Available at
www.ElsevierComputerScience.com
POWERED BY SCIENCE DIRECT
Information Processing Letters 90 (2004) 21–28
Information Processing Letters
www.elsevier.com/locate/ipl

Undecidable problems of decentralized observation and control on regular languages ☆

Stavros Tripakis

VERIMAG—C.N.R.S., Centre Equation, 2, avenue de Vignate, 38610 Gières, France

Received 18 January 2003; received in revised form 27 November 2003

Communicated by P.M.B. Vitanyi

Abstract

We introduce a decentralized observation problem, where the system under observation is modeled as a regular language L over a finite alphabet Σ and n subsets of Σ model distributed observation points. A regular language $K \subseteq L$ models a set of distinguished behaviors, say, correct behaviors of the system. The objective is to check the existence of a function which, given the n observations corresponding to a behavior $\rho \in L$, decides whether ρ is in K or not. We prove that checking the existence of such a function is undecidable. We then use this result to show undecidability of a decentralized supervisory control problem in the discrete event system framework.

Keywords: Distributed systems; Observation; Control; Synthesis; Undecidability

1. Introduction

In this paper we report negative (undecidability) results on two decentralized synthesis problems. The first is a problem of observation and the second is a problem of control.

The observation problem is as follows. We consider a plant modeled as a regular language L over some finite alphabet Σ. A letter in Σ is seen as an event generated by the plant and a finite word in L is seen as a behavior of the plant. We are also given a regular language $K \subseteq L$, modeling a set of distinguished behaviors in L. For example, behaviors in K may be considered "good" in some sense, while those in $L - K$ are "bad". The plant is observed at n distributed points, each with partial observation capabilities. More precisely, we assume that only a subset of events, $\Sigma_i \subseteq \Sigma$, can be observed at point i. For each behavior generated by the plant, the n separate observations are gathered in a (centralized) decision point. This point must determine whether the original behavior was in K or not.

Obviously, such a decision cannot always be made. For example, if $\Sigma = \{a, b\}$, $L = \{ab, ba\}$, $K = \{ab\}$, $n = 2$, $\Sigma_1 = \{a\}$ and $\Sigma_2 = \{b\}$, then based only on the observation pair (a, b), it cannot be determined whether ab or ba happened. When a decision can be made, we say that K is observable with respect to

☆ This work has been supported in part by European IST projects "Next TTA" under project No. IST-2001-32111 and "RISE" under project No. IST-2001-38117, and by CNRS STIC project "CORTOS".

E-mail address: tripakis@imag.fr.

0020-0190/$ – see front matter
doi:10.1016/j.ipl.2004.01.004

L and $(\Sigma_1, \ldots, \Sigma_n)$. The observation problem is to check observability.

Our first negative result is that the observation problem is undecidable in the decentralized case $(n \geqslant 2)$. Our proof reduces an instance of Post's Correspondence Problem to an observation problem for $n = 2$. We also give a necessary and sufficient condition for observability, and prove that observability is closed under union, intersection and relative complement (w.r.t. L).

We then consider a decentralized control problem, from the theory of *supervisory control for discrete event systems* [15]. Again there is a plant modeled as a regular language G over Σ. The plant is controlled by n supervisors. Supervisor i observes a subset of events $\Sigma_i \subseteq \Sigma$ and controls a subset of events $\Gamma_i \subseteq \Sigma$. The type of control that supervisor i can exercise is enable or disable the events it controls. A specification is given as a regular language $E \subseteq \Sigma^*$. The objective of the supervisors is to restrict the behaviors in G so that the language of the controlled plant is contained in E. To avoid trivial solutions (e.g., where the supervisors disable all controllable events at all times), we require that supervisors are *non-blocking*, which informally means that any "incomplete" behavior of the controlled plant can be extended to a "complete" (i.e., accepted) behavior. The control problem is to check existence of such supervisors.

Our second main result is that an observation problem for a given n can be reduced to a control problem for the same n. This implies undecidability of the decentralized control problem $(n \geqslant 2)$. The reduction is probably not surprising, since in every control problem there is an observation part. The only subtlety is that in the control problem there is no communication between the supervisors, whereas in the observation problem all distributed observations are communicated to a single decision point. To overcome this a priori difference of frameworks, we construct a plant that allows supervisor i to "transmit" its observations to supervisor j. This is done by supervisor i enabling the right events and supervisor j observing them. Thus, although communication between supervisors is not explicitly allowed in the control problem, it can be implicitly captured through an appropriate model of the plant.

The rest of the paper is organized as follows. In Section 3 we define the decentralized observation problem, state some of its properties and prove undecidability. In Section 4 we define the decentralized control problem and show how it can capture the observation problem. In Section 2 we discuss related work and future directions.

2. Related work

The observation problem we study in this paper has been introduced in [18]. There, it is compared to (and shown that it is different from) various decentralized observability notions that have appeared in the literature [10,3,17,6], serving mainly as necessary conditions for the corresponding control problems. Different decentralized observation problems have been also introduced in a context of fault detection [4]. Tsitsiklis and Athans [21] study a number of decentralized decision making problems where the set of possible observations is finite (thus, problems are decidable) and show that such problems are "inherently difficult" (NP-hard).

Our observation problem is related to the theory of rational relations (for instance, see [1]). In fact, an alternative proof of Theorem 3.1 can be obtained using the undecidability of the recognizability problem for rational relations. Here, we have opted for the more direct reduction of PCP, first, because this is the proof we originally found and, second, because it requires no background on rational relations.

Undecidability results on the synthesis of distributed reactive systems are reported in [13]. There, the authors consider a framework more general than ours, in the sense that the topology of the decentralized architecture is not fixed, but is a parameter to the problem. Other differences include the fact that their specifications are written in linear temporal logic (LTL) and that the plant is not explicitly modeled. The undecidability proof is done by a reduction of the halting problem for deterministic Turing machines.

In the context of discrete event systems (DES), Lamouchi and Thistle [8] report undecidability of a control problem similar to ours, with the difference that they consider an infinite-word framework and ω-regular specifications. Their proof is also based, as in [13], on a reduction of the halting problem for deterministic Turing machines.

Positive results on decentralized control synthesis problems also exist. The distributed reactive module synthesis problem is shown to be decidable for hierarchical (acyclic) architectures in [13]. This result is extended in [7] to CTL* specifications and some cyclic architectures, and in [11] to local specifications. In [12] it is shown that a decentralized controller synthesis problem is decidable for all architectures, provided: (1) specifications are trace-closed ω-regular languages, (2) the controllers are required to depend only on the length of their observation (and not the observation itself), and (3) certain restrictions on the communication policy between observers are imposed. It is also shown that the problem becomes undecidable as soon as one of the above three conditions is lifted. The undecidability proofs are based, as in [13], on reductions of the halting problem for deterministic Turing machines.

Different variants of the decentralized control problem under a fixed architecture are shown to be decidable in [10,3,22,17,6]. For instance, the problem is decidable when the language of the controlled plant is required to be *equal to*—instead of contained in—a specification language, or when the controlled language is prefix-closed. For complexity results, see, for instance, [20,16,2]. Necessary and/or sufficient conditions for existence of controllers in such a setting are given in a number of papers, for instance, see [3,17,9].

3. A decentralized observation problem

Let Σ be a finite alphabet. Σ^* denotes the set of all finite words over Σ, ϵ denotes the empty word and $\Sigma^+ = \Sigma^* - \{\epsilon\}$. Given $\rho, \rho' \in \Sigma^*$, $\rho \cdot \rho'$ (or $\rho\rho'$) is the concatenation of ρ and ρ'. Given sets $A, B \subseteq \Sigma^*$, $A \cdot B$ (or AB) is the set $\{\rho \cdot \rho' \mid \rho \in A,\ \rho' \in B\}$. We follow the notation of regular expressions and write $A \cdot a$, where $A \subseteq \Sigma^*$, $a \in \Sigma$, instead of $A \cdot \{a\}$. We also write $A + B$ for the union $A \cup B$. A word $\sigma \in \Sigma^*$ is a prefix of $\rho \in \Sigma^*$ if there exists $\tau \in \Sigma^*$ such that $\rho = \sigma\tau$. $\mathrm{Pref}(\rho)$ denotes the set of prefixes of ρ. Given $A \subseteq \Sigma^*$, $\mathrm{Pref}(A) = \bigcup_{\rho \in A} \mathrm{Pref}(\rho)$. $\mathrm{Pref}(A)$ is called the prefix-closure of A. Note that $A \subseteq \mathrm{Pref}(A)$. If $\mathrm{Pref}(A) = A$ then A is said to be prefix-closed.

Given $\rho \in \Sigma^*$ and $\Gamma \subseteq \Sigma$, the projection of ρ onto Γ, denoted $P_\Gamma(\rho)$, is the word $\pi \in \Gamma^*$ obtained from ρ by erasing all letters not in Γ. For example, if $\Sigma = \{a, b, c\}$ and $\Gamma = \{a, c\}$, then $P_\Gamma(abbcbacb) = acac$.

Definition 3.1 (*Decentralized observation problem*). Given finite alphabet Σ, subalphabets $\Sigma_i \subseteq \Sigma$, for $i = 1, \dots, n$, and regular languages $K \subseteq L \subseteq \Sigma^*$, does there exist a total function $f : \Sigma_1^* \times \dots \times \Sigma^* \to \{0, 1\}$, such that

$$\forall \rho \in L \quad \bigl(\rho \in K \Leftrightarrow f\bigl(P_{\Sigma_1}(\rho), \dots, P_{\Sigma_n}(\rho)\bigr) = 1\bigr).$$

If such a function f exists, we say that K is observable with respect to L and $(\Sigma_1, \dots, \Sigma_n)$.

Notice that if Σ_i is empty, then $P_{\Sigma_i}(\rho) = \epsilon$, for any $\rho \in \Sigma^*$. Thus, the ith observation does not provide any information. From now on, we will be assuming that all Σ_i are non-empty.

Lemma 3.1 (Necessary and sufficient condition). *K is observable with respect to L and $(\Sigma_1, \dots, \Sigma_n)$ iff*

$$\forall \rho \in K,\ \forall \rho' \in L - K,\ \exists i \in \{1, \dots, n\},\quad P_{\Sigma_i}(\rho) \neq P_{\Sigma_i}(\rho'). \tag{1}$$

Proof. Assume the negation of Condition (1), that is, assume there exist $\rho \in K$ and $\rho' \in L - K$, such that $\forall i \in \{1, \dots, n\}$, $P_{\Sigma_i}(\rho) = P_{\Sigma_i}(\rho')$. Let $\sigma_i = P_{\Sigma_i}(\rho) = P_{\Sigma_i}(\rho')$, for $i = 1, \dots, n$. Then, $f(\sigma_1, \dots, \sigma_n)$ must equal both 1 (because $\rho \in K$) and 0 (because $\rho' \in L - K$), thus, f cannot exist.

Conversely, assume Condition (1) holds and define, for $(\sigma_1, \dots, \sigma_n) \in \Sigma_1^* \times \dots \times \Sigma_n^*$,

$$f(\sigma_1, \dots, \sigma_n) = \begin{cases} 1, & \text{if } \exists \rho \in K,\ \forall i \in \{1, \dots, n\},\ P_{\Sigma_i}(\rho) = \sigma_i, \\ 0, & \text{otherwise.} \end{cases}$$

We claim that f solves the decentralized observation problem. Indeed, let $\rho \in L$. If $\rho \in K$ then, by definition, $f(P_{\Sigma_1}(\rho), \dots, P_{\Sigma_n}(\rho)) = 1$. If $\rho \notin K$ then we claim that $f(P_{\Sigma_1}(\rho), \dots, P_{\Sigma_n}(\rho)) = 0$. Otherwise, there must exist $\rho' \in K$ such that $\forall i \in \{1, \dots, n\}$, $P_{\Sigma_i}(\rho') = P_{\Sigma_i}(\rho)$, which contradicts Condition (1). □

Consider the following condition:

$$\exists i \in \{1, \dots, n\}, \quad \bigl(P_{\Sigma_i}(K) \cap P_{\Sigma_i}(L - K) = \emptyset\bigr). \tag{2}$$

Condition (2) states that there exists an observer i which alone can distinguish words of K from words of $L - K$ by their projection. This is obviously a sufficient condition for decentralized observability. However, it is not necessary.

Lemma 3.2 (Sufficient condition). *Condition* (2) *is a sufficient condition for decentralized observability. It is not necessary, unless $n = 1$.*

Proof. It can be easily seen that Condition (2) implies Condition (1), thus, the former is sufficient. It can also be seen that, for $n = 1$, the two conditions are equivalent.

It remains to show that Condition (2) is not necessary for $n > 1$. Let $\Sigma = \{a, b, c\}$, $\Sigma_1 = \{a, b\}$, $\Sigma_2 = \{b, c\}$. Let $\rho = abc$, $\rho_1 = ab$, $\rho_2 = bc$. Let $L = \{\rho, \rho_1, \rho_2\}$, $K = \{\rho_1, \rho_2\}$. We have $P_{\Sigma_1}(\rho) = P_{\Sigma_1}(\rho_1) = ab$, $P_{\Sigma_2}(\rho) = P_{\Sigma_2}(\rho_2) = bc$ and $P_{\Sigma_2}(\rho_1) = P_{\Sigma_1}(\rho_2) = b$. Then, $P_{\Sigma_1}(K) \cap P_{\Sigma_1}(L - K) = \{ab, b\} \cap \{ab\} = \{ab\}$ and $P_{\Sigma_2}(K) \cap P_{\Sigma_2}(L - K) = \{b, bc\} \cap \{bc\} = \{bc\}$. Thus, Condition (2) does not hold. However, Condition (1) holds. Indeed, observer 2 can distinguish between ρ and ρ_1, because $P_{\Sigma_2}(\rho) = bc \neq b = P_{\Sigma_2}(\rho_1)$. Also, observer 1 can distinguish between ρ and ρ_2, because $P_{\Sigma_1}(\rho) = ab \neq b = P_{\Sigma_1}(\rho_2)$. □

The following lemma states some closure properties of decentralized observability with respect to union, intersection and relative complement.

Lemma 3.3. *If K_1 and K_2 are observable w.r.t. L and $(\Sigma_1, \ldots, \Sigma_n)$ then $K_1 \cup K_2$, $L - K_1$ and $K_1 \cap K_2$ are also observable w.r.t. L and $(\Sigma_1, \ldots, \Sigma_n)$.*

Proof. Assume that $K_1 \cup K_2$ is not observable. Then, by Lemma 3.1, there exist $\rho \in K_1 \cup K_2$ and $\rho' \in L - (K_1 \cup K_2)$, such that $\forall i \in \{1, \ldots, n\}$, $P_{\Sigma_i}(\rho) = P_{\Sigma_i}(\rho')$. Observe that $\rho' \in (L - K_1) \cap (L - K_2)$. If $\rho \in K_1$ then ρ and ρ' contradict observability of K_1, whereas if $\rho \in K_2$ then ρ and ρ' contradict observability of K_2. Thus, $K_1 \cup K_2$ must be observable.

Assume that $L - K_1$ is not observable. Then, by Lemma 3.1, there exist $\rho \in L - K_1$ and $\rho' \in L - (L - K_1)$, such that $\forall i \in \{1, \ldots, n\}$, $P_{\Sigma_i}(\rho) = P_{\Sigma_i}(\rho')$. But $L - (L - K_1) = K_1$, therefore, ρ' and ρ contradict observability of K_1. Thus, $L - K_1$ must be observable. From the equality $K_1 \cap K_2 = L - ((L - K_1) \cup (L - K_2))$, we get that $K_1 \cap K_2$ is also observable. □

When $n = 1$, we are dealing with a centralized observation problem. In this case, checking observability is decidable, since Condition (1) is equivalent to the condition $P_{\Sigma_1}(K) \cap P_{\Sigma_1}(L - K) = \emptyset$. Since regular languages are closed under projection and set operations like complementation and intersection, the above condition can be easily checked. Moreover, if the condition holds (i.e., if K is observable w.r.t. L and Σ_1) then the decision function f can be represented as a deterministic finite-state automaton accepting $P_{\Sigma_1}(K)$. The worst-case complexity of checking the condition as well as the size of the automaton representing f are both exponential in the size of the regular expression or automaton representing K.

Unfortunately, these nice properties do not carry over to the decentralized case ($n \geqslant 2$). First, as it is shown in [14], the decision function f, when it exists, cannot always be represented by finite-state automata.[1] Second, as we show below, checking observability is no longer decidable.

Proposition 3.1. *Post's Correspondence Problem can be reduced to the decentralized observation problem for $n = 2$.*

Proof. First we recall Post's Correspondence Problem (PCP). We are given a finite alphabet Γ and two sets of words $A, B \subseteq \Gamma^*$, $A = \{w_1, w_2, \ldots, w_m\}$ and $B = \{u_1, u_2, \ldots, u_m\}$. We assume that $\forall i \in \{1, \ldots, m\}$, $w_i \neq u_i$, that is, the PCP has no trivial solution. We are asked: do there exist indices $i_1, \ldots, i_k \in \{1, \ldots, m\}$, $k \geqslant 1$, such that $w_{i_1} w_{i_2} \cdots w_{i_k} = u_{i_1} u_{i_2} \cdots u_{i_k}$.

We now reduce PCP to the decentralized observation problem for $n = 2$. We let $\Sigma_1 = \Gamma$ and set $\Sigma_2 = \{a_1, \ldots, a_m\}$, where $a_1, \ldots, a_m$ are new letters, so that $\Sigma_1 \cap \Sigma_2 = \emptyset$. We also set $\Sigma = \Sigma_1 \cup S_2$. We then define

$$K = (w_1 a_1 + \cdots + w_m a_m)^+$$

and

[1] For example, if $\Sigma = \{a, b\}$, $\Sigma_1 = \{a\}$, $\Sigma_2 = \{b\}$, $K = (ab)^*$ and $L = (ab)^* \cdot b^*$, then

$$f(a^k, b^l) = \begin{cases} 1, & \text{if } k = l, \\ 0, & \text{otherwise}, \end{cases}$$

solves the decentralized observation problem. However, since k and l are unbounded, f needs an unbounded "memory".

$$L = (w_1 a_1 + \cdots + w_m a_m)^+ \\ + (u_1 a_1 + \cdots + u_m a_m)^+.$$

We claim that K is observable with respect to L and Σ_1, Σ_2 iff PCP has no solution.

Assume first that PCP has a solution, $i_1, \ldots, i_k \in \{1, \ldots, m\}$, $k \geqslant 1$. Now, let

$$\rho = w_{i_1} a_{i_1} \cdots w_{i_m} a_{i_m}$$

and

$$\rho' = u_{i_1} a_{i_1} \cdots u_{i_m} a_{i_m}.$$

By definition, $\rho \in K$ and $\rho' \in L$. We claim that $\rho' \notin K$: otherwise, $u_{i_1} = w_{i_1}$ which contradicts the assumption on PCP. Now, observe that

$$P_{\Sigma_1}(\rho) = w_{i_1} \cdots w_{i_m} = u_{i_1} \cdots u_{i_m} = P_{\Sigma_1}(\rho')$$

and

$$P_{\Sigma_2}(\rho) = a_{i_1} \cdots a_{i_m} = P_{\Sigma_2}(\rho').$$

Thus, ρ and ρ' contradict Condition (1) and, by Lemma 3.1, K is not observable w.r.t. L and Σ_1, Σ_2.

Conversely, assume that K is not observable w.r.t. L and Σ_1, Σ_2. By Lemma 3.1, there exist $\rho \in K$ and $p' \in L - K$ such that $P_{\Sigma_i}(\rho) = P_{\Sigma_i}(\rho')$, for $i \in \{1, 2\}$. By definition of K, ρ must be of the form $w_{i_1} a_{i_1} \cdots w_{i_k} a_{i_k}$, with $k \geqslant 1$. By definition of L, ρ' must be of the form $u_{j_1} a_{j_1} \cdots u_{j_l} a_{j_l}$, with $l \geqslant 1$. The equality $P_{\Sigma_2}(\rho) = P_{\Sigma_2}(\rho')$ is equivalent to $a_{i_1} \cdots a_{i_k} = a_{j_1} \cdots a_{j_l}$, which implies $k = l$ and $i_q = j_q$, for $q \in \{1, \ldots, k\}$. Then, the equality $P_{\Sigma_1}(\rho) = P_{\Sigma_1}(\rho')$ is equivalent to $w_{i_1} \cdots w_{i_k} = u_{i_1} \cdots u_{i_k}$, which means that PCP has a solution. $\square$

PCP is known to be undecidable (e.g., see [5]), therefore, checking observability for $n = 2$ is also undecidable. Since the decentralized observation problem for a given n can be easily reduced to a decentralized observation problem for $n' > n$ (simply set $\Sigma_{n+1} = \cdots = \Sigma_{n'} = \emptyset$), we have the following result.

Theorem 3.1. *The decentralized observation problem is undecidable for $n \geqslant 2$.*

In [18] it is shown that the decentralized observation problem is undecidable for $n \geqslant 3$, even in the case where K and L are prefix closed. The question remains open for prefix-closed languages and $n = 2$.

4. A decentralized control problem

The plant is modeled as a regular language $G \subseteq \Sigma^*$ over a finite alphabet Σ.

To model the observation and control of the plant by a set of n supervisors, we introduce subalphabets $\Sigma_i, \Gamma_i \subseteq \Sigma$, for $i \in \{1, \ldots, n\}$. Σ_i models the set of events that supervisor i observes (*observable* events) and Γ_i the set of events that supervisor i controls (*controllable* events). Supervisor i is modeled as a total function $C_i : \Sigma_i^* \to 2^{\Gamma_i}$. Given $\pi \in \Sigma_i^*$, $C_i(\pi)$ represents the set of events that supervisor i "enables" once it has observed π.

The *controlled plant*, denoted $L(G/C_1, \ldots, C_n)$, is defined to be the language $G \cap G_C \subseteq \Sigma^*$, where G_C is a prefix-closed language defined recursively as follows:

- $\epsilon \in G_C$,
- for all $\rho \in \Sigma^*$ and $a \in \Sigma$, $\rho a \in G_C$ iff

$$\begin{aligned} & \rho \in G_C \\ \wedge \quad & \rho a \in \mathrm{Pref}(G) \\ \wedge \quad & \bigwedge_{i \in \{1,\ldots,n\}} a \in \Gamma_i \Rightarrow a \in C_i\big(P_{\Sigma_i}(\rho)\big). \end{aligned}$$

That is, the controlled plant can perform a after ρ iff the following two conditions hold:

(a) the plant can perform a after ρ;
(b) for each i, if a is controllable by supervisor i, then the latter enables a once it observes $P_{\Sigma_i}(\rho)$.

If $G_C \subseteq \mathrm{Pref}(L(G/C_1, \ldots, C_n))$ then the supervisors are called *non-blocking*. Notice that the inverse inclusion always holds. Indeed, $\mathrm{Pref}(L(G/C_1, \ldots, C_n)) = \mathrm{Pref}(G \cap G_C) \subseteq \mathrm{Pref}(G) \cap \mathrm{Pref}(G_C) = \mathrm{Pref}(G) \cap G_C \subseteq G_C$. Thus, the non-blocking property implies $G_C = \mathrm{Pref}(L(G/C_1, \ldots, C_n))$.

Definition 4.1 (*Decentralized control problem*). Given finite alphabet Σ, subalphabets $\Sigma_i, \Gamma_i \subseteq \Sigma$, for $i = 1, \ldots, n$, and regular languages $G, E \subseteq \Sigma^*$, do there exist non-blocking supervisors $C_i : \Sigma_i^* \to 2^{\Gamma_i}$, for $i = 1, \ldots, n$, such that $L(G/C_1, \ldots, C_n) \subseteq E$.

If supervisors as required above exist, we say that E is controllable with respect to G and $(\Sigma_1, \Gamma_1, \ldots, \Sigma_n, \Gamma_n)$.

Non-blockingness is required in order for trivial solutions to be eliminated. A trivial way of finding supervisors such that $L(G/C_1, \ldots, C_n) \subseteq E$ is to try the supervisors that disable all controllable events at all times (i.e., $\forall i \in \{1, \ldots, n\}$, $\forall \pi_i \in \Sigma_i^*$, $C_i(\pi_i) = \emptyset$). If for these supervisors $L(G/C_1, \ldots, C_n) \subseteq E$, then the problem is solved. Otherwise, no solution exists, since the above supervisors are *maximally restrictive*.

An alternative way of circumventing trivial solutions as the above is to require *minimally restrictive* supervisors, that is, such that the language $L(G/C_1, \ldots, C_n)$ is as large as possible (in the set-theoretic sense). However, as it is shown in [3], in the partial observability case (i.e., $\Sigma_i \neq \Sigma$), such unique solutions do not always exist, even for $n = 1$.

We now proceed to prove the main result of this section.

Proposition 4.1. *The decentralized observation problem can be reduced to the decentralized control problem.*

Proof. Given finite alphabet Λ, subalphabets $\Lambda_i \subseteq \Lambda$, $i = 1, \ldots, n$, and regular languages $K \subseteq L \subseteq \Lambda^*$, we will define a new alphabet Σ, subalphabets $\Sigma_i, \Gamma_i \subseteq \Sigma$, $i = 1, \ldots, n$, and regular languages $G, E \subseteq \Sigma^*$, such that K is observable w.r.t. L and $\{\Lambda_i\}_{i=1,\ldots,n}$ iff E is controllable w.r.t. G and $(\Sigma_1, \Gamma_1, \ldots, \Sigma_n, \Gamma_n)$.

The intuition is as follows. The plant first generates a behavior in L, which the supervisors simply observe. Then, supervisors 2 to n are allowed to "transmit" their observations to supervisor 1. Finally, supervisor 1 must "decide" whether the original behavior was in K or in $L - K$.

For each $i \in \{2, \ldots, n\}$, let $l_i = |\Lambda_i|$ and $\Lambda_i = \{a_i^1, \ldots, a_i^{l_i}\}$. Then, define T_i to be an alphabet of l_i new letters, $T_i = \{t_i^1, \ldots, t_i^{l_i}\}$. That is, the sets T_i are pairwise disjoint and disjoint from Λ. Let $T = \bigcup_{i=2,\ldots,n} T_i$. Also, define the following alphabets of new letters: $A = \{\alpha, \alpha'\}$, $B = \{\beta, \beta'\}$ and $\Delta = \{\delta_1, \ldots, \delta_n\}$.

The new alphabet Σ and the subalphabets of observable and controllable events for each supervisor are defined below.

$$\Sigma = \Lambda \cup T \cup A \cup B \cup \Delta,$$

$$\Sigma_1 = \Lambda_1 \cup T \cup \Delta,$$

$$\Sigma_i = \Lambda_i \cup T_i \cup \{\delta_{i-1}\}, \quad \text{for } i = 2, \ldots, n,$$

$$\Gamma_1 = \{\alpha', \beta'\},$$

$$\Gamma_i = T_i \cup \{\delta_i\}, \quad \text{for } i = 2, \ldots, n.$$

G is defined as follows:

$$G = \big(K\alpha + (L-K)\beta\big)\delta_1 T_2^* \delta_2 \cdots \delta_{n-1} T_n^* \delta_n (\alpha' + \beta').$$

That is, G starts by performing either a word in K followed by α or a word in $L - K$ followed by β. Then it outputs δ_1 signaling to supervisor 2 to start "transmitting" its observation. This is a word in T_2^*, followed by δ_2, signaling the end of transmission for supervisor 2 and the beginning of transmission for supervisor 3. And so on, until δ_n, signaling the end of transmission for supervisor n. Then, supervisor 1 must enable either α' or β'.

E is defined as follows:

$$E = \big((\Sigma - A)^* \alpha (\Sigma - A)^* \alpha'\big) + \big((\Sigma - B)^* \beta (\Sigma - B)^* \beta'\big).$$

That is, α must be followed by α' and β must be followed by β'. In other words, if the initial behavior of G was in K then supervisor 1 must enable α', otherwise it must enable β'.

We claim that K is observable w.r.t. L and $\{\Lambda_i\}_{i=1,\ldots,n}$ iff E is controllable w.r.t. G and $(\Sigma_1, \Gamma_1, \ldots, \Sigma_n, \Gamma_n)$.

($\Rightarrow$) Assume that K is not observable w.r.t. L and $\{\Lambda_i\}_{i=1,\ldots,n}$ and suppose supervisors $\{C_i\}_{i=1,\ldots,n}$ exist. By Lemma 3.1, there exist $\rho \in K$ and $\rho' \in L - K$, such that $\forall i \in \{1, \ldots, n\}$, $P_{\Lambda_i}(\rho) = P_{\Lambda_i}(\rho')$. By definition of G and since $\rho \in K$ and all events in $\Lambda \cup \{\alpha\}$ are uncontrollable, $\rho\alpha \in G_C$. Since the supervisors are non-blocking, $\rho\alpha \in \mathrm{Pref}(L(G/C_1, \ldots, C_n))$, thus, there exists $\tau \in \Sigma^*$ such that $\rho\alpha\tau \in L(G/C_1, \ldots, C_n)$. This implies $\rho\alpha\tau \in E$. By definition of E, $\tau = \sigma\alpha'$. We claim that $\rho'\beta\sigma\alpha' \in L(G/C_1, \ldots, C_n)$, which contradicts the correctness of the supervisors, since $\rho'\beta\sigma\alpha' \notin E$. The proof of the claim is done by induction on the length of σ and using the fact that for all $i \in \{1, \ldots, n\}$, $P_{\Sigma_i}(\rho\alpha) = P_{\Lambda_i}(\rho) = P_{\Lambda_i}(\rho') = P_{\Sigma_i}(\rho'\beta)$, thus, $C_i(\rho\alpha) = C_i(\rho'\beta)$.

($\Leftarrow$) Assume that K is observable w.r.t. L and $\{\Lambda_i\}_{i=1,\ldots,n}$. That is, there exists a total function $f : \Lambda_1^* \times \cdots \times \Lambda_n^* \to \{0, 1\}$, such that $\forall \rho \in L$,

$(\rho \in K \Leftrightarrow f(P_{\Lambda_1}(\rho), \ldots, P_{\Lambda_n}(\rho)) = 1)$. We will define non-blocking supervisors $\{C_i\}_{i=1,\ldots,n}$, such that $L(G/C_1, \ldots, C_n) \subseteq E$. Given a word $\tau \in T_i^*$, for some $i = 2, \ldots, n$, let $\hat{\tau} \in \Lambda_i^*$ be the word obtained by replacing each letter of τ by its "corresponding" letter in Λ_i. For example, if $\tau = t_i^{j_1} t_i^{j_2} t_i^{j_3}$ then $\hat{\tau} = a_i^{j_1} a_i^{j_2} a_i^{j_3}$. Then, supervisor 1 is defined as follows:

$$C_1(\pi) = \begin{cases} \{\alpha'\}, & \text{if } \pi = \pi_1 \delta_1 \tau_2 \delta_2 \cdots \delta_{n-1} \tau_n \delta_n \\ & \text{and } f(\pi_1, \hat{\tau}_2, \ldots, \hat{\tau}_n) = 1, \\ \{\beta'\}, & \text{if } \pi = \pi_1 \delta_1 \tau_2 \delta_2 \cdots \delta_{n-1} \tau_n \delta_n \\ & \text{and } f(\pi_1, \hat{\tau}_2, \ldots, \hat{\tau}_n) = 0, \\ \emptyset, & \text{otherwise.} \end{cases}$$

Intuitively, supervisor 1 disables both α' and β' until it receives all "messages" transmitted by the other supervisors (this is signaled by the last message terminator event, δ_n). Each such message, τ_i, contains the observation of supervisor i, for $i = 2, \ldots, n$. Then, supervisor 1 applies the observation function f. If f yields 1, the original behavior of the plant was in K, thus, α must have followed it, so supervisor 1 must enable α' in order to meet the specification E. Otherwise, β' must be enabled.

We only sketch the behavior of supervisors 2 to n. These supervisors only serve as to "transmit" their observations to supervisor 1. Supervisor i starts sending its observation immediately after it observes the event δ_{i-1}. Once it has finished transmitting, supervisor i sends the "message terminator" event δ_i. Sending a message amounts to enabling only one event in T_i at a time. For example, consider supervisor 2 and suppose it has observed the word $a_2^1 a_2^2 a_2^3 \in \Lambda_2^*$. Then, its behavior will be defined as follows:

$$C_2(a_2^1 a_2^2 a_2^3 \delta_1) = \{t_2^1\},$$
$$C_2(a_2^1 a_2^2 a_2^3 \delta_1 t_2^1) = \{t_2^2\},$$
$$C_2(a_2^1 a_2^2 a_2^3 \delta_1 t_2^1 t_2^2) = \{t_2^3\},$$
$$C_2(a_2^1 a_2^2 a_2^3 \delta_1 t_2^1 t_2^2 t_2^3) = \{\delta_2\}.$$

To see that the supervisors defined above solve the control problem, observe first that $(K\alpha + (L - K)\beta)\delta_1 \subseteq G_C$. This is by definition of G and because all events in $\Lambda \cup \{\alpha, \beta, \delta_1\}$ are uncontrollable. Now, by the informal definitions of supervisors 2 to n, we can convince ourselves that $\delta_2, \delta_3, \ldots, \delta_n$ will eventually occur, thus, by definition of supervisor 1, α' or β' will eventually occur. This means the supervisors are non-blocking. To see that $L(G/C_1, \ldots, C_n) \subseteq E$, consider a behavior $\rho\alpha \in G_C$ (the case $\rho'\beta$ is similar). By definition of G, $\rho \in K$. Then, with a similar reasoning as the one above, we can convince ourselves that the behavior observed by supervisor 1 is $\pi_1 \delta_1 \tau_2 \delta_2 \cdots \delta_{n-1} \tau_n \delta_n$, where $\pi = P_{\Lambda_1}(\rho)$ and $\hat{\tau}_i = P_{\Lambda_i}(\rho)$, for $i = 2, \ldots, n$. Since K is observable, $f(P_{\Lambda_1}(\rho), \ldots, P_{\Lambda_n}(\rho)) = 1$, which means that supervisor 1 enables α'. Thus, the entire behavior of the controlled plant is $\rho\alpha\delta_1 \tau_2 \delta_2 \cdots \delta_{n-1} \tau_n \delta_n \alpha' \in E$. □

Proposition 4.1 and Theorem 3.1 imply the following.

Theorem 4.1. *The decentralized control problem is undecidable for $n \geqslant 2$.*

5. Conclusion and future work

We have presented a simple decentralized observation and control setting and showed that fundamental questions such as existence of observers or controllers are undecidable in this setting.

Our motivation for the decentralized control problem comes from its relation to *protocol synthesis*. Indeed, ability to solve the control problem would imply ability to automatically synthesize communication protocols like the alternating bit protocol, which is modeled as a decentralized control problem in [14].

We believe that a way to obtain decidable decentralized observation and control problems is to endow the observers/controllers with a communication architecture, permitting them to exchange observations *online*. Work in this direction is presented in [19], where it is shown that the communication must be of bounded delay, otherwise undecidability persists.

Another work direction is to study versions of the above problems where observers and controllers are required to be finite-state. Indeed, as shown in [14], there are cases where infinite-state observers (or controllers) exist, but no finite-state ones exist.

References

[1] J. Berstel, Transductions and Context-Free Languages, Teubner, Wiesbaden, 1979.

[2] V.D. Blondel, J.N. Tsitsiklis, A survey of computational complexity results in systems and control, Automatica 36 (9) (2000) 1249–1274.

[3] R. Cieslak, C. Desclaux, A. Fawaz, P. Varaiya, Supervisory control of discrete-event processes with partial observations, IEEE Trans. Automat. Control 33 (1988) 249–260.

[4] R. Debouk, S. Lafortune, D. Teneketzis, Coordinated decentralized protocols for failure diagnosis of discrete event systems, in: IEEE Conference on Decision and Control, December, 1998.

[5] M. Garey, D. Johnson, Computers and Intractability: A Guide to the Theory of NP-Completeness, Freeman, San Francisco, CA, 1979.

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

[7] O. Kupferman, M. Vardi, Synthesizing distributed systems, in: 16th Annual Symp. on Logic in Computer Science, 2001.

[8] H. Lamouchi, J. Thistle, Effective control synthesis for DES under partial observations, in: IEEE Conference on Decision and Control, 2000.

[9] F. Lin, H. Mortazavian, A normality theorem for decentralized control of discrete event systems, IEEE Trans. Automat. Control 39 (5) (1994).

[10] F. Lin, W. Wonham, Decentralized supervisory control of discrete-event systems, Inform. Sci. 44 (1988) 199–224.

[11] P. Madhusudan, P.S. Thiagarajan, Distributed controller synthesis for local specifications, in: 28th ICALP, Crete, Greece, in: Lecture Notes in Computer Science, vol. 2076, Springer, Berlin, 2001, pp. 396–407.

[12] P. Madhusudan, P.S. Thiagarajan, A decidable class of asynchronous distributed controllers, in: CONCUR ’02, in: Lecture Notes in Computer Science, vol. 2421, Springer, Berlin, 2002, pp. 145–160.

[13] A. Pnueli, R. Rosner, Distributed reactive systems are hard to synthesize, in: Proceedings of the 31th IEEE Symposium Foundations of Computer Science, 1990, pp. 746–757.

[14] A. Puri, S. Tripakis, P. Varaiya, Problems and examples of decentralized observation and control for discrete event systems, in: CAV ’01 Symposium on Supervisory Control for Discrete Event Systems (SCODES), 2001;
Also in: B. Caillaud, P. Darondeau, L. Lavagno, X. Xie (Eds.), Synthesis and Control of Discrete Event Systems, Kluwer Academic, Dordrecht, 2002.

[15] P. Ramadge, W. Wonham, Supervisory control of a class of discrete event processes, SIAM J. Control Optim. 25 (1) (1987).

[16] K. Rudie, J.C. Willems, The computational complexity of decentralized discrete-event control problems, IEEE Trans. Automat. Control 40 (7) (1995).

[17] K. Rudie, W. Wonham, Think globally, act locally: Decentralized supervisory control, IEEE Trans. Automat. Control 37 (1992).

[18] S. Tripakis, Undecidable problems of decentralized observation and control, in: IEEE Conference on Decision and Control (CDC ’01), 2001.

[19] S. Tripakis, Decentralized control of discrete event systems with bounded or unbounded delay communication, in: 6th International Workshop on Discrete Event Systems (WODES ’02), IEEE CS Press, 2002.

[20] J.N. Tsitsiklis, On the control of discrete event dynamical systems, Math. Control Signals Systems 2 (2) (1989).

[21] J.N. Tsitsiklis, M. Athans, On the complexity of decentralized decision making and detection problems, IEEE Trans. Automat. Control 30 (5) (1985).

[22] Y. Willner, M. Heymann, On supervisory control of concurrent discrete-event systems, Internat. J. Control 54 (5) (1991).