal decomposable solution of Rudie and the projected solution of Kozak and Wonham are incomparable. However, if the given lower bound, $\mathrm{L}(A)$, is

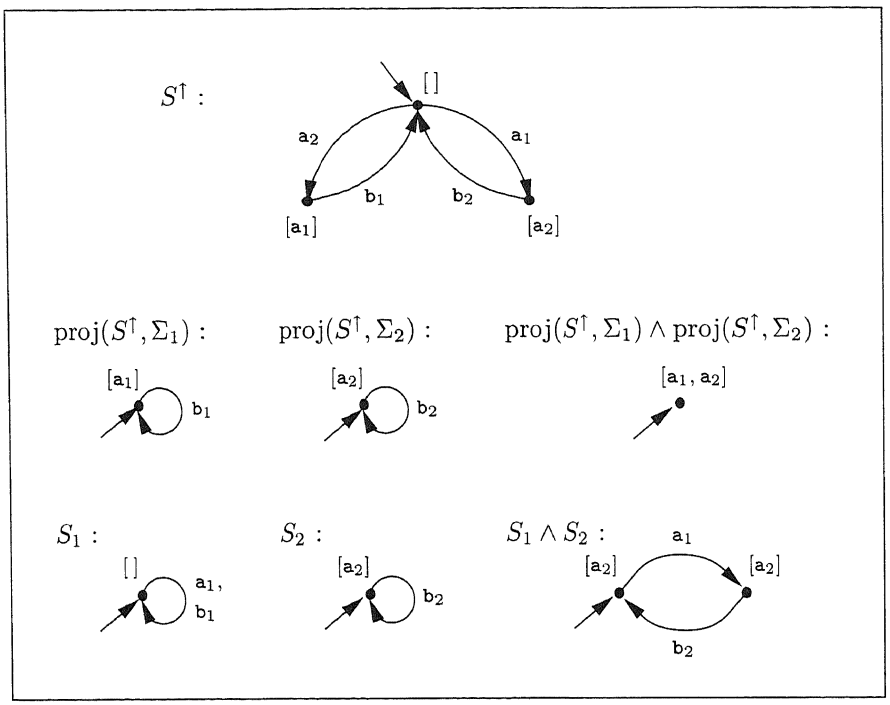


FIG. 2. *The fully decentralized solution is in general not maximal.*

the empty trace, then the projected solution is larger than the infimal decomposable solution. But, even if $\Sigma_1 \cap \Sigma_2 = \emptyset$, the fully decentralized solution is in general not maximal. Consider the following example.

Example 2.18. Consider the supremal supervisor and the fully decentralized solution given in Figure 2. In this example $\Sigma_1 = \{a_1, b_1\}$, $\Sigma_2 = \{a_2, b_2\}$, and $\Sigma_c = \{a_1, a_2\}$. The pair $(\text{proj}(S^\uparrow, \Sigma_1), \text{proj}(S^\uparrow, \Sigma_2))$ is not maximal, because the pair (S_1, S_2) results in a strictly larger controlled language.

Supervisor $\text{proj}(S^\uparrow, \Sigma_1)$ disables event a_1 because the uncontrolled system can execute event a_2, after which event a_1 must be disabled. However, as supervisor S_2 disables a_2 it is not necessary for supervisor S_1 to disable a_1. The pair of supervisors obtained by projection from the supremal supervisor is in general not maximal because the supervisors only take into account the control actions of the supremal supervisor. They do not consider the control law of the other supervisor. In order to obtain a maximal solution it is necessary that the supervisors take into account the control law of the other supervisor. So, to synthesize supervisor S_1 one should already know the control law of supervisor S_2, and to synthesize S_2 one should already know the control law of supervisor S_1. It is this cyclic dependency that makes the synthesis of decentralized controllers such a hard problem.

3. Nash equilibria and maximal solutions. Decentralized stochastic control has been studied extensively. It is related to game and team theory (see [1, 3, 8, 12]). In these fields of research a so-called cost function is used. This cost function maps a decentralized control law to a real number. A solution is considered optimal if it has the lowest cost. Using cost functions, all solutions can be compared with each other. In the field of decentralized supervisory control, solutions are compared by the

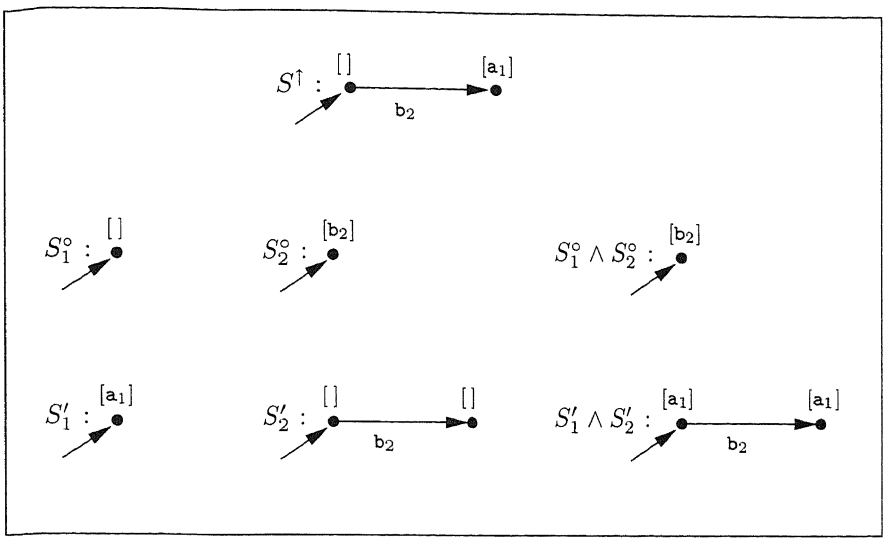


FIG. 3. *The pair* (S_1°, S_2°) *is a Nash equilibrium, yet it is not maximal.*

language of the controlled system. This ordering is not complete. Some solutions may not be comparable.

In game and team theory the notion of Nash equilibrium plays an important role. It will be shown that Nash equilibria are also important for decentralized supervisory control. A pair of supervisors forms a Nash equilibrium if a supervisor cannot improve the controlled language when the other supervisor is kept fixed and conversely.

DEFINITION 3.1. *Consider the DSSP of Definition* 2.11. *A pair of supervisors* $(S_1^\circ, S_2^\circ) \in \mathcal{C}(\Sigma_1) \times \mathcal{C}(\Sigma_2)$ *is called a* Nash equilibrium *if it is a solution, i.e.,*

$$S_1^\circ \wedge S_2^\circ \sqsubseteq S^\uparrow,$$

and

$$\forall S_2 \in \mathcal{C}(\Sigma_2) S_1^\circ \wedge S_2 \sqsubseteq S^\uparrow \Rightarrow \; \mathrm{L}(S_1^\circ \wedge S_2/G) \subseteq \mathrm{L}(S_1^\circ \wedge S_2^\circ/G), \text{ and}$$
$$\forall S_1 \in \mathcal{C}(\Sigma_1) S_1 \wedge S_2^\circ \sqsubseteq S^\uparrow \Rightarrow \; \mathrm{L}(S_1 \wedge S_2^\circ/G) \subseteq \mathrm{L}(S_1^\circ \wedge S_2^\circ/G).$$

In game theory, controllers have conflicting optimization criteria, whereas in team theory all controllers try to optimize the same cost criterion. Note that in the above definition the closed-loop language is analogous to the cost function in a team or game problem. The notion of Nash equilibrium has been introduced in game theory. In team theory it is also known as a person-by-person optimal solution.

In team theory, under certain convexity conditions, a set of controllers is maximal if and only if it is a Nash equilibrium [12]. This equivalence is quite useful because it is relatively easier to determine a Nash equilibrium than a maximum.

The following example shows that for discrete-event systems the Nash equilibrium condition is not sufficient to guarantee maximality.

Example 3.2. Consider the supremal supervisor $S^\uparrow$ and the decentralized implementation (S_1°, S_2°) given in Figure 3. $\Sigma_1 = \{a_1\}$, $\Sigma_2 = \{b_2\}$. All events are controllable. It is not difficult to check that the pair (S_1°, S_2°) is a Nash equilibrium. However, it is not maximal, because the pair (S_1', S_2') is a solution with a strictly larger controlled language. □

For discrete-event systems we need the stronger condition of a strong Nash equilibrium to guarantee maximality of a pair of supervisors.

DEFINITION 3.3. *Consider the DSSP of Definition* 2.11. *A pair of supervisors* $(S_1^\circ, S_2^\circ) \in \mathcal{C}(\Sigma_1) \times \mathcal{C}(\Sigma_2)$ *is called a* strong Nash equilibrium *if it is a Nash equilibrium, and for all* $(S_1, S_2) \in \mathcal{C}(\Sigma_1) \times \mathcal{C}(\Sigma_2)$,

$$\text{(7)} \qquad \mathrm{L}(S_1 \wedge S_2/G) = \mathrm{L}(S_1^\circ \wedge S_2^\circ/G) \Rightarrow (S_1, S_2) \text{ is a Nash equilibrium.}$$

. An intuitive interpretation of the need for the concept of a strong Nash equilibrium follows. The aim of the paper is to obtain a characterization of a maximal decentralized solution in terms of a person-by-person characterization as in game and team theory. Example 3.2 shows that there exists a pair of supervisors that is a Nash equilibrium but not a maximal solution. The condition for a pair of supervisors (S_1°, S_2°) to be a Nash equilibrium is phrased solely in terms of the closed-loop language $L(S_1^\circ \wedge S_2^\circ/G)$. Because of this formulation, it appears that it is necessary that any pair of languages that achieves the same closed-loop language is also a Nash equilibrium. If such a pair was not a Nash equilibrium, one would be able to construct a pair of supervisors with a strictly larger closed-loop language. This in turn would contradict maximality. The next theorem shows that the concept of a strong Nash equilibrium is appropriate.

By Proposition 2.7, $\mathrm{L}(S_1 \wedge S_2/G) = \mathrm{L}(S_1^\circ \wedge S_2^\circ/G)$ and $S_1^\circ \wedge S_2^\circ \sqsubseteq S^\uparrow$ together imply that $S_1 \wedge S_2 \sqsubseteq S^\uparrow$.

THEOREM 3.4. *A pair of supervisors* $(S_1, S_2) \in \mathcal{C}(\Sigma_1) \times \mathcal{C}(\Sigma_2)$ *is maximal if and only if it is a strong Nash equilibrium.*

Proof (strong Nash $\Rightarrow$ Maximal). Assume $(S_1^\circ, S_2^\circ) \in \mathcal{C}(\Sigma_1) \times \mathcal{C}(\Sigma_2)$ is strong Nash but not maximal. Then there exists a pair $(S_1, S_2) \in \mathcal{C}(\Sigma_1) \times \mathcal{C}(\Sigma_2)$ such that $S_1 \wedge S_2 \sqsubseteq S^\uparrow$ and $\mathrm{L}(S_1^\circ \wedge S_2^\circ/G) \subsetneqq \mathrm{L}(S_1 \wedge S_2/G)$. Define $S_1^\square \in \mathcal{C}(\Sigma_1)$ by

$$\gamma(S_1^\square, s_1) = \gamma(S_1^\circ, s_1) \cup \gamma(S_1, s_1) \quad \forall s_1 \in \mathrm{p}_1(\mathrm{L}(G)).$$

We will prove the following points.

1. $\mathrm{L}(S_1^\square/G) = \mathrm{L}(S_1^\circ/G) \cap \mathrm{L}(S_1/G)$,
2. $S_1^\square \wedge S_2^\circ \sqsubseteq S^\uparrow$,
3. $S_1^\square \wedge S_2 \sqsubseteq S^\uparrow$,
4. $\mathrm{L}(S_1^\circ \wedge S_2^\circ/G) = \mathrm{L}(S_1^\square \wedge S_2^\circ/G)$,
5. $\mathrm{L}(S_1^\square \wedge S_2^\circ/G) \subseteq \mathrm{L}(S_1^\square \wedge S_2/G)$.

(Point 1.) This point will be proven by complete induction. The initial step follows from $\varepsilon \in \mathrm{L}(S_1^\square/G)$ and $\varepsilon \in \mathrm{L}(S_1^\circ/G) \cap \mathrm{L}(S_1/G)$. For the inductive step let $s \in \mathrm{L}(S_1^\square/G)$ and $s \in \mathrm{L}(S_1^\circ/G) \cap \mathrm{L}(S_1/G)$. Then

$$\begin{aligned} s\sigma \in \mathrm{L}(S_1^\square/G) &\iff s\sigma \in \mathrm{L}(G),\ \sigma \notin \gamma(S_1^\square, \mathrm{p}_1(s)) \\ &\iff s\sigma \in \mathrm{L}(G),\ \sigma \notin \gamma(S_1^\circ, \mathrm{p}_1(s)),\ \sigma \notin \gamma(S_1, \mathrm{p}_1(s)) \\ &\iff s\sigma \in \mathrm{L}(S_1^\circ/G),\ s\sigma \in \mathrm{L}(S_1/G) \\ &\iff s\sigma \in \mathrm{L}(S_1^\circ/G) \cap \mathrm{L}(S_1/G). \end{aligned}$$

It follows that $\mathrm{L}(S_1^\square/G) = \mathrm{L}(S_1^\circ/G) \cap \mathrm{L}(S_1/G)$.

(Points 2 and 4.) From point 1 and Proposition 2.10, it follows that

$$\begin{aligned}\mathrm{L}(S_1^{\square} \wedge S_2^{\circ}/G) &= \mathrm{L}(S_1^{\circ}/G) \cap \mathrm{L}(S_1/G) \cap \mathrm{L}(S_2^{\circ}/G) \\ &= \mathrm{L}(S_1^{\circ} \wedge S_2^{\circ}/G) \cap \mathrm{L}(S_1/G) \\ &= \big[\text{because } \mathrm{L}(S_1^{\circ} \wedge S_2^{\circ}/G) \subseteq \mathrm{L}(S_1 \wedge S_2/G) \text{ and} \\ &\quad \text{by Proposition 2.10 } \mathrm{L}(S_1 \wedge S_2/G) \subseteq \mathrm{L}(S_1/G)\big] \\ &\quad \mathrm{L}(S_1^{\circ} \wedge S_2^{\circ}/G).\end{aligned}$$

This proves point 4. Point 2 follows from $S_1^{\circ} \wedge S_2^{\circ} \sqsubseteq S^{\uparrow}$ and Proposition 2.7.

(Point 3.) From point 1 and Proposition 2.10, it follows that

$$\begin{aligned}\mathrm{L}(S_1^{\square} \wedge S_2/G) &= \mathrm{L}(S_1^{\circ}/G) \cap \mathrm{L}(S_1/G) \cap \mathrm{L}(S_2/G) \\ &\subseteq \mathrm{L}(S_1/G) \cap \mathrm{L}(S_2/G) = \mathrm{L}(S_1 \wedge S_2/G) \subseteq \mathrm{L}(S^{\uparrow}/G).\end{aligned}$$

So, by Proposition 2.7, $S_1^{\square} \wedge S_2 \sqsubseteq S^{\uparrow}$.

(Point 5.) From point 4, it follows that

$$\begin{aligned}\mathrm{L}(S_1^{\square} \wedge S_2^{\circ}/G) &= \mathrm{L}(S_1^{\circ} \wedge S_2^{\circ}/G) \\ &= \big[\text{because } \mathrm{L}(S_1^{\circ} \wedge S_2^{\circ}/G) \subseteq \mathrm{L}(S_1 \wedge S_2/G)\big] \\ &\quad \mathrm{L}(S_1^{\circ} \wedge S_2^{\circ}/G) \cap \mathrm{L}(S_1 \wedge S_2/G) \\ &= \mathrm{L}(S_1^{\circ}/G) \cap \mathrm{L}(S_2^{\circ}/G) \cap \mathrm{L}(S_1/G) \cap \mathrm{L}(S_2/G) \\ &\subseteq \mathrm{L}(S_1^{\square}/G) \cap \mathrm{L}(S_2/G) = \mathrm{L}(S_1^{\square} \wedge S_2/G).\end{aligned}$$

As the pair $(S_1^{\circ}, S_2^{\circ})$ is strong Nash, it follows from point 4 that $(S_1^{\square}, S_2^{\circ})$ is Nash. So, by point 3 and the definition of Nash, $\mathrm{L}(S_1^{\square} \wedge S_2/G) \subseteq \mathrm{L}(S_1^{\square} \wedge S_2^{\circ}/G)$. Then, from points 4 and 5, $\mathrm{L}(S_1^{\circ} \wedge S_2^{\circ}/G) = \mathrm{L}(S_1^{\square} \wedge S_2^{\circ}/G) = \mathrm{L}(S_1^{\square} \wedge S_2/G)$. As $(S_1^{\circ}, S_2^{\circ})$ is strong Nash, the pair $(S_1^{\square}, S_2)$ is Nash. So

$$\mathrm{L}(S_1 \wedge S_2/G) \subseteq \mathrm{L}(S_1^{\square} \wedge S_2/G) = \mathrm{L}(S_1^{\circ} \wedge S_2^{\circ}/G).$$

But this contradicts our assumption that $\mathrm{L}(S_1^{\circ} \wedge S_2^{\circ}) \subsetneqq \mathrm{L}(S_1 \wedge S_2/G)$. We can conclude that if $(S_1^{\circ}, S_2^{\circ})$ is strong Nash, then it is maximal.

(Maximal $\Rightarrow$ Strong Nash.) Assume $(S_1, S_2) \in \mathcal{C}(\Sigma_1) \times \mathcal{C}(\Sigma_2)$ is maximal but not strong Nash. Then there exists a pair $(S_1', S_2') \in \mathcal{C}(\Sigma_1) \times \mathcal{C}(\Sigma_2)$ such that $\mathrm{L}(S_1' \wedge S_2'/G) = \mathrm{L}(S_1 \wedge S_2/G)$ and (S_1', S_2') is not Nash. So

$$\exists S_1'' \in \mathcal{C}(\Sigma_1) \text{ such that } S_1'' \wedge S_2' \sqsubseteq S^{\uparrow} \text{and } \mathrm{L}(S_1'' \wedge S_2'/G) \not\subseteq \mathrm{L}(S_1' \wedge S_2'/G)$$

or

$$\exists S_2'' \in \mathcal{C}(\Sigma_2) \text{ such that } S_1' \wedge S_2'' \sqsubseteq S^{\uparrow} \text{ and } \mathrm{L}(S_1' \wedge S_2''/G) \not\subseteq \mathrm{L}(S_1' \wedge S_2'/G).$$

Assume, without loss of generality, that such an S_2'' exists. Let $S_2^{\square} \in \mathcal{C}(\Sigma_2)$ be defined by

$$\gamma(S_2^{\square}, s_2) = \begin{cases} \gamma(S_2', s_2) \cap \gamma(S_2'', s_2), & \text{if } s_2 \in \mathrm{p}_2(\mathrm{L}(S_2'/G)) \text{ and } s_2 \in \mathrm{p}_2(\mathrm{L}(S_2''/G)), \\ \gamma(S_2', s_2), & \text{if } s_2 \in \mathrm{p}_2(\mathrm{L}(S_2'/G)) \text{ and } s_2 \notin \mathrm{p}_2(\mathrm{L}(S_2''/G)), \\ \gamma(S_2'', s_2), & \text{if } s_2 \notin \mathrm{p}_2(\mathrm{L}(S_2'/G)) \text{ and } s_2 \in \mathrm{p}_2(\mathrm{L}(S_2''/G)), \\ \Sigma_{2,\mathrm{c}}, & \text{otherwise.} \end{cases}$$

We will prove the following points.

1. $\mathrm{L}(S_2^{\square}/G) = \mathrm{L}(S_2'/G) \cup \mathrm{L}(S_2''/G)$,
2. $S_1' \wedge S_2^{\square} \sqsubseteq S^{\uparrow}$,
3. $\mathrm{L}(S_1' \wedge S_2'/G) \subseteq \mathrm{L}(S_1' \wedge S_2^{\square}/G)$,
4. $\mathrm{L}(S_1' \wedge S_2''/G) \subseteq \mathrm{L}(S_1' \wedge S_2^{\square}/G)$.

(Point 1.) This point will be proven by complete induction. The initial step follows from $\varepsilon \in \mathrm{L}(S_2^{\square}/G)$ and $\varepsilon \in \mathrm{L}(S_2'/G) \cup \mathrm{L}(S_2''/G)$. For the inductive step let $s \in \mathrm{L}(S_2^{\square}/G)$ and $s \in \mathrm{L}(S_2'/G) \cup \mathrm{L}(S_2''/G)$. Trace s can be in one of the three sets $\mathrm{L}(S_2'/G) \cap \mathrm{L}(S_2''/G)$, $\mathrm{L}(S_2'/G) - \mathrm{L}(S_2''/G)$, or $\mathrm{L}(S_2''/G) - \mathrm{L}(S_2'/G)$. If $s \in \mathrm{L}(S_2'/G) \cap \mathrm{L}(S_2''/G)$, then

$$\begin{aligned} s\sigma \in \mathrm{L}(S_2^{\square}/G) &\iff s\sigma \in \mathrm{L}(G) \wedge \sigma \notin \gamma(S_2^{\square}, \mathrm{p}_2(s)) \\ &\iff s\sigma \in \mathrm{L}(G) \wedge \big(\sigma \notin \gamma(S_2', \mathrm{p}_2(s)) \vee \sigma \notin \gamma(S_2'', \mathrm{p}_2(s))\big) \\ &\iff s\sigma \in \mathrm{L}(S_2'/G) \vee s\sigma \in \mathrm{L}(S_2''/G) \\ &\iff s\sigma \in \mathrm{L}(S_2'/G) \cup \mathrm{L}(S_2''/G). \end{aligned}$$

If $s \in \mathrm{L}(S_2'/G)$ but $s \notin \mathrm{L}(S_2''/G)$, then

$$\begin{aligned} s\sigma \in \mathrm{L}(S_2^{\square}/G) &\iff s\sigma \in \mathrm{L}(G) \wedge \sigma \notin \gamma(S_2^{\square}, \mathrm{p}_2(s)) \\ &\iff s\sigma \in \mathrm{L}(G) \wedge \sigma \notin \gamma(S_2', \mathrm{p}_2(s)) \\ &\iff s\sigma \in \mathrm{L}(S_2'/G) \\ &\iff s\sigma \in \mathrm{L}(S_2'/G) \cup \mathrm{L}(S_2''/G). \end{aligned}$$

A similar reasoning holds if $s \in \mathrm{L}(S_2''/G)$ but $s \notin \mathrm{L}(S_2'/G)$. Hence, it follows that $\mathrm{L}(S_2^{\square}/G) = \mathrm{L}(S_2'/G) \cup \mathrm{L}(S_2''/G)$.

(Points 2, 3, and 4.) From point 1 and Proposition 2.10, it follows that

$$\begin{aligned} \mathrm{L}(S_1' \wedge S_2^{\square}/G) &= \mathrm{L}(S_1'/G) \cap \big(\mathrm{L}(S_2'/G) \cup \mathrm{L}(S_2''/G)\big) \\ &= \big(\mathrm{L}(S_1'/G) \cap \mathrm{L}(S_2'/G)\big) \cup \big(\mathrm{L}(S_1'/G) \cap \mathrm{L}(S_2''/G)\big) \\ &= \mathrm{L}(S_1' \wedge S_2'/G) \cup \mathrm{L}(S_1' \wedge S_2''/G). \end{aligned}$$

This directly proves points 3 and 4. Point 2 follows from $S_1' \wedge S_2' \sqsubseteq S^{\uparrow}$, $S_1' \wedge S_2'' \sqsubseteq S^{\uparrow}$, and Proposition 2.7.

As (S_1, S_2) is maximal, so is (S_1', S_2'). Then, by point 3, $\mathrm{L}(S_1' \wedge S_2'/G) = \mathrm{L}(S_1' \wedge S_2^{\square}/G)$. From point 4 it follows that $\mathrm{L}(S_1' \wedge S_2''/G) \subseteq \mathrm{L}(S_1' \wedge S_2'/G)$. But this contradicts our assumption that $\mathrm{L}(S_1' \wedge S_2''/G) \not\subseteq \mathrm{L}(S_1' \wedge S_2'/G)$. Hence it can be concluded that if (S_1, S_2) is maximal, then it is strong Nash. □

Consider two pairs of supervisors to be *control equivalent* if their controlled languages are equal, or

$$(S_1, S_2) \equiv (S_3, S_4) \in \mathcal{C}(\Sigma_1) \times \mathcal{C}(\Sigma_2) \text{ if } L(S_1 \wedge S_2/G) = L(S_3 \wedge S_4/G).$$

Then a pair of supervisors is maximal if and only if all control equivalent pairs are Nash equilibria. Let the *control equivalence class* corresponding with the language $K \subseteq \mathrm{L}(G)$ be the set of pairs for which the controlled language is equal to K. A prefix closed and decomposable language can be considered maximal if and only if all pairs in its corresponding control equivalence class are Nash equilibria.

If the event sets Σ_1 and Σ_2 are disjoint, then a weaker condition can be found to characterize maximal solutions. Define $(\widehat{S}_1, \widehat{S}_2)$ as the pair of most restrictive supervisors in the control equivalence class of (S_1, S_2).

DEFINITION 3.5. *Let* $(S_1, S_2) \in \mathcal{C}(\Sigma_1) \times \mathcal{C}(\Sigma_2)$. *The supervisor* $\widehat{S}_1 \in \mathcal{C}(\Sigma_1)$ *is defined by*

$$\gamma(\widehat{S}_1, s_1) = \{\sigma \in \Sigma_c(S_1) : s_1\sigma \notin \mathrm{p}_1(\mathrm{L}(S_1 \wedge S_2/G))\} \quad \forall s_1 \in \mathrm{p}_1(\mathrm{L}(G)).$$

The supervisor $\widehat{S}_2 \in \mathcal{C}(\Sigma_2)$ *is defined analogously.*

Supervisor $\widehat{S}_1$ can be seen as the most restrictive supervisor of all supervisors S_1' for which there exists a supervisor S_2' such that $\mathrm{L}(S_1' \wedge S_2'/G) = \mathrm{L}(S_1 \wedge S_2/G)$. That is, if such an S_1' disables event σ after trace s, then $s\sigma$ is not an element of $\mathrm{L}(S_1'/G) \supseteq \mathrm{L}(S_1' \wedge S_2'/G) = \mathrm{L}(S_1 \wedge S_2/G)$. So $\widehat{S}_1$ will also disable this event.

First it needs to be proven that $(\widehat{S}_1, \widehat{S}_2)$ is a solution and that it is control equivalent with (S_1, S_2).

PROPOSITION 3.6. *Let* $(S_1, S_2) \in \mathcal{C}(\Sigma_1) \times \mathcal{C}(\Sigma_2)$ *be a decentralized solution implementing* $S^{\uparrow}$, *and let* $(\widehat{S}_1, \widehat{S}_2)$ *be defined as above. Then*

1. $\widehat{S}_1 \wedge \widehat{S}_2 \sqsubseteq S^{\uparrow}$, *and*
2. $\mathrm{L}(\widehat{S}_1 \wedge \widehat{S}_2/G) = \mathrm{L}(S_1 \wedge S_2/G)$.

Proof (point 2, $\mathrm{L}(\widehat{S}_1 \wedge \widehat{S}_2/G) \subseteq \mathrm{L}(S_1 \wedge S_2/G)$). First we will prove by induction that $\mathrm{L}(\widehat{S}_1/G) \subseteq \mathrm{L}(S_1/G)$. The initial step follows from $\varepsilon \in \mathrm{L}(\widehat{S}_1/G)$ and $\varepsilon \in \mathrm{L}(S_1/G)$. For the inductive step let $s \in \mathrm{L}(\widehat{S}_1/G)$ and $s \in \mathrm{L}(S_1/G)$.

$$\begin{aligned}
s\sigma \in \mathrm{L}(\widehat{S}_1/G) &\Rightarrow s\sigma \in \mathrm{L}(G) \ \wedge \ \sigma \notin \gamma(\widehat{S}_1, \mathrm{p}_1(s)) \\
&\Rightarrow s\sigma \in \mathrm{L}(G) \ \wedge \ \big(\sigma \notin \Sigma_c(S_1) \vee \mathrm{p}_1(s)\sigma \in \mathrm{p}_1(\mathrm{L}(S_1 \wedge S_2/G))\big) \\
&\Rightarrow \big[\text{because } \mathrm{L}(S_1 \wedge S_2/G) \subseteq \mathrm{L}(S_1/G)\big] \\
&\quad\ s\sigma \in \mathrm{L}(G) \ \wedge \ \big(\sigma \notin \Sigma_c(S_1) \vee \mathrm{p}_1(s)\sigma \in \mathrm{p}_1(\mathrm{L}(S_1/G))\big) \\
&\Rightarrow s\sigma \in \mathrm{L}(G) \ \wedge \ \big(\sigma \notin \Sigma_c(S_1) \vee \sigma \notin \gamma(S_1, \mathrm{p}_1(s))\big) \\
&\Rightarrow \big[\text{because } \sigma \notin \Sigma_c(S_1) \Rightarrow \sigma \notin \gamma(S_1, \mathrm{p}_1(s))\big] \\
&\quad\ s\sigma \in \mathrm{L}(G) \ \wedge \ \sigma \notin \gamma(S_1, \mathrm{p}_1(s)) \\
&\Rightarrow s\sigma \in \mathrm{L}(S_1/G).
\end{aligned}$$

By symmetry it follows that $\mathrm{L}(\widehat{S}_2/G) \subseteq \mathrm{L}(S_2/G)$. So

$$\mathrm{L}(\widehat{S}_1 \wedge \widehat{S}_2/G) = \mathrm{L}(\widehat{S}_1/G) \cap \mathrm{L}(\widehat{S}_2/G) \subseteq \mathrm{L}(S_1/G) \cap \mathrm{L}(S_2/G) = \mathrm{L}(S_1 \wedge S_2/G).$$

(Point 2, $\mathrm{L}(S_1 \wedge S_2/G) \subseteq \mathrm{L}(\widehat{S}_1 \wedge \widehat{S}_2/G)$.) First it will be proven by induction that $\mathrm{L}(S_1 \wedge S_2/G) \subseteq \mathrm{L}(\widehat{S}_1/G)$. The initial step follows from $\varepsilon \in \mathrm{L}(S_1 \wedge S_2/G)$ and $\varepsilon \in \mathrm{L}(\widehat{S}_1/G)$. For the inductive step let $s \in \mathrm{L}(S_1 \wedge S_2/G)$ and $s \in \mathrm{L}(\widehat{S}_1/G)$.

$$\begin{aligned}
&s\sigma \in \mathrm{L}(S_1 \wedge S_2/G) \\
&\Rightarrow \sigma \notin \Sigma_1 \ \vee \ \big(\sigma \in \Sigma_1 \wedge \mathrm{p}_1(s\sigma) = \mathrm{p}_1(s)\sigma \in \mathrm{p}_1(\mathrm{L}(S_1 \wedge S_2/G))\big) \\
&\Rightarrow \big[\text{by construction of } \gamma(\widehat{S}_1, \mathrm{p}_1(s))\big] \\
&\quad\ \sigma \notin \Sigma_1 \ \vee \ (\sigma \in \Sigma_1 \wedge \sigma \notin \gamma(\widehat{S}_1, \mathrm{p}_1(s))) \\
&\Rightarrow \big[\text{because } \gamma(\widehat{S}_1, \mathrm{p}_1(s)) \subseteq \Sigma_1\big] \ \ \sigma \notin \gamma(\widehat{S}_1, \mathrm{p}_1(s)) \\
&\Rightarrow \big[\text{because } s \in \mathrm{L}(\widehat{S}_1/G) \text{ and } s\sigma \in \mathrm{L}(G)\big] \ \ s\sigma \in \mathrm{L}(\widehat{S}_1/G).
\end{aligned}$$

By symmetry it follows that $\mathrm{L}(S_1 \wedge S_2/G) \subseteq \mathrm{L}(\widehat{S}_2/G)$. So

$$\mathrm{L}(S_1 \wedge S_2/G) \subseteq \mathrm{L}(\widehat{S}_1/G) \cap \mathrm{L}(\widehat{S}_2/G) = \mathrm{L}(\widehat{S}_1 \wedge \widehat{S}_2/G).$$

(Point 1.) This follows directly from point 2 and Proposition 2.7. □

THEOREM 3.7. *Let* $\Sigma_1 \cap \Sigma_2 = \emptyset$. *Let* $(S_1, S_2) \in \mathcal{C}(\Sigma_1) \times \mathcal{C}(\Sigma_2)$. *Let* $(\widehat{S}_1, \widehat{S}_2) \in \mathcal{C}(\Sigma_1) \times \mathcal{C}(\Sigma_2)$ *be defined by Definition* 3.5. *Then* (S_1, S_2) *is maximal if and only if* $(\widehat{S}_1, \widehat{S}_2)$ *is a Nash equilibrium.*

Proof (Maximal $\Rightarrow$ Nash). If (S_1, S_2) is maximal, then by Theorem 3.4 (S_1, S_2) is a strong Nash equilibrium, which by points 1 and 2 of Proposition 3.6 implies that $(\widehat{S}_1, \widehat{S}_2)$ is a Nash equilibrium.

(Nash $\Rightarrow$ Maximal.) Assume $(\widehat{S}_1, \widehat{S}_2)$ is a Nash equilibrium, but (S_1, S_2) is not maximal. Then, by point 2 of Proposition 3.6 $(\widehat{S}_1, \widehat{S}_2)$ is not maximal. There exists a pair $(S_1', S_2') \in \mathcal{C}(\Sigma_1) \times \mathcal{C}(\Sigma_2)$ such that $S_1' \wedge S_2' \sqsubseteq S^\uparrow$ and $\mathrm{L}(\widehat{S}_1 \wedge \widehat{S}_2/G) \subsetneq \mathrm{L}(S_1' \wedge S_2'/G)$. We will first prove that

$$\widehat{S}_1 \wedge S_2' \sqsubseteq S^\uparrow \text{ and } S_1' \wedge \widehat{S}_2 \sqsubseteq S^\uparrow.$$

It will be proven by induction that $\mathrm{L}(\widehat{S}_1/G) \subseteq \mathrm{L}(S_1'/G)$. The initial step follows from $\varepsilon \in \mathrm{L}(\widehat{S}_1/G)$ and $\varepsilon \in \mathrm{L}(S_1'/G)$. For the inductive step let $s \in \mathrm{L}(\widehat{S}_1/G)$ and $s \in \mathrm{L}(S_1'/G)$.

$$\begin{aligned}
s\sigma \in \mathrm{L}(\widehat{S}_1/G) &\Rightarrow s\sigma \in \mathrm{L}(G) \ \wedge \ \sigma \notin \gamma(\widehat{S}_1, \mathrm{p}_1(s)) \\
&\Rightarrow s\sigma \in \mathrm{L}(G) \ \wedge \ (\sigma \notin \Sigma_{\mathrm{c}}(S_1) \vee \mathrm{p}_1(s)\sigma \in \mathrm{p}_1(\mathrm{L}(S_1 \wedge S_2/G))) \\
&\Rightarrow [\text{because } \mathrm{L}(S_1 \wedge S_2/G) \subseteq \mathrm{L}(S_1' \wedge S_2'/G) \subseteq \mathrm{L}(S_1'/G)] \\
&\quad s\sigma \in \mathrm{L}(G) \ \wedge \ (\sigma \notin \Sigma_{\mathrm{c}}(S_1) \vee \mathrm{p}_1(s)\sigma \in \mathrm{p}_1(\mathrm{L}(S_1'/G))) \\
&\Rightarrow s\sigma \in \mathrm{L}(G) \ \wedge \ (\sigma \notin \Sigma_{\mathrm{c}}(S_1) \vee \sigma \notin \gamma(S_1', \mathrm{p}_1(s))) \\
&\Rightarrow [\text{because } \sigma \notin \Sigma_{\mathrm{c}}(S_1) \Rightarrow \sigma \notin \gamma(S_1', \mathrm{p}_1(s))] \\
&\quad s\sigma \in \mathrm{L}(G) \ \wedge \ \sigma \notin \gamma(S_1', \mathrm{p}_1(s)) \\
&\Rightarrow s\sigma \in \mathrm{L}(S_1'/G).
\end{aligned}$$

It follows that $\mathrm{L}(\widehat{S}_1/G) \subseteq \mathrm{L}(S_1'/G)$. Now

$$\begin{aligned}
\mathrm{L}(\widehat{S}_1 \wedge S_2'/G) &= \mathrm{L}(\widehat{S}_1/G) \cap \mathrm{L}(S_2'/G) \subseteq \mathrm{L}(S_1'/G) \cap \mathrm{L}(S_2'/G) \\
&= \mathrm{L}(S_1' \wedge S_2'/G) \subseteq \mathrm{L}(S^\uparrow/G).
\end{aligned}$$

So, by Proposition 2.7, $\widehat{S}_1 \wedge S_2' \sqsubseteq S^\uparrow$. It follows by symmetry that $S_1' \wedge \widehat{S}_2 \sqsubseteq S^\uparrow$.

As $\mathrm{L}(\widehat{S}_1 \wedge \widehat{S}_2/G) \subsetneq \mathrm{L}(S_1' \wedge S_2'/G)$ there exists a trace $s \in \mathrm{L}(S_1' \wedge S_2'/G)$ such that $s \notin \mathrm{L}(\widehat{S}_1 \wedge \widehat{S}_2/G)$. Let $v\sigma$ be the prefix of s such that $\sigma \in \Sigma$, $v \in \mathrm{L}(\widehat{S}_1 \wedge \widehat{S}_2/G)$, and $v\sigma \notin \mathrm{L}(\widehat{S}_1 \wedge \widehat{S}_2/G)$. Assume without loss of generality that $\sigma \in \Sigma_2$. Then, by the assumption that $\Sigma_1 \cap \Sigma_2 = \emptyset$, $\sigma \notin \Sigma_1$. So $\sigma \notin \gamma(\widehat{S}_1, \mathrm{p}_1(v)) \subseteq \Sigma_1$. Thus $v\sigma \in \mathrm{L}(\widehat{S}_1/G)$. As $v\sigma \in \mathrm{L}(S_1' \wedge S_2'/G) \subseteq \mathrm{L}(S_2'/G)$, it follows that $v\sigma \in \mathrm{L}(\widehat{S}_1 \wedge S_2'/G)$. But this contradicts the fact that $(\widehat{S}_1, \widehat{S}_2)$ is a Nash equilibrium. Hence we can conclude that if $(\widehat{S}_1, \widehat{S}_2)$ is a Nash equilibrium, then (S_1, S_2) is maximal. □

A prefix closed and decomposable language $K \subseteq \mathrm{L}(G)$ can be considered maximal if and only if the pair of most restricting supervisors in the control equivalence class corresponding with language K is a Nash equilibrium.

4. Construction of Nash equilibria. Theorems 3.4 and 3.7 give characterizations of the maximal solutions in terms of Nash equilibria. However, they do not state how Nash equilibria can be obtained. For dynamic games in the field of game and team theory, a necessary condition for a Nash equilibrium can be given by the coupled Bellman–Hamilton–Jacobi equations. A solution to these equations is under certain additional conditions also sufficient for a Nash equilibrium. A procedure for the construction of a solution is known [9]. It alternately keeps one of the controllers fixed and tries to optimize the other. At each iteration only one of the controllers is optimized. For dynamic games it is not guaranteed that the procedure converges. And if it converges, it is not guaranteed that it does so in a finite number of steps.

For supervisory control the Bellman–Hamilton–Jacobi equations are not applicable. Yet, the procedure can still be used. At each iteration one of the supervisors is kept fixed and the other is optimized. Only one supervisor is synthesized in each step. This can be seen as a supervisory control problem for a single supervisor. The combination of the fixed supervisor and the uncontrolled system is taken as the uncontrolled system for this control problem. As only one supervisor is synthesized (and all controllable events are observable) a unique optimal solution exists. In the next iteration this optimal supervisor is taken fixed and the other supervisor is optimized. This procedure is repeated until the pair of supervisors remains invariant. Below this procedure is formalized.

Assume without loss of generality that S_1 is the supervisor which is kept fixed. Consider the supervisory control problem with partial observations for the plant S_1/G and with legal language $L(S^\uparrow)$. Then define

$$K = \sup \left\{ \begin{array}{l} K' \subseteq L(S_1/G) | (1)\ K' \subseteq L(S^\uparrow), \\ (2)\ K' \text{ controllable with respect to } (L(S_1/G), \Sigma_c(\Sigma_2)), \\ (3) \text{ and normal with respect to } L(S_1/G) \text{ and } p_2 \end{array} \right\}. \tag{8}$$

The supremal supervisor S_2 with respect to the uncontrolled system S_1/G is defined by

$$\gamma(S_2, s_2) = \{\sigma \in \Sigma_c(S_2) : s_2\sigma \notin \mathrm{p}_2(K)\} \quad \forall s_2 \in \mathrm{p}_2(\mathrm{L}(G)). \tag{9}$$

If S_2 is kept fixed, then S_1 is computed analogously. The formula for K in this case is obtained from that of (8) by interchanging the indices 1 and 2.

LEMMA 4.1. *Let S_1 be the supervisor which is kept fixed. Let S_2 and K be as defined above. Then* $\mathrm{L}(S_1 \wedge S_2/G) = K$.

Proof. The proof will be by complete induction. As $\varepsilon \in \mathrm{L}(S_1 \wedge S_2/G)$ and $\varepsilon \in K$ the initial step is satisfied. For the inductive step let $s \in \mathrm{L}(S_1 \wedge S_2/G)$ and $s \in K$.

$$\begin{aligned} s\sigma \in \mathrm{L}(S_1 \wedge S_2/G) &\iff s\sigma \in \mathrm{L}(S_1/G) \wedge \sigma \notin \gamma(S_2, \mathrm{p}_2(s)) \\ &\iff s\sigma \in \mathrm{L}(S_1/G) \ \wedge \ (\sigma \notin \Sigma_c(\Sigma_2) \vee \mathrm{p}_2(s)\sigma \in \mathrm{p}_2(K)) \\ &\iff [\text{because } s \in K \text{ and } K \text{ is controllable}] \\ &\qquad s\sigma \in \mathrm{L}(S_1/G) \ \wedge \ (s\sigma \in K \vee \mathrm{p}_2(s)\sigma \in \mathrm{p}_2(K)) \\ &\iff [\text{because } K \subseteq \mathrm{p}_2^{-1}(\mathrm{p}_2(K))] \\ &\qquad s\sigma \in \mathrm{L}(S_1/G) \wedge s\sigma \in \mathrm{p}_2^{-1}(\mathrm{p}_2(K)) \\ &\iff [\text{because } K \text{ is normal with respect to } \mathrm{L}(S_1/G)] \\ &\qquad s\sigma \in K. \qquad \square \end{aligned}$$

The procedure is described by the following steps.

PROCEDURE 4.2.

1. *Choose a pair of most restrictive supervisors* (S_1^0, S_2^0) *as starting point of the procedure. Take, for instance, the pair of most restrictive supervisors corresponding with the fully decentralized solution. Let* $j = 0$.
2. *If* j *is even, then let* S_2^{j+1} *be the supremal supervisor with respect to uncontrolled system* S_1^j/G *and event set* Σ_2. *Let* $S_1^{j+1} = S_1^j$. *If* j *is odd, then let* S_1^{j+1} *be the supremal supervisor with respect to uncontrolled system* S_2^j/G *and event set* Σ_1. *Let* $S_2^{j+1} = S_2^j$.
3. *If* $(S_1^{j+1}, S_2^{j+1}) \neq (S_1^j, S_2^j)$, *then increment* j *and continue with step* 2.

First it will be shown that all pairs of supervisors (S_1^j, S_2^j) are most restricting.

LEMMA 4.3. *Let* $j \in N$ *and assume that* (S_1^j, S_2^j) *is most restrictive. Then* (S_1^{j+1}, S_2^{j+1}) *obtained in the second step of the procedure is also most restrictive.*

Proof. Assume without loss of generality that j is odd. So $S_2^{j+1} = S_2^j$ and S_1^{j+1} is the supremal supervisor with respect to S_2^j/G. Let $K^j = \mathrm{L}(S_1^j \wedge S_2^j/G)$ and $K^{j+1} = \mathrm{L}(S_1^{j+1} \wedge S_2^{j+1}/G)$. Comparing (9) with Definition 3.5, it is not difficult to see that S_1^{j+1} is most restrictive with respect to K^{j+1}. Supervisor $S_2^{j+1} = S_2^j$ is most restrictive with respect to language K^j. It remains to show that it is most restrictive with respect to K^{j+1}.

$$\begin{aligned} &\sigma \in \gamma(S_2^{j+1}, s_2) = \gamma(S_2^j, s_2) \Rightarrow \sigma \in \Sigma_{\mathrm{c}}(S_2^j) \wedge s_2\sigma \notin \mathrm{L}(S_2^j/G) \\ &\Rightarrow [\text{because } K^{j+1} \subseteq \mathrm{L}(S_2^j/G)]\ \sigma \in \Sigma_{\mathrm{c}}(S_2^j) \wedge s_2\sigma \notin K^{j+1}. \end{aligned}$$

As K^{j+1} is supremal it follows that $K^j \subseteq K^{j+1}$.

$$\begin{aligned} \sigma \notin \gamma(S_2^{j+1}, s_2) = \gamma(S_2^j, s_2) \Rightarrow &\left[\text{because } S_2^j \text{ is most restrictive with respect to } K^j\right] \\ &\sigma \notin \Sigma_{\mathrm{c}}(S_2^j) \vee s_2\sigma \in K^j \\ \Rightarrow &\left[\text{because } K^j \subseteq K^{j+1}\right]\ \sigma \notin \Sigma_{\mathrm{c}}(S_2^j) \vee s_2\sigma \in K^{j+1}. \end{aligned}$$

It follows that S_2^{j+1} is most restrictive with respect to K^{j+1}. And thus (S_1^{j+1}, S_2^{j+1}) is most restrictive. □

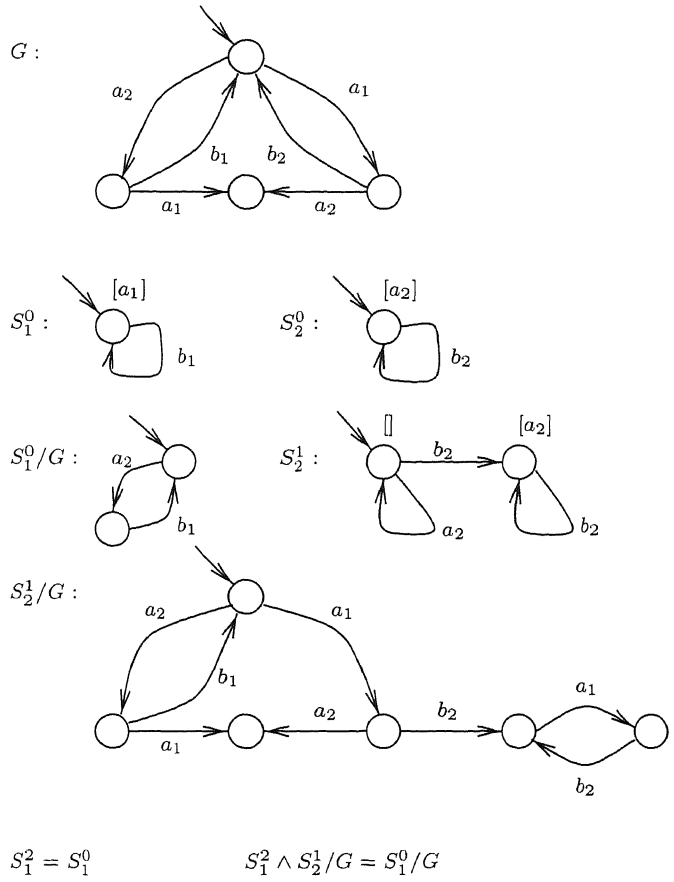
Next it will be shown that if $(S_1^{j+1}, S_2^{j+1}) = (S_1^j, S_2^j)$, then (S_1^j, S_2^j) forms a Nash equilibrium. So if Σ_1 and Σ_2 are disjoint, then this pair is a maximal solution.

THEOREM 4.4. *Let* $j \in \mathbb{N}$ *and let* $S_1^j, S_2^j, S_1^{j+1}, S_2^{j+1}$ *be constructed by the procedure above. If* $(S_1^{j+1}, S_2^{j+1}) = (S_1^j, S_2^j)$, *then* (S_1^j, S_2^j) *forms a Nash equilibrium.*

Proof. Assume without loss of generality that j is odd. Then, according to the second step of the procedure, $S_2^{j+1} = S_2^j$ and S_1^{j+1} is the supremal supervisor with respect to S_2^j/G. As $S_1^{j+1} = S_1^j$ it follows that S_1^j is optimal if S_2^j is kept fixed. This proves the first part of the Nash equilibrium condition.

From the previous iteration of the procedure it follows that $S_1^j = S_1^{j-1}$ and that S_2^j is the supremal supervisor with respect to S_1^{j-1}/G. In the next iteration supervisor S_2^{j+2} will be synthesized. Supervisor S_2^{j+2} is the optimal solution with respect to $S_1^{j+1}/G = S_1^j/G = S_1^{j-1}/G$. So S_2^{j+2} will be equal to S_2^j. Supervisor S_2^j is optimal if S_1^j is kept fixed. This proves the second part of the Nash equilibrium condition. And thus (S_1^j, S_2^j) is a Nash equilibrium. □

Example 4.5. Consider the system described in Example 2.18 and Figure 2. Take the pair of most restrictive supervisors corresponding with the fully decentralized solution as starting point of the procedure. In this case $S_1^0 = \mathrm{proj}(S^\uparrow, \Sigma_1)$ and $S_2^0 = \mathrm{proj}(S^\uparrow, \Sigma_2)$. Let $\Sigma_1 = \{\mathtt{a}_1, \mathtt{b}_1\}$, $\Sigma_2 = \{\mathtt{a}_2, \mathtt{b}_2\}$, and $\Sigma_{\mathrm{c}} = \{\mathtt{a}_1, \mathtt{a}_2\}$. Note that



$S_1^2 = S_1^0$ $\qquad$ $S_1^2 \wedge S_2^1/G = S_1^0/G$

FIG. 4. *Construction of a Nash equilibrium pair of supervisors for Example* 4.5.

the event sets Σ_1 and Σ_2 are disjoint. The construction of the Nash equilibrium is summarized in Figure 4. In this construction use is made of the expression for the control law of (9). First S_1^0 is kept fixed and the optimal supervisor S_2^1 with respect to the uncontrolled system S_1^0/G is derived. Note that $L(S_1^0/G) \subseteq L(S^\uparrow)$ and thus by (8) $K = L(S_1^0/G)$.

Next, S_2^1 is kept fixed. The language $L(S_2^1/G) \not\subseteq L(S^\uparrow)$, so K is a proper subset of $L(S_2^1/G)$. The optimal supervisor S_1^2 with respect to the uncontrolled system S_2^1/G is derived. It turns out that $S_1^2 = S_1^0$. In subsequent steps the pair of supervisors remains invariant. The pair (S_1^2, S_2^1) is thus a Nash equilibrium, and therefore, according to Theorem 3.7, a maximal solution. The closed-loop system according to this Nash equilibrium is identical to S_1^0/G.

Example 4.6. Now, consider a slight alteration of the control problem of Example 4.5. Let $\Sigma_c = \Sigma$ and let the rest be unchanged. Take, as before, the pair of most restrictive supervisors corresponding with the fully decentralized solution as a starting point. In this case also the b-events are disabled. The construction of the pair of supervisors is then illustrated in Figure 5.

The procedure will converge to the limit pair (S_1^*, S_2^*). However, this solution will not be obtained in a finite number of steps.

The example shows that a small change in the parameters of the problem may

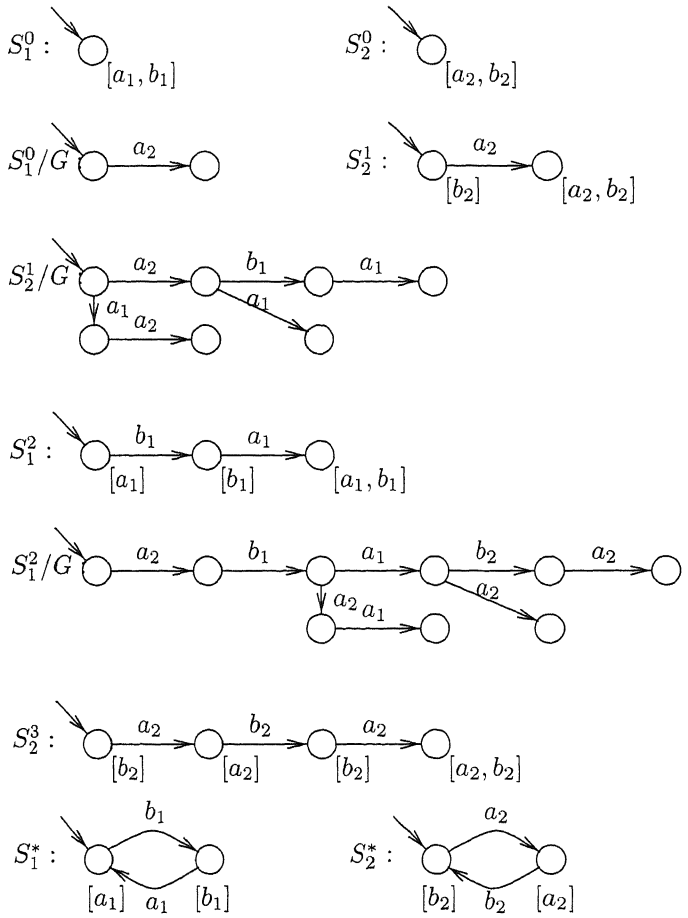


FIG. 5. *Construction of a Nash equilibrium pair of supervisors for Example* 4.6.

lead to a different solution. It may even cause the procedure to become nonhalting.

Up until now these particularities are not fully understood. Further research is needed to adapt the algorithm such that it always converges in a finite number of steps. Also, further research is required to understand the relationship between the initial parameters and the eventual solution. For decentralized control of finite-dimensional linear systems, there is an example for which the procedure does not stop after a finite number of steps and for which a decentralized controller is infinite-dimensional (see [18]). For concrete decentralized supervisory control problems a few steps of the procedure may yield a useful pair of supervisors.

It would be ideal if the procedure could produce a representation of all maximal solutions. It is not certain whether such a representation is finite.

5. Conclusions. For decentralized supervisory control a pair of supervisors is defined to be a maximal solution if there does not exist another such pair with a strictly larger closed-loop language. It has been argued that a maximal solution is of interest to control synthesis of decentralized discrete-event systems. The construction of a maximal solution is handled by analogy with game and team problems. A pair of supervisors is defined to be a Nash equilibrium if, when one supervisor is kept fixed,

the other cannot be changed so as to enlarge the closed-loop language and conversely. The main result is then that a pair of supervisors is a strong Nash equilibrium if and only if it is a maximal solution.

A procedure is presented for the construction of a strong Nash equilibrium of a pair of supervisors. The procedure alternatingly keeps one supervisor fixed and solves a supervisory control problem for the other supervisor. The procedure is shown to work on an example. Another example establishes that the procedure may not stop after any finite number of steps.

Major open questions are (1) the classification of all maximal solutions for the decentralized supervisory control problem, and (2) conditions under which Procedure 4.2 stops after a finite number of steps. Experience should be gained with this approach to decentralized supervisory control.

REFERENCES

[1] T. BASAR AND G. OLSDER, *Dynamic Noncooperative Game Theory*, Academic Press, New York, London, 1982.

[2] R. CIESLAK, C. DESCLAUX, A. FAWAZ, AND P. VARAIYA, *Supervisory control of discrete-event processes with partial observations*, IEEE Trans. Automat. Control, 33 (1988), pp. 249–260.

[3] Y. HO, *Team decision theory and information structures*, Proc. IEEE, 68 (1980), pp. 644–654.

[4] P. KOZAK AND W. M. WONHAM, *Fully Decentralized Solutions of Supervisory Control Problems*, report 9310, Systems and Control Group, Department of Electrical Engineering, University of Toronto, Toronto, Canada, 1993.

[5] R. KUMAR AND M. A. SHAYMAN, *Centralized and decentralized supervisory control of nondeterministic systems under partial observation*, SIAM J. Control Optim., 35 (1997), pp. 363–383.

[6] F. LIN AND W. M. WONHAM, *Decentralized control and coordination of discrete-event systems*, in Proceedings of the 27th IEEE Conference on Decision and Control, Austin, TX, IEEE, New York, 1988, pp. 1125–1130.

[7] F. LIN AND W. M. WONHAM, *Decentralized control and cooperation of discrete event systems with partial observations*, IEEE Trans. Automat. Control, 35 (1990), pp. 1330–1337.

[8] J. NASH, *Non-cooperative games*, Ann. of Math. (2), 54 (1951), pp. 286–295.

[9] G. OLSDER, *Comments on a numerical procedure for the solution of differential games*, IEEE Trans. Automat. Control, 20 (1975), pp. 704–705.

[10] A. OVERKAMP, *Discrete Event Control Motivated by Layered Network Architectures*, Ph.D. thesis, University of Groningen, Groningen, The Netherlands, 1996.

[11] A. OVERKAMP AND J. VAN SCHUPPEN, *A characterization of maximal solutions for decentralized discrete event control problems*, in Proceedings of the International Workshop on Discrete Event Systems, Edinburgh, Scotland, UK, IEE, London, UK, 1996, pp. 278–283.

[12] R. RADNER, *Team decision problems*, Ann. Math. Statist., 33 (1962), pp. 857–881.

[13] P. RAMADGE AND W. M. WONHAM, *The control of discrete event systems*, Proc. IEEE, 77 (1989), pp. 81–98.

[14] K. RUDIE, *Decentralized Control of Discrete-Event Systems*, Ph.D. thesis, Department of Electrical Engineering, University of Toronto, Toronto, Canada, 1992.

[15] K. RUDIE AND W. M. WONHAM, *Supervisory control of communicating processes*, in Protocol Specification, Testing and Verification X, L. Logrippo, R. Probert, and H. Ural, eds., North-Holland, Amsterdam, 1990, pp. 243–257.

[16] K. RUDIE AND W. M. WONHAM, *Protocol verification using discrete-event systems*, in Proceedings of the 31st IEEE Conference on Decision and Control, New York, NY, 1992, pp. 3770–3777.

[17] K. RUDIE AND W. M. WONHAM, *Think globally, act locally: Decentralized supervisory control*, IEEE Trans. Automat. Control, 37 (1992), pp. 1692–1708.

[18] N. SANDELL JR., P. VARAIYA, M. ATHANS, AND M. SAFONOV, *Survey of decentralized control methods for large scale systems*, IEEE Trans. Automat. Control, 23 (1978), pp. 108–128.

[19] A. TANENBAUM, *Computer Networks*, Prentice-Hall International, London, UK, 1981.