Undecidable Problems of Decentralized Observation and Control

Stavros Tripakis [1]
VERIMAG, France
tripakis@imag.fr, www-verimag.imag.fr

Abstract

We introduce a new notion of decentralized observability for discrete-event systems, which we call *joint observability*. We prove that checking joint observability of a regular language w.r.t. one observer is decidable, whereas for two (or more) observers the problem becomes undecidable. Based on this result, we show that a related decentralized control problem is also undecidable. We finally provide an extensive study relating our work to existing work in the literature.

1 Introduction

Consider the following observation problem. We would like to check whether a given system behaves correctly or not, based solely on its observed behavior, which may be partial and decentralized. The system is formalized as a regular language L over an alphabet Σ. The correct behaviors are formalized as a regular language $K \subseteq L$. Partial and decentralized observation is formalized by considering subsets $\Sigma_i \subseteq \Sigma, i = 1, ..., k$, with the meaning that observer i can only observe projections of strings on Σ_i. For example, if $abcab$ is a behavior of the system, and $\Sigma = \{a, b, c\}, \Sigma_1 = \{a, b\}, \Sigma_2 = \{b, c\}$, then observer 1 observes $abab$ and observer 2 observes bcb.

Detecting whether the system behaves correctly or not is possible only if there do not exist two behaviors of the system, one correct and the other erroneous, yet both yielding the same observation to each of the observers. If this is not the case, we say that K is *jointly observable* with respect to L and the observers, and we can hope to synthesize such observers.

Joint observability is interesting in many contexts. For example, we could imagine an *off-line diagnosis* process, where data-logs are gathered at a distributed set of locations in a plant, and then are jointly examined to check for absence of faults or any other desired behavior.

Unfortunately, checking joint observability turns out to be undecidable in the case of two or more observers (though it is decidable for one). This may seem surprising, since we only consider regular languages, which means both the system and the specification can be modeled by finite-state automata. The proof is by reduction of Post's Correspondence Problem (PCP), which is known to be undecidable. The proof provides some intuition to why the problem is undecidable: we do not know how "long" pairs of behaviors we should look at, in order to check whether they both give the same projections, yet one behavior is correct and the other not.

Observability is very much related to control: controllers take actions based on the sequence of events they observe. Indeed, by essentially reducing a decentralized control problem to a problem of checking joint observability, we show that the decentralized control problem is also undecidable for two controllers.

Our work follows the framework of supervisory controller synthesis for discrete-event systems [16]. There has been a lot of work in this area. We devote section 6 to relating our work to previous work.

2 The Decentralized Observation Problem

2.1 Preliminaries

Let Σ be a finite alphabet. Σ^* denotes the set of all finite strings over Σ, ϵ denotes the empty string, and $\Sigma^+ = \Sigma^* \setminus \{\epsilon\}$. Given two strings ρ and ρ', $\rho\rho'$ or $\rho \cdot \rho'$ is the concatenation of ρ and ρ'. Given sets of strings A and B, AB or $A \cdot B$ is the set $\{\rho\rho' \mid \rho \in A, \rho' \in B\}$. Given a string $\rho = a_1 a_2 \cdots a_n$, the set of prefixes of ρ is $\{\epsilon, a_1, a_1a_2, ..., a_1a_2 \cdots a_n\}$. We say that ρ' is a strict prefix of ρ if ρ' is a prefix of ρ and $\rho' \neq \rho$. Given a set of strings L, the prefix closure of L, denoted $\overline{L}$, is the set of strings obtained by adding to L the prefixes of all strings in L.

Given $\Sigma_i \subseteq \Sigma$, we define the *projection* of a string $\rho \in \Sigma^*$ to Σ_i, denoted $\rho/_{\Sigma_i}$, as the string $\rho_i \in \Sigma_i^*$ obtained from ρ by erasing all letters not in Σ_i. For example, if $\Sigma = \{a, b, c\}$ and $\Sigma_i = \{a, c\}$, then $abbcbacb/_{\Sigma_i} = acac$.

[1]This is an extended version of a paper published in CDC'01.

$P_i : \Sigma^* \rightarrow \Sigma_i^*$ is the projection function w.r.t. Σ_i, i.e., $P_i(\rho) = \rho/_{\Sigma_i}$. We will also use P_i^{-1} to denote the inverse of P_i, that is, for $\sigma \in \Sigma_i^*$, $P_i^{-1}(\sigma) = \{\rho \in \Sigma^* \mid P_i(\rho) = \sigma\}$. The definitions of P_i and $P_i{-}1$ naturally extend from single strings to sets of strings.

2.2 Jointly Observable Languages

Definition 2.1 (Joint observability) *Let $K, L \subseteq \Sigma^*$ be two regular languages over Σ, such that $K \subseteq L$. Given $\Sigma_i \subseteq \Sigma$, $i = 1, ..., k$, we say that K is* jointly observable with respect to L and $\Sigma_1, ..., \Sigma_k$, *if*

$$\forall \rho \in K, \rho' \in L - K, \exists i = 1, ..., k, \rho/_{\Sigma_i} \neq \rho'/_{\Sigma_i}.$$

That is, K is jointly observable[1] iff there are no two behaviors ρ, ρ' in L, such that $\rho \in K$, $\rho' \notin K$, yet ρ and ρ' yield the same observation to each observer i.

A physical interpretation of the above definition is as follows. A system generates finite behaviors in L. K includes the correct behaviors and $L - K$ the erroneous behaviors. Each behavior is monitored by each of the k observers. At the end of the execution of the system, the observers get together and decide whether the behavior was correct or not. The following lemma makes this precise.

Lemma 2.1 *K is jointly observable w.r.t. L, Σ_i, $i = 1, ..., k$, iff there exists a function $f : \Sigma_1^* \times \cdots \times \Sigma_k^* \rightarrow \{0, 1\}$ such that $\forall \rho \in L$, if $f(P_1(\rho), \cdots, P_k(\rho)) = 1$ then $\rho \in K$ and if $f(P_1(\rho), \cdots, P_k(\rho)) = 0$ then $\rho \notin K$.*

Proof: If K is not jointly observable, then there exist $\rho \in K$, $\rho' \in L - K$, such that $P_i(\rho) = P_i(\rho')$, for all $i = 1, ..., k$. But then, $f(P_1(\rho), \cdots, P_k(\rho)) = f(P_1(\rho'), \cdots, P_k(\rho'))$ can be neither 1 (because $\rho' \notin K$), nor can it be 0 (because $\rho \in K$).

For the converse, assume that K is jointly observable, consider some $\rho \in L$ and let $\sigma_i = P_i(\rho)$, for $i = 1, ..., k$. Consider the set $O_\rho = P_1^{-1}(\sigma_1) \cap \cdots \cap P_k^{-1}(\sigma_k) \cap L$. If $O_\rho \subseteq K$, we let $f(\sigma_1, ..., \sigma_k) = 1$. If $O_\rho \cap K = \emptyset$, we let $f(\sigma_1, ..., \sigma_k) = 0$. The case $O_\rho \cap K \neq \emptyset$ and $O_\rho \cap (L - K) \neq \emptyset$ is not possible. Indeed, assume it true, and suppose $\rho \in K$. Then, let $\rho' \in O_\rho \cap (L - K)$. Since $\rho' \in O_\rho$, $P_i(\rho') = P_i(\rho)$, for $i = 1, ..., k$, which contradicts the assumption that K is jointly observable. A similar contradiction arises if we assume $\rho \in L - K$ and take $\rho' \in O_\rho \cap K$. ∎

[1] We would like to use the term observable instead of jointly observable, but we prefer not to confuse our definition with those of (centralized) *observability* [10] or (decentralized) *co-observability* [21]. Indeed, as we show in section 6, joint observability is stronger than observability in the centralized case, and incomparable to co-observability in the decentralized case.

An important property of joint observability is that it is closed under union, intersection and complement with respect to L.

Theorem 2.1 *If K_1 and K_2 are jointly observable, then $K_1 \cup K_2$, $L - K_1$ and $K_1 \cap K_2$ are also jointly observable.*

Proof: Assume that $K_1 \cup K_2$ is not jointly observable. Then, there must exist $\rho, \rho' \in L$, such that $\rho \in K_1 \cup K_2$, $\rho' \notin K_1 \cup K_2$ (i.e., $\rho' \notin K_1$ and $\rho' \notin K_2$), yet $P_i(\rho) = P_i(\rho')$, for $i = 1, ..., k$. Now, if $\rho \in K_1$, then ρ, ρ' provide a counter-example to K_1 being jointly observable, whereas if $\rho \in K_2$, then ρ, ρ' provide a counter-example to K_2 being jointly observable. In both cases we get a contradiction, therefore $K_1 \cup K_2$ must be jointly observable.

It is also easy to show that if $L - K_1$ was not jointly observable, then K_1 would not be jointly observable either. 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 jointly observable. ∎

2.3 The Decentralized Observation Problem

Given L, K and Σ_i, $i = 1, ..., k$, we want to check whether K is jointly observable w.r.t. L and Σ_i, $i = 1, ..., k$.

3 Decidability of Centralized Observation

Theorem 3.1 *Given regular languages $K \subseteq L \subseteq \Sigma^*$, and $\Sigma_1 \subseteq \Sigma$, there is an algorithm to decide whether or not K is jointly observable with respect to L and Σ_1.*

Proof sketch: Let A_K and A_{L-K} be automata that recognize K and $L - K$ respectively. Let $A = A_K \times_{\Sigma_1} A_{L-K}$ be the product of A_K and A_{L-K}, obtained by synchronizing all transitions of A_K and A_{L-K} labeled with the same letter in Σ_1, and interleaving asynchronously all transitions labeled otherwise. A state (s, s') of A is defined to be accepting iff s is an accepting state of A_K and s' is an accepting state of A_{L-K}. Then, we can see that K is jointly observable iff A has an accepting behavior. Indeed, A has an accepting behavior iff there exist runs ρ, ρ' accepted by A_K and A_{L-K} respectively, and giving the same projections to Σ_1 (in order for the corresponding transitions to synchronize). ∎

The centralized observation problem was first solved in [26].

4 Undecidability of Decentralized Observation

Theorem 4.1 *Suppose $K \subseteq L$ are regular languages over Σ. The problem of checking whether K is jointly observable with respect to L and $\Sigma_1, \Sigma_2 \subseteq \Sigma$, is undecidable.*

Proof: We reduce Post's Correspondence Problem (PCP) to an observation problem with two observers. PCP is known to be undecidable [6].

First we recall PCP. We are given a finite alphabet Σ_1 and two sets of strings $A, B \subseteq \Sigma_1^*$, $A = \{w_1, w_2, ..., w_n\}$ and $B = \{u_1, u_2, ..., u_n\}$. We assume that for all $i = 1, ..., n$, $w_i \neq u_i$, that is, the PCP has no trivial solution. We are asked: do there exist indices $i_1, ..., i_k \in \{1, ..., n\}$, $k \geq 1$, such that $w_{i_1} w_{i_2} \cdots w_{i_k} = u_{i_1} u_{i_2} \cdots u_{i_k}$.

We now translate the above instance of PCP to an observation problem. Let $\Sigma_2 = \{a_1, a_2, ..., a_n\}$ be a set of new letters, not in Σ_1. We will construct languages K and L over $\Sigma_1 \cup \Sigma_2$, such that K is jointly observable with respect to L and Σ_1, Σ_2 iff the answer to the above PCP instance is "no" (i.e., the PCP has no solution).

L is defined by the following regular expression:

$$\textit{good}\ (w_1 a_1 + \cdots + w_n a_n)^+ + \textit{bad}\ (u_1 a_1 + \cdots + u_n a_n)^+$$

The specification K is defined to be all strings in L that start with *good*.

Assume first that the answer to the above PCP is "yes", that is, there exist indices $i_1, ..., i_k \in [1..n]$, $k \geq 1$, such that $w_{i_1} w_{i_2} \cdots w_{i_k} = u_{i_1} u_{i_2} \cdots u_{i_k}$. Then, let $\rho = \textit{good}\, w_{i_1} a_{i_1} w_{i_2} a_{i_2} \cdots w_{i_k} a_{i_k}$ and $\rho' = \textit{bad}\, u_{i_1} a_{i_1} u_{i_2} a_{i_2} \cdots u_{i_k} a_{i_k}$. Both $\rho, \rho' \in L$, but only $\rho \in K$. However, $\rho/_{\Sigma_1} = w_{i_1} w_{i_2} \cdots w_{i_k} = u_{i_1} u_{i_2} \cdots u_{i_k} = \rho'/_{\Sigma_1}$. And $\rho/_{\Sigma_2} = a_{i_1} a_{i_2} \cdots a_{i_k} = \rho'/_{\Sigma_2}$. Therefore, K is not jointly observable.

In the other direction, assume that K is not jointly observable. This means there exist $\rho, \rho' \in L$, such that $\rho/_{\Sigma_i} = \rho'/_{\Sigma_i}$, for $i = 1, 2$, yet only $\rho \in K$. By definition of L and K, ρ must be of the form $\textit{good}\, w_{i_1} a_{i_1} w_{i_2} a_{i_2} \cdots w_{i_k} a_{i_k}$ and ρ' must be of the form $\textit{bad}\, u_{j_1} a_{j_1} u_{j_2} a_{j_2} \cdots u_{j_l} a_{j_l}$. Since $\rho/_{\Sigma_2} = \rho'/_{\Sigma_2}$, it must be that $k = l$ and $i_1 = j_1, i_2 = j_2, ..., i_k = j_k$. Moreover, since $\rho/_{\Sigma_1} = \rho'/_{\Sigma_1}$, it must be that $w_{i_1} w_{i_2} \cdots w_{i_k} = u_{i_1} u_{i_2} \cdots u_{i_k}$, which means that the answer to the above PCP is "yes". ∎

The language L used in the proof of theorem 4.1 is not prefix-closed. The question remains open whether checking joint observability of a language K with respect to a prefix-closed language L and two observers is decidable. However, we can still prove that the problem is undecidable for three observers, even in the case where L and K are prefix closed.

Theorem 4.2 *Suppose $K \subseteq L$ are prefix-closed regular languages over Σ. The problem of checking whether K is jointly observable with respect to L and $\Sigma_1, \Sigma_2, \Sigma_3 \subseteq \Sigma$, is undecidable.*

Proof: The proof is much like the one of theorem 4.1. Let Σ_1, Σ_2 be as in that proof, and let Σ_3 be $\{b_1, ..., b_n\}$, where b_i's are new letters. L is defined to be the prefix closure of the language defined by the following regular expression:

$$(a_1 w_1 b_1 + \cdots + a_n w_n b_n)^* + (b_1 u_1 a_1 + \cdots + b_n u_n a_n)^*$$

K is defined to be the prefix closure of the language defined by the following regular expression:

$$(a_1 w_1 b_1 + \cdots + a_n w_n b_n)^*$$

We show that PCP has a solution iff K is not jointly observable w.r.t. L, Σ_1, Σ_2, Σ_3.

Assume PCP has a solution $w_{i_1} \cdots w_{i_k} = u_{i_1} \cdots u_{i_k}$. Then, let $\rho = a_{i_1} w_{i_1} b_{i_1} \cdots a_{i_k} w_{i_k} b_{i_k}$ and $\rho' = b_{i_1} u_{i_1} a_{i_1} \cdots b_{i_k} u_{i_k} a_{i_k}$. Both $\rho, \rho' \in L$, but only $\rho \in K$. However, $\rho/_{\Sigma_1} = w_{i_1} \cdots w_{i_k} = u_{i_1} \cdots u_{i_k} = \rho'/_{\Sigma_1}$, $\rho/_{\Sigma_2} = a_{i_1} \cdots a_{i_k} = \rho'/_{\Sigma_2}$, and $\rho/_{\Sigma_3} = b_{i_1} \cdots b_{i_k} = \rho'/_{\Sigma_3}$. Therefore, K is not jointly observable.

Now assume K is not jointly observable. This means there exist $\rho, \rho' \in L$, such that $\rho/_{\Sigma_i} = \rho'/_{\Sigma_i}$, for $i = 1, 2, 3$, yet only $\rho \in K$. By definition of L and K, ρ must be of the form $a_{i_1} w_{i_1} b_{i_1} \cdots a_{i_k} w_{i_k} b_{i_k} \gamma_1$, where γ_1 is a strict prefix of $a_i w_i b_i$ for some i, and ρ' must be of the form $b_{j_1} u_{j_1} a_{j_1} \cdots b_{j_l} u_{j_l} a_{j_l} \gamma_2$, where γ_2 is a strict prefix of $b_j u_j a_j$ for some j. By the fact that $\rho/_{\Sigma_2} = \rho'/_{\Sigma_2}$ and $\rho/_{\Sigma_3} = \rho'/_{\Sigma_3}$, we conclude that $\gamma_1 = \gamma_2 = \epsilon$, $k = l$ and $i_1 = j_1, i_2 = j_2, ..., i_k = j_k$. Moreover, since $\rho/_{\Sigma_1} = \rho'/_{\Sigma_1}$, it must be that $w_{i_1} w_{i_2} \cdots w_{i_k} = u_{i_1} u_{i_2} \cdots u_{i_k}$, which means that the PCP has a solution. ∎

5 Undecidability of Decentralized Control

We now extend the undecidability result to decentralized control. First, we define the decentralized control problem (for simplicity, we consider the case of two controllers). We follow the setting of [14].

Consider an *automaton* $G = (S, q_0, \delta, \Sigma, F)$ where S is a set of states, $q_0 \in S$ is the initial state, Σ is the event set, $\delta : S \times \Sigma \rightarrow S$ is the transition function (in general, δ is a partial function) and $F \subseteq S$ is the set of *accepting* states. Given $\rho = \gamma_1 \cdots \gamma_k \in \Sigma^*$, we define $\delta(\rho) = s_k$, if there exists a sequence of states $s_0 s_1 \ldots s_k$, such that $s_0 = q_0$ and $s_{i+1} = \delta(s_i, \gamma_{i+1})$; otherwise, $\delta(\rho)$ is undefined. The *marked language* of G, denoted $L_m(G)$, is the set of all $\rho \in \Sigma^*$ such that $\delta(\rho) \in F$ (this means that $\delta(\rho)$ is of course defined). The *unmarked language* of G, denoted $L(G)$, is the set of all $\rho \in \Sigma^*$ such that $\delta(\rho) \in S$. Note that $L(G)$ is the prefix-closure of $L_m(G)$.

Now, consider event sets $\Sigma_{1O}, \Sigma_{2O}, \Sigma_{1C}, \Sigma_{2C} \subseteq \Sigma$. Σ_{iO} is the set of events that controller i observes, and Σ_{iC} is the set of events that it controls.

Each controller $C_i = (S_{iC}, q_{iC}, \delta_{iC}, \Sigma_{iO}, \Sigma_{iC}, \Lambda_i)$ is an *automaton with outputs*, where S_{iC} are the states, q_{iC} is the initial state, $\delta_{iC} : S_{iC} \times \Sigma_{iO} \to S_{iC}$ is the transition function (total), and $\Lambda_i : S_{iC} \to 2^{\Sigma_{iC}}$ is the output function (total). The meaning is that $\Lambda_i(s)$ is the set of events that the controller "allows" when it is in state s.

Given an automaton G and controllers C_1, C_2, we define the *conjunctively controlled system*, denoted $(G, C_1 \wedge C_2)$, to be the automaton $((S \times S_{1C} \times S_{2C}), (q, q_{1C}, q_{2C}), \delta_C, \Sigma, (F \times S_{1C} \times S_{2C}))$, where:

- $\delta_C((s, s_1, s_2), \alpha)$ is undefined if $\delta(s, \alpha)$ is undefined, or if for some $i = 1, 2$, $\alpha \in \Sigma_{iC} - \Lambda_i(s_i)$;
- otherwise, $\delta_C((s, s_1, s_2), \alpha) = (\delta(s, \alpha), s'_1, s'_2)$, where for each $i = 1, 2$, if $\alpha \notin \Sigma_{iO}$ then $s'_i = s_i$, otherwise $s'_i = \delta_{iC}(s_i, \alpha)$ (since δ_{iC} is total, $\delta_{iC}(s_i, \alpha)$ is always defined).

That is, $\rho \in L_m((G, C_1 \wedge C_2))$ if $\rho \in L_m(G)$, and each event in ρ is allowed by each controller (if any) that controls the event.

We say that the controllers are *non-blocking* if $L((G, C_1 \wedge C_2)) = \overline{L_m((G, C_1 \wedge C_2))}$, that is, from every reachable state, there is a path that leads to an accepting state.

Definition 5.1 (Decentralized Control Problem) *Given a finite-state automaton G over Σ, a regular language E, and event sets $\Sigma_{1O}, \Sigma_{2O}, \Sigma_{1C}, \Sigma_{2C} \subseteq \Sigma$, do there exist non-blocking controllers C_1 and C_2, such that $L_m((G, C_1 \wedge C_2)) \subseteq E$?*

Theorem 5.1 *The decentralized control problem is undecidable.*

Proof: Our proof uses the idea of reducing the control problem to an observation problem [14].

Consider again the PCP setting used in the proof of theorem 4.1 and let Σ_1, Σ_2 be as in that proof.

We will construct an automaton G and a language E, such that PCP has a solution iff there exist no controllers C_1, C_2 such that $L_m((G, C_1 \wedge C_2)) \neq \emptyset$ and $L_m((G, C_1 \wedge C_2)) \subseteq E$. First, define L to be the language $(a_1 w_1 + \cdots + a_n w_n)^+ + (a_1 u_1 + \cdots + a_n u_n)^+$ and K to be the language $(a_1 w_1 + \cdots + a_n w_n)^+$. Now, let G be the finite automaton accepting the following language:

$$L \cdot \Big(stop \ (t_1 + \cdots + t_n)^* \ stop_t \ (good + bad) \Big),$$

where $t_1, ..., t_n, stop, stop_t, good, bad$ are new letters not in Σ_1, Σ_2. We can assume that $L(G) = \overline{L_m(G)}$, that is, G is non-blocking.[2]

Let

$$\begin{aligned}
\Sigma_{1O} &= \Sigma_1 \cup \{t_1, ..., t_n\} \cup \{stop, stop_t\}, \\
\Sigma_{1C} &= \{good, bad\}, \\
\Sigma_{2O} &= \Sigma_2 \cup \{t_1, ..., t_n\} \cup \{stop, stop_t\}, \\
\Sigma_{2C} &= \{t_1, ..., t_n\} \cup \{stop_t\}.
\end{aligned}$$

The intuition is that G performs some "PCP string", then stops and "asks" the controllers whether the string was "good" (i.e., made up of w_i's) or "bad" (i.e., made up of u_i's). G allows C_2 to transmit its observation to C_1, which is modeled by the symbols $t_1, ..., t_n, stop_t$. C_1 will take the decision, based on its own observation and the one it receives from C_2.

Let E be the following language [3]:

$$\begin{aligned}
&\Big(L \cdot stop \ (t_1 + \cdots + t_n)^* \ stop_t \ good \Big) \\
&+ \\
&\Big((L - K) \cdot stop \ (t_1 + \cdots + t_n)^* \ stop_t \ bad \Big).
\end{aligned}$$

Assume PCP has no solution. Then, it can be shown that K is jointly observable w.r.t. L, Σ_1, Σ_2. Thus, by lemma 2.1, there exists a function $f : \Sigma_1^* \times \Sigma_2^* \to \{0, 1\}$ such that $\forall \rho \in L$, $f(\rho/_{\Sigma_1}, \rho/_{\Sigma_2}) = 1$ if $\rho \in K$ and $f(\rho/_{\Sigma_1}, \rho/_{\Sigma_2}) = 0$ if $\rho \notin K$.

Now, the controllers will function as follows. C_2 memorizes what it observes, until it sees *stop*. Let $a_{i_1} \cdots a_{i_k}$ be what C_2 has memorized.[4] Once *stop* is seen, C_2 "transmits" the sequence $t_{i_1} \cdots t_{i_k} \ stop_t$ (this is done by enabling t_{i_1}, then observing it and moving to another state, enabling t_{i_2}, and so on). C_2 does nothing

[2] Note that this is not a restricting assumption. Indeed, if there is a transition in G that leads to a state from which no accepting state can be reached, this transition can be removed without changing $L_m(G)$.

[3] Observe that $E \subseteq L_m(G)$.

[4] Note that the controllers may use an infinite set of states.

after transmitting $stop_t$. C_1 also memorizes what it observes, say σ_1, until it sees *stop*. Once *stop* is seen, C_1 starts memorizing separately what it receives from C_2, until $stop_t$ is received. Let $t_{i_1} \cdots t_{i_k}$ be what C_1 has received when it sees $stop_t$. Then, C_1 computes $f(\sigma_1, \sigma_2)$, where $\sigma_2 = a_{i_1} \cdots a_{i_k}$. If this turns out to be 1, then C_1 allows *good*, otherwise it allows *bad*.

The controllers are non-blocking. This is because, by assumption, G is non-blocking. Therefore, since nothing can be disabled until *stop* happens, from each reachable state before *stop* occurs, there is always a path that ends with *stop*. Now, once *stop* occurs, a number of t_i events will happen (possibly zero) and then $stop_t$ will be allowed. Finally, either *good* or *bad* will be enabled, reaching an accepting state.

We now show that $L_m((G, C_1 \wedge C_2))$ is non-empty and $L_m((G, C_1 \wedge C_2)) \subseteq E$. Either $a_1\, w_1\, stop\, t_1\, stop_t\, good$ or $a_1\, w_1\, stop\, t_1\, stop_t\, bad$ is in $L_m((G, C_1 \wedge C_2))$ (only one of them, depending on f), therefore $L_m((G, C_1 \wedge C_2))$ is not empty. Consider a word $\phi \in L_m((G, C_1 \wedge C_2))$: ϕ must be of the form $\phi = \rho\, stop\, \tau\, stop_t\, \kappa$, where $\rho \in L$, $\tau \in (t_1 + \cdots + t_n)^*$ and $\kappa \in (good + bad)$. Now, $\sigma_1 = \rho/_{\Sigma_1}$ and $\sigma_2 = \rho/_{\Sigma_2}$. If $\kappa = good$, it means that $f(\sigma_1, \sigma_2) = 1$, which implies $\rho \in K$, therefore $\phi \in E$. If $\kappa = bad$, it means that $f(\sigma_1, \sigma_2) = 0$, which implies $\rho \in L - K$, therefore again $\phi \in E$.

In the other direction, assume PCP has a solution $w_{i_1} \cdots w_{i_k} = u_{i_1} \cdots u_{i_k}$, and define $\rho = a_{i_1} w_{i_1} \cdots a_{i_k} w_{i_k}$, $\rho' = a_{i_1} u_{i_1} \cdots a_{i_k} u_{i_k}$. By the assumption that PCP has no trivial solution, we can deduce that $\rho \neq \rho'$: otherwise, w_{i_1} must be equal to u_{i_1}, since the a_i's are in a different alphabet than Σ_1. Observe that the string $\rho\, stop\, \tau\, stop_t\, good$ is in E, whereas $\rho'\, stop\, \tau\, stop_t\, good$ is not, since $\rho \neq \rho'$. Similarly, $\rho'\, stop\, \tau\, stop_t\, bad$ is in E, but $\rho\, stop\, \tau\, stop_t\, bad$ is not.

Now, suppose there exist non-blocking controllers C_1, C_2. Observe that $\rho/_{\Sigma_{iO}} = \rho'/_{\Sigma_{iO}}$, for $i = 1, 2$, thus, $\delta_i(q_{iC}, \rho/_{\Sigma_{iO}}) = \delta_i(q_{iC}, \rho'/_{\Sigma_{iO}})$, for $i = 1, 2$. That is, each C_i will be in the same state, whether ρ occurs or ρ'. Thus, in both cases, C_2 will transmit to C_1 the same t_i sequence, say, τ. We claim that, at some point, C_2 must transmit $stop_t$ and C_1 must allow at least one of *good*, *bad*. If $stop_t$ is never enabled, neither *good* nor *bad* will happen, and no accepting state can be reached (thus, controllers are blocking, which contradicts the assumption). The same is true if C_1 disables both *good* and *bad*. Now, C_1 cannot allow *good*, since then, the string $\rho'\, stop\, \tau\, stop_t\, good$ would be allowed, which is not in E. It cannot allow *bad* either, since then, the string $\rho\, stop\, \tau\, stop_t\, bad$ would be allowed, which is not in E. Therefore, controllers C_1 and C_2, such that $L_m((G, C_1 \wedge C_2)) \subseteq E$, cannot exist. ∎

6 Related Work

Observation and control has been studied by many researchers in the discrete-event systems community, e.g., see [11, 4, 21, 27, 7, 29]. In this section, we first review some of the definitions of observability (centralized and decentralized) and relate them to ours. We also discuss some existing results on decentralized control.

6.1 Centralized Observability Definitions

In [4] the authors introduce the notion of a (M, L)-*recognizable* language: K is (M, L)-recognizable if $K = L \cap M^{-1}(M(K))$. In the above definition, M is a *mask*, which is more general than a projection. In the case where M is a projection, (M, L)-recognizability is equivalent to centralized joint observability.

In [10] the authors introduce two different notions: *observability* and *normality*. Normality is equivalent to centralized joint observability, whereas observability is strictly weaker. However, observability is a necessary and sufficient condition for the centralized control problem: the reason for this is that the controller is allowed to disable some events even if it cannot observe them.

In summary, let us denote by $\mathsf{rec}_{[4]}$, $\mathsf{nor}_{[10]}$, $\mathsf{obs}_{[10]}$, $\mathsf{j-obs}$, the classes of recognizable, normal, observable, and jointly observable languages (in the centralized case), respectively. Then:

$$\mathsf{rec}_{[4]} = \mathsf{nor}_{[10]} = \mathsf{j-obs} \subset \mathsf{obs}_{[10]}$$

(To be able to compare with $\mathsf{rec}_{[4]}$, we assume that M is a projection.)

The inclusion is strict, as we illustrate with a simple example. Consider languages $L = \{\epsilon, a, b, ab\}$ and $K = \{\epsilon, b\}$. Let $\Sigma_1 = \{b\}$. Then, K is not jointly observable w.r.t. L, Σ_1, since ab and b give the same projection b, but $ab \notin K$ whereas $b \in K$. However, K is observable because the controller could disable a.

6.2 Decentralized Observability Definitions

The authors of [4] extend their definition of recognizable languages to the decentralized case: K is recognizable by k observers if K is (M_i, L)-recognizable for each $i = 1, ..., k$.

The authors of [21] introduce three notions of decentralized observability, namely, *weak decomposability*, *strong decomposability* and *co-observability*. K is weakly decomposable (w.r.t. two observers) if $K = L \cap P_1^{-1}(P_1(K)) \cap P_2^{-1}(P_2(K))$. K is strongly decomposable (w.r.t. two observers) if $K = L \cap (P_1^{-1}(P_1(K)) \cup P_2^{-1}(P_2(K)))$. The definitions extend directly to more than two observers. Strong decomposability implies weak decomposability. It is shown in [21] that strong decomposability implies co-observability.

We can show that the following is also true:

Theorem 6.1 *Strong decomposability is equivalent to decentralized recognizability, in case each mask M_i is a projection P_i.*

Proof: Observe that $K \subseteq L \cap \bigcap_i P_i^{-1}(P_i(K)) \subseteq L \cap \bigcup_i P_i^{-1}(P_i(K))$ holds, provided $K \subseteq L$.

First assume that K is strongly decomposable, i.e., $K = L \cap \bigcup_i P_i^{-1}(P_i(K))$. To show that K is recognizable, we need to show that $K = L \cap P_i^{-1}(P_i(K))$, for any i. Since $K \subseteq L \cap P_i^{-1}(P_i(K))$, for any i, it suffices to show that $L \cap P_i^{-1}(P_i(K)) \subseteq K$. Assume $x \in L \cap P_i^{-1}(P_i(K))$, for some i. This means that $x \in L \cap \bigcup_i P_i^{-1}(P_i(K))$, which implies $x \in K$.

Now assume K is recognizable, i.e., $K = L \cap P_i^{-1}(P_i(K))$, for any i. To show that K is strongly decomposable, we need to show that $K = L \cap \bigcup_i P_i^{-1}(P_i(K))$. Since $K \subseteq L \cap \bigcup_i P_i^{-1}(P_i(K))$, it suffices to show that $L \cap \bigcup_i P_i^{-1}(P_i(K)) \subseteq K$. Assume $x \in L \cap \bigcup_i P_i^{-1}(P_i(K))$. Then, there must exist an i such that $x \in L \cap P_i^{-1}(P_i(K))$, which implies $x \in K$. ∎

We can also prove the following:

Theorem 6.2 *Weak decomposability implies joint observability. The converse is not always true.*

Proof: Assume $K = L \cap \bigcap_i P_i^{-1}(P_i(K))$. If K was not jointly observable then there would exist $\rho, \rho' \in L$ such that $\rho \in K$, $\rho' \notin K$, yet $P_i(\rho) = P_i(\rho')$ for all i. Since $\rho \in K$ and $P_i(\rho) = P_i(\rho')$, we have $\rho' \in P_i^{-1}(P_i(K))$ for all i. But this means that $\rho' \in K$, which is a contradiction.

The converse does not hold in general, as we show with an example. Let $L = \{\rho, \rho_1, \rho_2\}$, $K = \{\rho_1, \rho_2\}$, $\rho = abc$, $\rho_1 = ab$, $\rho_2 = bc$, and $\Sigma_1 = \{a, b\}$, $\Sigma_2 = \{b, c\}$. Then, K is not weakly decomposable. Indeed, $P_1(\rho) = P_1(\rho_1) = ab$, therefore $\rho \in P_1^{-1}(P_1(K))$. Also, $P_2(\rho) = P_2(\rho_2) = bc$, therefore $\rho \in P_2^{-1}(P_2(K))$. But $\rho \notin K$. On the other hand, K is jointly observable: indeed, observer 2 can distinguish between ρ and ρ_1 (since $P_2(\rho) = bc \neq b = P_2(\rho_1)$) and observer 1 can distinguish between ρ and ρ_2 (since $P_1(\rho) = ab \neq b = P_1(\rho_2)$). ∎

Finally, we relate joint observability with co-observability. For the latter, we use the definition of [3]:

$$\forall s \in \overline{K}, \sigma \in \Sigma_c, (s\sigma \in L - \overline{K} \Rightarrow \exists i, P_i^{-1}(P_i(s)) \cap \overline{K} = \emptyset)$$

where Σ_c is the set of controllable events. To be able to relate the two definitions, we take the set of controllable events to be the set of all events, Σ. Then, we show the following:

Theorem 6.3 *Joint observability and co-observability are incomparable.*

Proof: We first show that joint observability does not imply co-observability. Consider L to be the prefix closure of the language $\{abcd, abd, bcd\}$. Let $K = L - \{abcd\}$ (observe that $\overline{K} = K$). Let $\Sigma_1 = \{a, b, d\}$ and $\Sigma_2 = \{b, c\}$. One can check that K is jointly observable w.r.t. L, Σ_1, Σ_2. Indeed, for any pair of strings $\rho = abcd, \rho' \in L$, either $P_1(\rho) \neq P_1(\rho')$ or $P_2(\rho) \neq P_2(\rho')$, which means either the first observer or the second observer can distinguish between the single erroneous string and a correct one. One the other hand, K is not co-observable. As a counter example to the definition of co-observability given above, take $s = abc$, $\sigma = d$, $s_1 = ab$ and $s_2 = bc$. Then, $s\sigma \in L - \overline{K}$, though $s_1\sigma \in K$ and $P_1(s_1) = P_1(s)$, and $s_2\sigma \in K$ and $P_2(s_2) = P_2(s)$. Intuitively, controller 1 does not know whether to disable d after observing ab, because abd is allowed but $abcd$ is not. Also, controller 2 does not know whether to disable d after observing bc, because bcd is allowed but $abcd$ is not.

The fact that co-observability does not imply joint observability can be shown by a simple example, similar to the one used to show that observability does not imply normality. Let $L = \{ab, ac\}$ and let $K = \{ab\}$. Assume both observers observe a. Then K is not jointly observable since ab, ac give same observations, yet only ab is legal. However, K is co-observable, because the controllers can disable c. ∎

We summarize the above results as follows (all inclusions are strict):

$$\mathsf{rec}_{[4]} = \mathsf{strong}_{[21]} \subset \mathsf{weak}_{[21]} \subset \mathsf{j-obs},$$
$$\mathsf{strong}_{[21]} \subset \mathsf{co-obs}_{[21]},$$
$$\mathsf{j-obs} \neq \mathsf{co-obs}_{[21]}.$$

6.3 Decentralized Control

In [21] it is shown that, given regular language E such that $\emptyset \neq E \subseteq L_m(G)$, checking whether there exist controllers C_1, C_2 such that $L_m(G, C_1 \wedge C_2) = E$ is decidable. They also show that, given regular languages A and E such that $A \subseteq E \subseteq L_m(G)$, checking whether there exist controllers C_1, C_2 such that $A \subseteq L(G, C_1 \wedge C_2) \subseteq E$ is decidable. Note that $L_m(G, C_1 \wedge C_2)$ is checked in the first problem, whereas $L(G, C_1 \wedge C_2)$ (the *unmarked* language, which is prefix-closed) in the second problem. In this paper we have shown that checking $L_m(G, C_1 \wedge C_2) \subseteq E$ is undecidable. Undecidability results for decentralized control in a slightly different setting (ω-languages) and using a different proof have been reported also in [9].

We believe that the undecidable case, namely, checking that $L_m(G, C_1 \wedge C_2) \subseteq E$, with E non-prefix-closed, is one of the most interesting in practice. Indeed, suppose we want to model the requirement $(pg)^*$, that is, every p is followed by a g before the next p occurs. This is typical for a communication protocol: p models the sending client giving a message to the protocol and g the protocol delivering the message to the receiving client (e.g., see [19, 14]). Since there is likely to be other events in the plant than just p, g, say, Σ, we should define $E = (\Sigma^* + (p\,\Sigma^* g))^*$.

Now, if we make E prefix-closed, then we have to define some $A \neq \emptyset$, otherwise the controllers might allow behaviors where g never occurs after p. But it is not easy to define such an A, without requiring the controllers to allow too many behaviors. For instance, we cannot make $A = E$, because in that case *any* behavior in Σ^* must be allowed by the controllers, which is unrealistic to expect.[5] Another possibility would be to try and solve the problem $L_m(G, C_1 \wedge C_2) = E'$, for some other E', non-prefix-closed in general. Again, however, it is very difficult to find such an E': in fact, since we request the language of the controlled system to be exactly E', knowing E' means knowing exactly what the behaviors of the controllers should be!

6.4 Other Related Work

The problem of decentralized control has been studied by many researchers in the discrete-event systems community, e.g., see [11, 4, 21, 27, 7, 1, 17, 29].

[11] study the problem of decentralized control with respect to "local" specifications, e.g., two supervisors S_1 and S_2 are synthesized independently with respect to local specifications ϕ_1 and ϕ_2, and their combined control on the plant results in the behavior $\phi_1 \wedge \phi_2$. This may be called "modular" controller synthesis and essentially has to do with breaking the problem into smaller ones.

[4] study the problem of decentralized control with respect to "global" specifications, as we do in this paper. Their setting is slightly more general than ours, in that they use a *masking* function, whereas we use a *projection*, which is less general. The authors obtain necessary and sufficient conditions for the existence of a controller which constrains the plant to a behavior K (a language).

[7] study decentralized control for a more general class of controllers where the composition with the plant is not necessarily synchronous. [13, 29] consider other ways of combining control actions with the plant, including "fusion by union" of events.

[27] and [1] consider the problem of decentralized control where the controllers are more powerful, in the sense that they can exchange information during the execution of the plant. [17] consider the same problem and develop a strategy to minimize communication between the controllers.

On a slightly different setting, [12] studies the (un)decidability and complexity of the problems of distributing a centralized program on a decentralized processor architecture, or synthesizing a decentralized program from scratch.

In [14] we give examples where infinite-state decentralized observers and controllers exist, but no finite-state ones, as well as necessary and sufficient conditions for finite-state observers to exist. We also model the Alternating Bit Protocol (ABP) as a decentralized control problem and study different variants of the protocol.

[5] We could define an interesting variant of the problem, where the controllers must ensure $A \subseteq P(L(G, C_1 \wedge C_2)) \subseteq E$, where P is a projection to an appropriate alphabet (or we could even have two projections, one for A and one for E). Unfortunately, this problem is undecidable as well (slight modification of the proof of theorem 5.1).

7 Discussion and Open Questions

It may seem surprising that joint observability is undecidable, especially since strong and weak decomposability are decidable and co-observability is shown to be decidable in [21] (a polynomial algorithm is given in [18]). An open question is whether joint observability with respect to a prefix-closed regular language and exactly two observers is decidable.

The definitions and undecidability results of this paper can be extended to ω-languages. However, it is not clear what the physical interpretation of such a definition would be: since behaviors are infinite, when do observers get together in order to decide?

Acknowledgments

I thank Raja Sengupta, Anuj Puri, Pravin Varaiya, David de Frutos, Stephane Lafortune, Albert Benveniste, Karen Rudie and John Thistle.

References

[1] G. Barrett and S. Lafortune, "On the Synthesis of Communicating Controllers with Decentralized Information Structures for Discrete-Event Systems". IEEE Conference on Decision and Control 1998.

[2] J. Berstel, "Transductions and context-free languages", 1979.

[3] C.G. Cassandras and S. Lafortune, "Introduction to Discrete Event Systems", Kluwer Academic Publishers, September 1999.

[4] R. Cieslak, C. Desclaux, A.S. Fawaz and P. Varaiya. "Supervisory control of discrete-event processes with partial

observations". IEEE Trans. Automatic Control, vol. 33, pp. 249-260, 1988.

[5] A. Church, "Logic, arithmetic and automata", *Proceedings of the International Congress of Mathematics*, 1963.

[6] M. Garey and D. Johnson. *Computers and Intractability: a guide to the theory of NP-completeness.* Freeman, 1979.

[7] R. Kumar, M.A. Shayman, "Centralized and Decentralized Supervisory Control of Nondeterministic Systems Under Partial Observation", SIAM Journal on Control and Optimization, vol. 35(2), pp. 363-383, 1997.

[8] O. Kupferman and M. Vardi, "Synthesis with Incomplete Information", ICTL 1997.

[9] H. Lamouchi and J. Thistle, "Effective Control Synthesis for DES Under Partial Observations", Conference on Decision and Control, December 2000.

[10] F. Lin and W.M. Wonham. "On observability of discrete-event systems", Information Sciences, vol. 44, pp. 173-198, 1988.

[11] F. Lin and W.M. Wonham. "Decentralized supervisory control of discrete-event systems", Information Sciences, vol. 44, pp. 199-224, 1988.

[12] A. Pnueli and R. Rosner, "Distributed Reactive Systems are Hard to Synthesize", Proceedings of the 31th IEEE Symposium Foundations of Computer Science (FOCS 1990), pages 746-757, 1990.

[13] J.H. Prosser, M. Kam, H.G. Kwatny, "Decision Fusion and Supervisor Synthesis in Decentralized Discrete Event Systems", American Control Conference, 1997.

[14] A. Puri, S. Tripakis and P. Varaiya. "Problems and Examples of Decentralized Observation and Control for Discrete Event Systems", SCODES 2001.

[15] M. Rabin, Automata on Infinite Objects and Church's Problem, vol. 13 in Regional conference series in Mathematics, 1972.

[16] P.J.G. Ramadge and W.M. Wonham, "The control of discrete event systems", Proceedings of the IEEE, January, 1989.

[17] K. Rudie, S. Lafortune and F. Lin. "Minimal Communication in a Distributed Discrete-Event Control System". Proc. of the American Control Conference, 1999.

[18] K. Rudie and J.C. Willems, "The Computational Complexity of Decentralized Discrete-Event Control Problems", IEEE Transactions on Automatic Control, 40(7), July 1995.

[19] K. Rudie and W.M. Wonham. "Supervisory control of communicating processes", Protocol Specification, Testing and Verification X, Elsevier, IFIP, 1990.

[20] K. Rudie and W.M. Wonham. "The infimal prefix-closed and observable superlanguage of a given language", Systems and Control Letters 15, 361–371, 1990.

[21] K. Rudie and W.M. Wonham. "Think globally, act locally: Decentralized supervisory control", IEEE Trans. Automatic Control, vol. 37, pp. 1692-1708, 1992.

[22] K. Rudie and W.M. Wonham. "Protocol Verification Using Discrete-Event Systems", 31st Conference on Decision and Control, 1992.

[23] W. Thomas, "Automata on Infinite Objects", Formal Models and Semantics, Volume B of Handbook of Theoretical Computer Science, Elsevier Science, 1990.

[24] S. Tripakis, "Undecidable Problems in Decentralized Observation and Control", To appear in Conference on Decision and Control 2001.

[25] J.N. Tsitsiklis and M. Athans, "On the Complexity of Decentralized Decision Making and Detection Problems", IEEE Transactions on Automatic Control, Vol. 30, No.5, 1985, pp. 440-446.

[26] J.N. Tsitsiklis, "On the Control of Discrete Event Dynamical Systems", Mathematics of Control, Signals and Systems, Vol. 2, No. 2, 1989, pp. 95-107.

[27] K.C. Wong and J.H. van Schuppen, "Decentralized supervisory control of discrete-event systems with communication", WODES '96, IEE, 1996.

[28] H. Wong-Toi and D. L. Dill. "Synthesizing processes and schedulers from temporal specifications", Proceedings of the 1990 Computer-Aided Verification Workshop, Lecture Notes in Computer Science, Volume 531, Springer-Verlag, 1991.

[29] T. Yoo and S. Lafortune, "New Results on Decentralized Supervisory Control of Discrete-Event Systems", IEEE Conference on Decision and Control 2000.

[30] Y. Willner, M. Heymann, "Language convergence in controlled discrete-event systems", IEEE Transactions on Automatic Control, Vol. 40, No 4, pp. 616-627, 1995.