Proceedings of the
44th IEEE Conference on Decision and Control, and
the European Control Conference 2005
Seville, Spain, December 12-15, 2005

Decentralized Observation Problems

Stavros Tripakis

***Abstract*— A fundamental observation problem is, given a model of the system to be observed and a specification of the property to be observed, to check whether the property is observable (i.e., the observer can resolve potential ambiguities due to partial observation capabilities) and if so to (automatically) synthesize an observer.**

We consider a simple case of the above setting, namely, where the system model and specifications are expressed as regular languages over a finite alphabet. In this setting, and in the case of one observer (centralized observation), both checking observability and synthesizing an observer can be done automatically, and the observer is guaranteed to be finite-state. The situation is more complicated in the case of more than one observers (decentralized observation). Here, many definitions are possible, depending on where the decisions are taken, as well as on whether the observers are required to be finite-state. We examine some of these possibilities, surveying recent results on the topic and providing some new results as well.

I. Introduction

In this paper we study problems of decentralized observation. Such problems arise naturally in contexts of decentralized control, where a number of agents control a single plant, each observing (and acting upon) only part of the plant. Decentralized control often "hides" a decentralized observation problem, since the agents must infer, based on a set of partial observations, facts about the original behavior of the plant [1]. This remains true independently of whether the agents are allowed to communicate or not [2].

Decentralized observation problems are also interesting for their own sake. For instance, when monitoring a large, distributed system such as a network, a vehicle controller consisting of many components ("electronic control units" or ECUs), a manufacturing plant, etc., one usually relies on local monitors which collect information at different parts of the system. This information can be gathered and analyzed off-line at a central point. Alternatively, the information can be processed on-line and, possibly when it shows that some properties have been violated, may result in corrective actions being taken.

Even for finite sets of observations, decentralized observation problems are inherently difficult from a complexity point of view [3]. Notice that it is often impractical to endow the local monitors with advanced capabilities such as communication and clock synchronization, which would change the distributed nature of the problem in a significant way.

Work partially supported by CNRS STIC project "CORTOS" and IST Network of Excellence "ARTIST2".

CNRS and Verimag Laboratory, Centre Equation, 2, avenue de Vignate, 38610 Gières, France, `tripakis@imag.fr`.

In this paper, we consider a setting with a possibly infinite set of observations. Still, the setting is one of the simplest possible. The system under observation (or *plant*) is modeled as a regular language L over some finite alphabet Σ. A letter in Σ can be seen as an event generated by the plant and a finite word in L can be seen as a behavior of the plant. Observer i can only observe a subset of events $\Sigma_i \subseteq \Sigma$. Thus, the observation that observer i collects from a behavior $\rho \in L$ is the *projection* of ρ onto Σ_i. Notice that Σ_i are not necessarily disjoint. A regular language $K \subseteq L$ models a set of *distinguished* behaviors of the plant. For example, behaviors in K may be those satisfying a given requirement while those in $L - K$ do not. The objective of the observation process is to determine whether the plant produced a behavior in K or in $L - K$. Thus, we have essentially a property monitoring problem.

Obviously, it is not always possible to make a correct decision, based only on partial observations. For example, if $\Sigma = \{a, b\}$, $\Sigma_1 = \{a\}$, $\Sigma_2 = \{b\}$, $L = \{ab, ba\}$ and $K = \{ab\}$, then it is impossible, based on the observations (a, b), to determine whether ab or ba happened. Thus, one concern is to *check observability*. When observability holds, the next concern is to *synthesize* the observers automatically.

In the centralized case, that is, the case of one observer, both checking observability and synthesizing the observer can be done algorithmically, using standard techniques from finite automata theory. This is not always true in the decentralized case. Checking observability is sometimes undecidable [4]. Existence of observers does not always imply existence of finite-state observers [5].

In this paper, we consider four versions of the decentralized observation problem, depending on two criteria. First, on the memory requirements of the observers, and second, on where the monitoring decisions are being made. According to the first criterion, we distinguish between *unbounded-memory* observation, where each observer is allowed to record the entire observed sequence, and *finite-memory* observation, where the observers are required to be finite-state automata. According to the second criterion, we distinguish between *joint* observation, where the observers send their observations to a central decision point, and *local* observation, where each observer makes a local one-bit decision and the local decisions are combined into a global decision by a boolean function.

In the rest of the paper, we examine these four problems, relating them to one another and studying their decidability. Most of the results we provide for joint observation are taken from [4], [1], [6]. The results on local observation and its comparison to joint observation are, to our knowledge, new.

Related work

The problems we study in this paper are related to various distributed synthesis problems studied in concurrency theory. In particular, the joint observation problems are related to trace theory [7] and rational-relation theory [8], as well as the general problem of synthesis of a distributed system starting from a "centralized" specification (e.g., see [9]). See [6] for a detailed discussion and list of references.

The observation problems are also related to decentralized versions of the *fault diagnosis* problem for discrete event systems (DES) [10], [11], [12]. Fault diagnosis is different from the observation problems we study in this paper in the sense that in fault-diagnosis the observers are only required to detect a fault after a certain (bounded) delay. Still, some fundamental issues are common to both problems. For instance, the undecidability techniques of [4] have been used to show undecidability of a decentralized fault-diagnosis problem in [13]. Also, local observation with respect to conjunction is related to what is termed "F-codiagnosability" in [11] and "co-diagnosability" in [12]. Local observation with respect to disjunction is related to the notion of "NF-codiagnosability" [11]. Notice that no distinction between finite- and unbounded-memory observers is made in these two papers. It is worth mentioning that [11] also consider observers that can make "conditional" decisions, such as "Fault if nobody says No Fault" or "No Fault if nobody says Fault", and show that these are more powerful than F- or NF-codiagnosability.

Finally, as mentioned in the introduction, decentralized observation is related to decentralized control. In fact, unbounded-memory joint observation can be reduced to decentralized control without communication [2]. This may seem surprising, since on the one hand joint observation is done in two phases (collection and decision) and the decision phase is centralized, whereas on the other hand the decentralized control problem used in [2] is on-line, without *a priori* communication between the controllers. The paradox is resolved by the fact that there exist plants which allow the controllers to communicate *indirectly*, by enabling and disabling plant transitions. Thus, joint observation can be simulated by on-line decentralized control without communication.

II. Setting and Notation

Let us first establish the setting which will be used throughout the paper. We consider a finite alphabet Σ. Each letter in Σ represents an event that may be produced by the system under observation. Σ^* denotes, as usual, the set of all finite words over Σ, including the empty word ϵ. The behaviors of the system under observation are modeled as a regular language[1] $L \subseteq \Sigma^*$. A regular language $K \subseteq L$ models the property to be observed. In other words, we want to know whether a behavior produced by the system is in K or not.[2]

We model partial observation by considering sub-alphabets $\Sigma_i \subseteq \Sigma$, for $i = 1, ..., n$. The understanding is that observer i can only observe events in Σ_i. We set $\Sigma_o = \bigcup_{i=1,\dots,n} \Sigma_i$. Notice that Σ_i need not be disjoint and Σ_o need not be equal to Σ. Given a behavior $\rho \in \Sigma^*$, observer i observes the *projection* of ρ to Σ_i, denoted $P_{\Sigma_i}(\rho)$, or $P_i(\rho)$ for simplicity.[3] The projection is obtained simply by "erasing" from ρ all events not in Σ_i. For example, if $\Sigma = \{a, b, c\}$ and $\Sigma_1 = \{a, c\}$, then $P_{\Sigma_1}(abbcbab) = aca$.

We note some properties that are going to be useful to prove the results of this paper. First, the following holds for any sets A, B, C:

$$A = B \cap C \Leftrightarrow A \subseteq B \wedge A \subseteq C \wedge (B - A) \cap C = \emptyset \quad (1)$$

The following are properties of any projection P and its inverse. $\overline{A}$ denotes the complement of set A.

$$A \subseteq P^{-1}(P(A)) \quad (2)$$
$$P(A \cup B) = P(A) \cup P(B) \quad (3)$$
$$P(A \cap B) \subseteq P(A) \cap P(B) \quad (4)$$
$$P^{-1}(A \cup B) = P^{-1}(A) \cup P^{-1}(B) \quad (5)$$
$$P^{-1}(A \cap B) = P^{-1}(A) \cap P^{-1}(B) \quad (6)$$
$$P^{-1}(\overline{A}) = \overline{P^{-1}(A)} \quad (7)$$

III. A Centralized Observation Problem

Definition 1: Let $K \subseteq L \subseteq \Sigma^*$. We say that K *is centrally observable with respect to* L *and* Σ_1 iff there exists a total function $f : \Sigma_1^* \to \{0, 1\}$ such that

$$\forall \rho \in L \,.\, \rho \in K \Leftrightarrow f(P_1(\rho)) = 1.$$

It can be easily seen that f does not always exist, because of ambiguities resulting from partial observability.

Example 1: Take $\Sigma = \{a, b\}$, $\Sigma_1 = \{a\}$, $L = \{a, ab\}$ and $K = \{ab\}$. Here, we have two behaviors, a and ab, which must be distinguished, since $ab \in K$ while $a \notin K$. But they cannot be distinguished by an observer observing only a, since $P_1(ab) = P_1(a) = a$.

Checking whether f exists in the centralized case is easy: f exists iff $P_1(K) \cap P_1(L - K) = \emptyset$. The latter condition can be checked since regular languages are closed under intersection, complementation and projection, and checking their emptiness is decidable. When the above condition holds, f can be synthesized. In fact, f can be represented as a finite-state deterministic automaton over Σ_1, the automaton recognizing $P_1(K)$. This also implies that, in the centralized case, existence of an observer implies existence of a finite-state observer.

IV. Decentralized Observation Problems

A. Definitions

Definition 1 can be extended to more than one observers in different ways. We next provide a set of definitions and comment on their meaning.

[1] a language recognized by a finite-state automaton

[2] The assumption $K \subseteq L$ is not limiting. For any property K', we can restrict our attention to $K = K' \cap L$, which satisfies the assumption.

[3] More generally, we could associate to each observer a *mask* function [14], $M_i : \Sigma \to O_i \cup \{\tau\}$, with the meaning that event $a \in \Sigma$ is either totally unobservable to observer i (when $M_i(a) = \tau$) or is perceived as $o \in O_i$ (when $M_i(a) = o$). This would somewhat complicate the discussion without affecting the results in a substantial way.

Perhaps the most straightforward extension of Definition 1 to the decentralized case is Definition 2 given below. Notice that for $n = 1$ the two definitions coincide.

Definition 2 (Joint unbounded-memory observation): We say that K *is jointly observable with respect to* L *and* $(\Sigma_1, ..., \Sigma_n)$ iff there exists a total function $f : \Sigma_1^* \times \cdots \times \Sigma_n^* \to \{0, 1\}$, such that

$$\forall \rho \in L \,.\, \rho \in K \Leftrightarrow f\big(P_1(\rho), \cdots, P_n(\rho)\big) = 1.$$

This type of observation may be called "joint" [1] or "two-phase" [6] observation, because function f plays the role of a "centralized decision point", where all observations are sent. It is the role of this central point to decide whether the original behavior ρ was in K or not. This type of observation uses unbounded memory because each local observer i must "record" the entire observed sequence $P_i(\rho)$, and the latter can have arbitrary length. We next consider the case where the observers are required to have finite memory.

Definition 3 (Joint finite-memory observation): We say that K *is finitely jointly observable with respect to* L *and* $(\Sigma_1, ..., \Sigma_n)$ iff there exist finite-state deterministic automata[4] A_i over Σ_i, $A_i = (S_i, s_i^0, t_i)$, $i = 1, ..., n$, and a total function $f : S_1 \times \cdots \times S_n \to \{0, 1\}$, such that

$$\forall \rho \in L \,.\, \rho \in K \Leftrightarrow f\big(t_1(P_1(\rho)), \cdots, t_n(P_n(\rho))\big) = 1.$$

Here observer i is represented as a finite-state deterministic automaton $A_i = (S_i, s_i^0, t_i)$, where S_i is the set of states, s_i^0 the initial state and $t_i : S_i \times \Sigma_i \to S_i$ is the (total) transition function. There is still a central decision point, however, observer i does not record the entire observation $P_i(\rho)$, it only records the final state after the observation is received, $t_i(P_i(\rho)) \in S_i$; and the set of states S_i is finite.

The last two definitions involve a central decision point. Let us now consider another definition which involves a set of local decision points f_i, one per observer. Each makes a decision of one bit. There is still the need, however, to combine the local decisions into a global decision: this role is played by an n-ary boolean function B.

Definition 4 (Local unbounded-memory observation): Let $B : \{0, 1\}^n \to \{0, 1\}$ be an n-ary boolean function. We say that K *is locally observable with respect to* L*,* B *and* $(\Sigma_1, ..., \Sigma_n)$ iff there exist total functions $f_i : \Sigma_i^* \to \{0, 1\}$, for $i = 1, ..., n$, such that

$$\forall \rho \in L \,.\, \rho \in K \Leftrightarrow B\big(f_1(P_1(\rho)), ..., f_n(P_n(\rho))\big) = 1.$$

Notice that for $n = 1$ and B the identity or negation function, we obtain essentially Definition 1. Also note that the above definition assumes observers with unbounded memory. As in the case of joint observation, we can give an alternative definition requiring finite memory.

Definition 5 (Local finite-memory observation): Let $B : \{0, 1\}^n \to \{0, 1\}$ be an n-ary boolean function. We say that K *is finitely locally observable with respect to* L*,* B *and* $(\Sigma_1, ..., \Sigma_n)$ iff there exist finite-state deterministic automata A_i over Σ_i, $A_i = (S_i, s_i^0, t_i)$, $i = 1, ..., n$, and total functions $f_i : \Sigma_i^* \to \{0, 1\}$, for $i = 1, ..., n$, such that

$$\forall \rho \in L. \rho \in K \Leftrightarrow B\big(f_1(t_1(P_1(\rho))), ..., f_n(t_n(P_n(\rho)))\big) = 1.$$

It should be noted that K is finitely observable in a trivial way when $K = \emptyset$ or $K = L$. Also, if an observer cannot observe anything, i.e., $\Sigma_i = \emptyset$, then this observer can be ignored, since it does not bring any information. On the other hand, if there exists an observer i that can observe any event that another observer may observe, i.e., $\forall j, \Sigma_j \subseteq \Sigma_i$, then the problem degenerates to a centralized observation problem, since all observers except the i-th one are redundant. In the rest of the paper, we will be assuming that we are not in one of these degenerate cases. Finally, note that, in all above definitions, the order of Σ_i does not matter. So, for instance, K is jointly observable w.r.t. L and (Σ_1, Σ_2) iff K is jointly observable w.r.t. L and (Σ_2, Σ_1).

B. Comparison and examples

We proceed to compare the above notions. We use the acronyms JO, FJO, LO$_B$ and FLO$_B$, respectively, for joint observability, finite joint observability, local observability w.r.t. B and finite local observability w.r.t. B.

By definition, FJO implies JO and, for any B, FLO$_B$ implies LO$_B$, FLO$_B$ implies FJO and LO$_B$ implies JO. For instance, to see that LO$_B$ implies JO, observe that the global function f required for JO can be defined as the composition of the boolean function B and the local functions f_i required for LO$_B$.

JO does not generally imply FJO as the following example shows.

Example 2: Let $\Sigma = \{a, b\}$, $\Sigma_1 = \{a\}$, $\Sigma_2 = \{b\}$, $K = (ab)^*$ and $L = (ab)^*b^*$.[5] It can be checked that K is jointly observable w.r.t. L and (Σ_1, Σ_2). Indeed, given observations a^k and b^l, it suffices to check whether $k = l$: if so, the original behavior was in K, otherwise not. Since k and l are unbounded, this cannot be checked with finite memory.

This example also shows that JO may not imply LO$_B$, for any B. Indeed, there is no B such that the above K is locally observable w.r.t. L, B and (Σ_1, Σ_2): because f_i can only transmit an information of one bit each, whereas observability of K requires two unbounded integers k and l.

For similar reasons, FJO may not imply FLO$_B$ either. The one-bit information provided by functions f_i to the central decision point in the case of FLO, may not be enough, whereas more bits would suffice. Here is an example where this occurs.

Example 3: Let $\Sigma = \{a, b\}$, $\Sigma_1 = \{a\}$ and $\Sigma_2 = \{b\}$. Let $L = \{\epsilon, a, aa, b, ab, aab\}$ and $K = \{\epsilon, aa, b\}$. Since $\epsilon \in K$ but $a \notin K$, f_1 must distinguish between the projections of these words to Σ_1, which in this case are the words themselves. Thus, $f_1(\epsilon) \neq f_1(a)$. Similarly, $f_1(aa) \neq f_1(a)$. Since f_1 may assume only two possible values, we conclude that $f_1(\epsilon) = f_1(aa)$. On the other hand, $b \in K$ but

[4] As usual, a deterministic automaton over Σ is a tuple (S, s^0, t), where S is a set of states, $s^0 \in S$ is the initial state, and $t : S \times \Sigma \to S$ is the transition function (assumed to be total). Given $\rho \in \Sigma^*$, $t(\rho)$ denotes the unique state the automaton moves to having consumed its input ρ.

[5] We use standard computer-science notation for regular expressions, so $K = (ab)^* = \{\epsilon, ab, abab, ...\}$ and $L = (ab)^*b^*$ is obtained by appending at the end of each word of K an arbitrary number of b's (possibly zero).

$aab \notin K$. Since the projection of these two words to Σ_2 is the same, namely b, f_2 cannot distinguish them. Thus, f_1 must provide the necessary information to distinguish them, thus, $f_1(P_1(b)) \neq f_1(P_1(aab))$, or $f_1(\epsilon) \neq f_1(aa)$, which contradicts the previous conclusion. Therefore, there is no B such that K is finitely locally observable w.r.t. L, B and (Σ_1, Σ_2).

We proceed to compare LO with FLO. First, let us note that Definition 4 can be reformulated in a set-theoretic way. Indeed, the functions f_i can be viewed as characteristic functions of corresponding sets $F_i \subseteq \Sigma_i^*$. Also, the boolean function B can be viewed as a function on sets, for example $\bigwedge$ as $\bigcap$, $\bigvee$ as $\bigcup$, and so on. Then, local observability holds when K can be expressed as a "boolean" combination of the inverse projections of F_i, or

$$K = L \cap B(P_1^{-1}(F_1), ..., P_n^{-1}(F_n)).$$

Finite local observability holds when each F_i is a regular language.

The question arises, then, does existence of F_i imply existence of regular F_i? We give a positive answer in the case where B is either the conjunction function $\bigwedge$ or the disjunction function $\bigvee$. In the case of general B the question remains open.

Lemma 1: K is locally observable w.r.t. L, $\bigwedge$ and $(\Sigma_1, ..., \Sigma_n)$ iff

$$K = L \cap \bigcap_i P_i^{-1}(P_i(K)) \tag{8}$$

Proof: From the reformulation above, what we must show is that $\exists F_1, ..., F_n \,.\, K = L \cap \bigcap_i P_i^{-1}(F_i)$ iff Equality (8) holds. The $\Leftarrow$ direction is trivial. For the $\Rightarrow$ direction, we shall use Equivalence (1). First, $K \subseteq \bigcap_i P_i^{-1}(P_i(K))$ holds, because of Property (2).

It remains to show that $(L - K) \cap \bigcap_i P_i^{-1}(P_i(K)) = \emptyset$. Suppose not, that is, suppose there is $\rho \in L - K$ such that $P_i(\rho) \in P_i(K)$, for any i. We claim that $P_i(K) \subseteq F_i$, for any i. Let $\pi \in P_i(K)$. So, there exists $\rho' \in K$ such that $P_i(\rho') = \pi$. Since $\rho' \in K$, from the hypothesis, $\rho' \in P_i^{-1}(F_i)$, for any i. Thus, $P_i(\rho') \in F_i$. The claim implies $P_i(\rho) \in F_i$, for any i, thus, from the hypothesis, $\rho \in K$. Contradiction. ∎

Lemma 2: K is locally observable w.r.t. L, $\bigvee$ and $(\Sigma_1, ..., \Sigma_n)$ iff

$$K = L \cap \bigcup_i P_i^{-1}(\overline{P_i(L - K)}) \tag{9}$$

Proof: From the reformulation above, what we must show is that $\exists F_1, ..., F_n \,.\, K = L \cup \bigcap_i P_i^{-1}(F_i)$ iff Equality (9) holds. The $\Leftarrow$ direction is trivial. For the $\Rightarrow$ direction, we shall use Equivalence (1). First, $(L - K) \cap \bigcup_i P_i^{-1}(\overline{P_i(L-K)}) = \emptyset$ holds, since $(L-K) \cap P_i^{-1}(\overline{P_i(L-K)})$ equals $(L-K) \cap \overline{P_i^{-1}(P_i(L-K))}$ (by Property (7)), which is empty for any i.

It remains to show that $K \subseteq \bigcup_i P_i^{-1}(\overline{P_i(L-K)})$, or equivalently, $K \cap \bigcap_i P_i^{-1}(P_i(L-K)) = \emptyset$. Suppose not, that is, suppose there is $\rho \in K$ such that $P_i(\rho) \in P_i(L-K)$, for any i. Since $\rho \in K$, from the hypothesis, there is some j such that $P_j(\rho) \in F_j$. Since $P_j(\rho) \in P_j(L-K)$, there is some $\rho' \in L - K$ such that $P_j(\rho') = P_j(\rho)$. Thus, $P_j(\rho') \in F_j$ and, from the hypothesis, $\rho' \in K$. Contradiction. ∎

$$\begin{array}{ccc}
\text{FJO} & \begin{array}{c}\Rightarrow\\ \not\Leftarrow\end{array} & \text{JO} \\
\Uparrow \not\Downarrow & & \Uparrow \not\Downarrow \\
\text{FLO}_B & \begin{array}{c}\Rightarrow\\ \Leftarrow\end{array} & \text{LO}_B \\
 & \begin{array}{c}\text{for } \wedge, \vee\\ \text{otherwise?}\end{array} &
\end{array}$$

Fig. 1. Implications between different observation problems

In order to illustrate the above two results, we provide some examples.

Example 4: Let $\Sigma = \{a, b, c, d\}$, $\Sigma_1 = \{a, d\}$, $\Sigma_2 = \{b, c\}$. Let $L = \{\epsilon, a, c, ab, cd\}$ and $K = \{\epsilon, a, c\}$. Then, Equality (8) holds, thus, K is finitely locally observable with respect to $\bigwedge$. Indeed, it suffices for observer 1 to issue 0 when d is observed, 1 otherwise, and for observer 2 to issue 0 when b is observed, 1 otherwise.

Example 5: Next, let us provide an example which shows that F_i cannot be chosen to be $P_i(K)$ in general, and this is in the case B is the disjunction function, conforming to Lemma 2. Let $\Sigma = \{a, b, c\}$, $\Sigma_1 = \{a, b\}$ and $\Sigma_2 = \{b, c\}$. Let $L = \{ab, bc, abc\}$ and $K = \{ab, bc\}$. Let $B = \bigvee$. Then $K_1 = P_1(K) = \{ab, b\}$ and $K_2 = P_2(K) = \{b, bc\}$. Now, $L \cap (P_1^{-1}(K_1) \cup P_2^{-1}(K_2)) = L \neq K$. But if we take $F_1 = F_2 = \{b\}$ then we have $L \cap (P_1^{-1}(F_1) \cup P_2^{-1}(F_2)) = \{ab\} \cup \{bc\} = K$.

Example 6: The fact that Σ_i are not disjoint in the above example is coincidental. Here is another example where Σ_i are disjoint. Let $\Sigma = \{a, b\}$, $\Sigma_1 = \{a\}$ and $\Sigma_2 = \{b\}$. Let $L = \{a, ab\}$ and $K = \{ab\}$. Let $B = \bigvee$. Then $K_1 = P_1(K) = \{a\}$ and $K_2 = P_2(K) = \{b\}$. Now, $L \cap (P_1^{-1}(K_1) \cup P_2^{-1}(K_2)) = L \neq K$. But if we take $F_1 = \emptyset$ and $F_2 = \{b\}$ then we have $L \cap (P_1^{-1}(F_1) \cup P_2^{-1}(F_2)) = \emptyset \cup \{ab\} = K$.

The results of this comparison are summarized in Fig. 1.

Lemmata 1 and 2 provide necessary and sufficient conditions for local observability, in the cases where B is a conjunction or a disjunction. These conditions imply that, in these cases, the two types of local observability, unbounded- or finite-memory, coincide and are decidable (recall that we consider regular languages). They also imply that, when observability holds, automatic synthesis of observers is possible: in the conjunction (resp. disjunction) case, the i-th observer can be taken to be the finite-state automaton recognizing $P_i(K)$ (resp. $\overline{P_i(L-K)}$).

In the rest of the paper, we focus on the joint observation problems. The results of the rest of the paper are taken from [4], [1], [6].

V. Necessary and sufficient conditions

A necessary and sufficient condition for unbounded-memory joint observability is the following:

$$\forall \rho \in K, \rho' \in L - K \,.\, \exists i \,.\, P_i(\rho) \neq P_i(\rho') \tag{10}$$

The condition states that K is not observable iff there exist two behaviors in L which yield the same observations, yet one is in K and the other is not. This condition cannot be verified algorithmically, as we shall see in Section VII.

A necessary condition for unbounded-memory (thus, also for finite-memory) joint observability is the following:

$$P_{\Sigma_o}(K) \cap P_{\Sigma_o}(L - K) = \emptyset \tag{11}$$

Condition (11) requires that K be observable in a centralized manner.

The results of the previous section already provide sufficient conditions for joint observability (see Fig. 1). Another sufficient condition for finite-memory (thus, also unbounded-memory) joint observability is the following:

$$\exists i \,.\, P_i(K) \cap P_i(L - K) = \emptyset \tag{12}$$

Condition (12) essentially states that all observers except the i-th one are redundant. Thus, this is a case where the problem reduces to a centralized observation problem. For this reason, Condition (12) is also sufficient for (finite-memory) local observability, with respect to a decision function B that ignores all inputs except the i-th one, i.e., defined as $B(b_1, ..., b_i, ..., b_n) = b_i$.

Condition (12) is not necessary, as Example 4 shows. Recall that in that example K is finitely locally observable with respect to $\bigwedge$. However, Condition (12) is not verified, because observer 1 cannot distinguish a from ab and observer 2 cannot distinguish c from cd.

Conditions (11) and (12) can be checked algorithmically, since K and L are regular languages. Note that for $n = 1$, Conditions (10-12) all coincide.

The following lemma provides another necessary condition for joint observability.

Lemma 3: If K is observable (resp. finitely observable) w.r.t. L and $(\Sigma_1, ..., \Sigma_n)$ then $P_{\Sigma_o}(K)$ is observable (resp. finitely observable) w.r.t. $P_{\Sigma_o}(L)$ and $(\Sigma_1, ..., \Sigma_n)$.

VI. A SPECIAL CASE: L TRACE LANGUAGE

We now examine the special case where L is a *trace language*. Trace languages are defined with respect to a reflexive and symmetric relation $D \subseteq \Sigma \times \Sigma$, called the *dependence relation*. D induces the irreflexive and symmetric relation $I = (\Sigma \times \Sigma) - D$, called the *independence relation*. D also induces the equivalence $\equiv$, called the trace equivalence, defined as the reflexive, symmetric and transitive closure of the relation $\equiv^1 \subseteq \Sigma^* \times \Sigma^*$, where $\pi \equiv^1 \rho$ iff there exist words $\sigma_1, \sigma_2 \in \Sigma^*$ and $(a, b) \in I$, such that $\pi = \sigma_1 ab \sigma_2$ and $\rho = \sigma_1 ba \sigma_2$. That is, $\pi \equiv \rho$ iff π can be obtained from ρ by repeatedly swapping adjacent independent letters. $L \subseteq \Sigma^*$ is a trace language over (Σ, D) if it is closed under $\equiv$, that is, $\forall \rho, \rho' \in \Sigma^*$, $\rho \equiv \rho' \Rightarrow (\rho \in L \Leftrightarrow \rho' \in L)$.

For our purposes, we define the following dependence relation:

$$D_{\Sigma_1,...,\Sigma_n} = (\bigcup_{i=1,...,n} \Sigma_i \times \Sigma_i) \cup \{(u, u) \mid u \in \Sigma - \Sigma_o\} \tag{13}$$

In words, two distinct letters are dependent iff there is a sub-alphabet Σ_i containing both. When two distinct letters a and b are independent, either they belong to different sub-alphabets Σ_i, or at least one of them is unobservable. In either case, their order of occurrence cannot be reconstructed based on the observations provided by projections onto Σ_i. This is captured in the following lemma.

Lemma 4: Let $\equiv$ be the trace equivalence induced by $D_{\Sigma_1,...,\Sigma_n}$. For any $\rho, \rho' \in \Sigma^*$, $\rho \equiv \rho'$ iff $\forall i \in \{1, ..., n\} \,.\, P_i(\rho) = P_i(\rho')$ and $P_{\Sigma - \Sigma_o}(\rho) \equiv P_{\Sigma - \Sigma_o}(\rho')$.

Using this lemma, we can prove the following.

Theorem 1: Assume L is a trace language over $(\Sigma, D_{\Sigma_1,...,\Sigma_n})$. Then, K is jointly observable w.r.t. L and $(\Sigma_1, ..., \Sigma_n)$ iff K is a trace language over $(\Sigma, D_{\Sigma_1,...,\Sigma_n})$ and Condition (11) holds.

Checking trace-closure of regular languages is decidable.[6] Thus, the above theorem implies that, when L is trace-closed, unbounded-memory joint observability is decidable.

Trace-closure of L does not ensure that JO implies FJO, as the following example shows.

Example 7: Let $\Sigma_1 = \{a_1, b\}$, $\Sigma_2 = \{a_2, b\}$ and $\Sigma = \{a_1, a_2, b\}$. Let $K = ((a_1 + a_2)(a_1 + a_2 + b))^*$. K is jointly observable w.r.t. Σ^* and (Σ_1, Σ_2). Intuitively, it suffices for the decision function to check that between every two consecutive b's (or from the beginning of the computation until the first b observed) the sum of a_1's and a_2's is odd. However, K is not finitely jointly observable. Intuitively, this is because the number of b's can be arbitrary, and observer i needs to record the parity of the number of a_i's it observes between *every* two consecutive b's.

In the example above, $\Sigma_1 \cap \Sigma_2 \neq \emptyset$. It turns out that when Σ_i are pairwise disjoint, and K is regular, JO indeed implies FJO.

Theorem 2: Assume Σ_i are pairwise disjoint, K is regular and L is a trace language over $(\Sigma, D_{\Sigma_1,...,\Sigma_n})$. Then, K is finitely jointly observable w.r.t. L and $(\Sigma_1, ..., \Sigma_n)$ iff K is jointly observable w.r.t. L and $(\Sigma_1, ..., \Sigma_n)$.

The proof of the above result employs Mezei's theorem [8], which states that a regular trace language over $(\Sigma, D_{\Sigma_1,...,\Sigma_n})$ is isomorphic to a finite union of Cartesian products of regular languages over Σ_i.

VII. UNDECIDABILITY

The results of the previous sections already provided some positive answers regarding the decidability of the observation problems we consider. In this section we present some undecidability results.

Theorem 3: Checking unbounded-memory joint observability for regular languages is undecidable for $n \geq 2$.

The above theorem was first proven in [4], using a direct reduction of Post's Correspondence Problem. It turns out

[6] First, observe that L is closed with respect to $\equiv$ iff it is closed with respect to $\equiv^1$. Checking whether L is closed with respect to $\equiv^1$ can be done by building the deterministic, minimal, finite-state automaton recognizing L, call it A_L, and then checking whether A_L has the so-called "diamond property". The latter states that for every state s of A_L, and every pair of independent letters (a, b), the successor of s by ab is the same as the successor of s by ba.

	L trace	L not trace
JO	Equivalent to Condition (11) and K trace. Decidable.	Generally undecidable for $n \geq 2$. When K, L prefix-closed, undecidable for $n \geq 3$. When K, L prefix-closed and $n = 2$?
FJO	When Σ_i pairwise disjoint, equivalent to JO and decidable. Otherwise?	?
LO_B FLO_B	When $B \in \{\wedge, \vee\}$, equivalent and decidable. For general B?	

TABLE I

SUMMARY OF RESULTS AND OPEN QUESTIONS.

that similar undecidability proofs are common in the theory of rational relations [8]. An alternative proof of the above theorem using results from this theory can be found in [6].

The first proof of the above theorem relies on the fact that K and L are not prefix-closed (they are regular, of course). The proof can be modified in the special case where K and L are prefix-closed, to show that the problem is undecidable for $n \geq 3$ [4].

Theorem 4: Checking unbounded-memory joint observability for prefix-closed regular languages is undecidable for $n \geq 3$.

The problem remains open for $n = 2$ in the case where K and L are prefix-closed. Decidability of finite-memory joint observability in the general case also remains open.

The results and open questions are summarized in Table I.

VIII. CONCLUSIONS AND PERSPECTIVES

We have studied some basic problems of decentralized observation in the context of regular languages. A number of interesting open questions remain, summarized in Table I. Apart from those, some other possible directions for further research are the following. First, in the definitions of local observation, a natural extension is to allow the observers to make local decisions that are richer than a single bit. We have already seen that the one-bit restriction can result in non-observability in some cases. Obviously, the local information amount must be fixed, otherwise we fall into the finite-memory joint observation case. Fixing the amount of local information appears to generalize the "conditional" architecture of [11]. It probably also allows for simple decidability results by enumeration, however, it is worth looking for more efficient algorithms (or proving that there is none). Another direction is to consider on-line communication among observers, that is, communication that occurs during the operation of the plant (as opposed to the communication that takes place in joint or local observation, which is done after the plant stops). Such a direction has been partially explored in [2], where it has been shown that, for lossless, order-preserving networks with bounded delays, the problem is decidable.

REFERENCES

[1] S. Tripakis, "Undecidable problems of decentralized observation and control on regular languages," *Information Processing Letters*, vol. 90, no. 1, pp. 21–28, Apr. 2004.

[2] ——, "Decentralized control of discrete event systems with bounded or unbounded delay communication," *IEEE Transactions on Automatic Control*, vol. 49, no. 9, Sept. 2004.

[3] J. Tsitsiklis and M. Athans, "On the complexity of decentralized decision making and detection problems," *IEEE Transactions on Automatic Control*, vol. 30, no. 5, 1985.

[4] S. Tripakis, "Undecidable problems of decentralized observation and control," in *IEEE Conference on Decision and Control (CDC'01)*, 2001.

[5] A. Puri, S. Tripakis, and P. Varaiya, "Problems and examples of decentralized observation and control for discrete event systems," in *Synthesis and Control of Discrete Event Systems*. Kluwer, 2002.

[6] S. Tripakis, "Two-phase distributed observation problems," in *5th Intl. Conf. on Application of Concurrency to System Design (ACSD'05)*. IEEE, 2005.

[7] V. Diekert and G. R. (eds.), *The Book of Traces*. World Scientific, 1995.

[8] J. Berstel, *Transductions and context-free languages*. Teubner, 1979.

[9] M. Mukund, "From global specifications to distributed implementations," in *Synthesis and Control of Discrete Event Systems*. Kluwer, 2002.

[10] R. Debouk, S. Lafortune, and D. Teneketzis, "Coordinated decentralized protocols for failure diagnosis of discrete event systems," *Discrete Event Dynamic Systems*, vol. 10, no. 1-2, pp. 33–86, 2000.

[11] Y. Wang, T. Yoo, and S. Lafortune, "New results on decentralized diagnosis of discrete event systems," Allerton Conference, 2004.

[12] W. Qiu and R. Kumar, "Decentralized failure diagnosis of discrete event systems," in *WODES'04*, 2004.

[13] R. Sengupta and S. Tripakis, "Decentralized fault diagnosis for discrete event systems," in *IEEE Conference on Decision and Control, invited session "Decentralized Approaches for Failure Diagnosis"*, 2002.

[14] R. Cieslak, C. Desclaux, A. Fawaz, and P. Varaiya, "Supervisory control of discrete-event processes with partial observations," *IEEE Transactions on Automatic Control*, vol. 33, pp. 249–260, 1988.