

Decentralized Observability of Discrete Event Systems with Synchronizations^{*}

Alessandro Giua^{a,b}, Cristian Mahulea^c, Carla Seatzu^b

^a*Aix-Marseille University, CNRS, ENSAM, University of Toulon, LSIS UMR 7296, Marseille 13397, France*

^b*Department of Electrical and Electronic Engineering, University of Cagliari, Piazza D'Armi, 09123 Cagliari, Italy*

^c*Aragón Institute of Engineering Research (I3A), University of Zaragoza, Maria de Luna 1, 50018 Zaragoza, Spain*

Abstract

This paper deals with the problem of decentralized observability of discrete event systems. We consider a set of sites each capable of observing a subset of the total event set. When a synchronization occurs, each site transmits its own observation to a coordinator that decides if the word observed belongs to a reference language K or not. Two different properties are studied: uniform q -observability and q -sync observability. It is proved that both properties are decidable for regular languages. Finally, under the assumption that languages K and L are regular, and all the events are observable by at least one site, we propose a procedure to determine the instants at which synchronization should occur to detect the occurrence of any word not in K , as soon as it occurs. The advantage of the proposed approach is that most of the burdensome computations could be moved off-line.

Key words: Discrete Event Systems, Decentralized Observability, Formal Languages

Published as:

A. Giua, C. Mahulea, C. Seatzu, "Decentralized Observability of Discrete Event Systems with Synchronizations," *Automatica*, Vol. 85, pp. 468–477, 2017. DOI: 10.1016/j.automatica.2017.08.009.

^{*} This work has been partially supported by the Ministry of Economy and Competitiveness of Spain and FEDER under Grant DPI2014-57252-R.

The authors want to thank Maria Paola Cabasino, co-author of the conference paper on which is based this manuscript, for all her help during the initial version.

Email addresses: alessandro.giua@univ-amu.fr, giua@diee.unica.it (Alessandro Giua), cmahulea@unizar.es (Cristian Mahulea), seatzu@diee.unica.it (Carla Seatzu).

1 Introduction

Local observability is an important property of discrete event systems defined by Tripakis in [1]. The idea is the following: n local sites observe, through their own projection masks P_i (with $i = 1, \dots, n$), a word w of symbols that is known to belong to a language L . A language $K \subset L$ is locally observable if, assuming all local sites send to a coordinator all observed words $P_i(w)$, the coordinator can decide for any w if the word belongs to K or to $L \setminus K$. Note that this property was shown in [1] to be undecidable even when languages L and K are regular: this is due to the fact that the length of the observed words can be arbitrarily long and the information they contain cannot be compacted in a finite number of states. Moreover, for prefix-closed languages and more than 3 sites the problem is also undecidable. On the contrary, assuming the observed words have bounded length q , one can define the property of *q-observability* that is decidable for arbitrary languages, since it must only be checked over a finite number of words. This property is closely related to local *diagnosability* as defined by Sampath *et al.* [2]. In fact, language K in this setting represents the set of all fault-free evolutions, while the larger set L also includes the faulty ones.

In this paper, which is an extended version of [3], the considered problem is the following. Assume w describes the event driven evolution of a system. The coordinator can at any moment send a request to all local sites to get the locally observed words since the previous request: such a mechanism is called *synchronization*. After each synchronization (which in general is costly) a coordinator should be able to decide if, on the basis of the information received so far from the local sites, the word w generated belongs to the reference language K . We assume that the maximal number of events that can be generated by the system between two consecutive synchronizations is bounded. The coordinator should request as few synchronizations as needed to solve the observability problem. Also the distance between two consecutive synchronizations, expressed in terms of the number of events generated between them, may opportunistically vary with the word generated so far.

In this setting, although the basic notion of local observability given by Tripakis is still fundamental, two major extensions are needed. In fact the observability property defined in [1] makes two rather restrictive assumptions.

- The first assumption is that the observability property is defined only with respect to words in L . On the contrary, in our setting synchronizations occur repeatedly. Thus if a synchronization occurs after a word w has been generated we are interested in the observability of the residual language $w^{-1}K$, i.e., the set of words that can be generated after w , with respect to the residual language $w^{-1}L$. Correspondingly, we introduce the notion of *uniform observability*.
- The second assumption in [1] is that when the observation starts, the word generated so far (that as discussed in the previous paragraph is always the empty word) is perfectly known. On the contrary, in our setting when a synchronization occurs the coordinator should be able to determine if the generated word belongs to the reference language K or not, but may not be able to unambiguously estimate it. Thus when next observation starts the word generated so far is only known to belong to a given set.

Combining the two extensions above, we introduce the notion of *q-sync observability*.

We point out a limitation of our approach: we assume that the coordinator at any time instant knows how many events have occurred so far, although it cannot directly observe *which* events have occurred. This assumption does not fit in a general asynchronous setting, where events may occur at arbitrary time instants. On the contrary, it makes sense in a synchronous setting where events occur with a fixed timing. Furthermore, we point out that our results can also be applied in those asynchronous cases in which any two consecutive events are spaced by a fixed known time interval. In such a case the coordinator knows an upper bound on the number of events that have occurred since last synchronization and can use this bound to determine when next synchronization should occur.

Literature review. Observability is a fundamental property that has received a lot of attention during the last decades. Several contributions have been presented in the framework of automata since late eighties and nineties [4–7]. Caines *et al.* [4] showed how it is possible to use the information contained in the past sequence of observations (given as a sequence of observation states and control inputs) to compute the set of consistent states, while in [5] the observer output is used to steer the state of the plant to a desired terminal state. A similar approach was used by Kumar *et al.* [7] when defining observer based dynamic controllers in the framework of supervisory predicate control problems.

Özveren and Willsky [6] proposed an approach for building observers that allows one to reconstruct the state of finite automata after a word of bounded length has been observed, showing that an observer may have an exponential number of states.

A very general approach for observability with communication has been presented by Barret and Lafortune in [8] in the context of supervisory control, and several techniques for designing a possibly optimal communication policy have also been discussed therein. By optimal we mean that the local sites communicate as late as possible, only when strictly necessary to prevent the undesirable behavior. Our work is by large a special case of the architecture in [8] because we allow communications only between the coordinator and the local observers — and not among local observers — and we do not consider a control problem but simply an observation one. There are, however, a few differences in our approach with respect to [8] that motivate the need for additional investigation. First, we frame our results in the context of languages, rather than automata: this means that some of our definitions and results apply to possibly non regular languages. Secondly, while in [8] communications are decided by the local observers and are triggered by the observation of an event, in our case the communications are triggered by the coordinator.

Preliminary results of this paper have been presented in [3]. The actual paper has been substantially improved by adding new theoretical results and new examples in order to clarify the theoretical results while the structure has been changed in order to improve the readability. Moreover, we are introducing a new notion called *uniform observability* that permits us to establish new connections between our work and the work of Tripakis[1].

Other interesting contributions related to the problem considered in this paper have been recently published. Fabre and Benveniste in [9] consider a distributed/modular system with several modules, each associated with a local observer/supervisor that only has access to the local observations and the model of the local module. Giua and Seatzu in [10] propose a procedure that produces an estimation of the state, while the special structure of Petri nets allows one to determine, using linear algebraic tools, if a given marking is consistent with the observed behavior without the explicit enumeration of the (possibly infinite) consistent set. Petri Nets with unobservable transitions, i.e., transitions labeled with the empty word, were studied in [11]. Here the notion of basis marking has been introduced. The idea is that under very general conditions, namely the acyclicity of the unobservable subnet, it is possible to characterize the set of markings consistent with an observation in terms of sequences of minimal length. The markings reached by these sequences are called basis markings and all other markings consistent with the observation can be obtained from the knowledge of this smaller set. Li and Hadjicostis in [12] consider the problem of state estimation in a Petri net framework assuming multiple observation sites with a partial order model of time. Finally, Hadjicostis and Seatzu in [13] focus on the problem of decentralized state estimation where two or more observation sites send information to a coordinator who aims to determine the set of possible current states of a given discrete event system modeled as a nondeterministic finite automaton.

Finally the approaches we present in this paper may also be useful to address other related problems in the area of discrete event systems, including (decentralized) diagnosis [14,15], prognosis [16,17], and recovery, distributed supervisory control [18] and minimal sensor activation for communicating observers [19]. Summarizing, the proposed results may be useful in all the applications where the state observation is done in a decentralized way, but it is important to minimize the cost and the energy consumption resulting from synchronization. A typical example in this context are sensor networks. Analogously, it may be important to minimize synchronizations in any application where security and privacy requirements are pressing, and when intrusions may suddenly occur.

Structure of the paper. The paper is structured as follows. In Section 2 we introduce basic notations on finite state automata and formal languages. In Section 3 we provide some language observability definitions and properties and discuss relationships among them. Section 4 focusses on uniform q -observability and provides specific results in the case of regular languages. A new property called q -sync observability is introduced and studied in Section 5. Again, special results are proved in the case of regular languages. The problem of determining the instants at which synchronize the observations from the different sites, so that a word not belonging to the reference language is identified as soon as occurred, is studied in Section 6. Conclusions are finally drawn in Section 7 where our future lines of research in this framework are pointed out.

2 Basic notations

Let Σ be a finite *alphabet*: Σ^* denotes the set of all finite words or words over Σ , i.e., its *Kleene star*, including the *empty word* ε . A language on Σ is a set of words $L \subseteq \Sigma^*$.

The *concatenation* of two words u and v is the word $w = uv$: in this case u is called a prefix of w . The set $Pref(L)$ contains all prefixes of words in L .

Given a word $w \in \Sigma^*$, and an alphabet $\Sigma_i \subseteq \Sigma$, we denote as $P_i(w)$ the *projection of w over Σ_i* , that can be recursively defined as follows:

$$P_i(w) = \begin{cases} \varepsilon & \text{if } w = \varepsilon, \\ P_i(u) & \text{if } w = ue, e \notin \Sigma_i, \\ P_i(u)e & \text{if } w = ue, e \in \Sigma_i. \end{cases}$$

Given a word $t \in \Sigma_i^*$, where $\Sigma_i \subseteq \Sigma$, we denote as $P_i^{-1}(t)$ the *inverse projection of t* as: $P_i^{-1}(t) = \{w \in \Sigma^* \mid P_i(w) = t\}$. Projections and their inverses are extended to languages by applying them to all the words in the language.

A *deterministic finite automaton* (DFA) is a 5-tuple $G = (X, \Sigma, \delta, x_0, X_m)$ where X is the finite set of states, Σ is the finite set of events, a partial function $\delta : X \times \Sigma \rightarrow X$ is the transition function, $x_0 \in X$ is the initial state, and $X_m \subseteq X$ is the set of *marked states*. The languages generated and accepted by G , denoted by $L(G)$ and $L_m(G)$, respectively, are defined as $L(G) = \{w \in \Sigma^* \mid \delta^*(x_0, w) \text{ is defined}\}$ and $L_m(G) = \{w \in \Sigma^* \mid \delta^*(x_0, w) \in X_m\}$, where $\delta^* : X \times \Sigma^* \rightarrow X$ is the reflexive and transitive closure of the transition function.

Given a language L and a word $w \in \Sigma^*$, the *residual of L with respect to (wrt) w* is the language $w^{-1}L = \{z \mid wz \in L\}$. The language L is *regular* iff the set of its residuals as w ranges over Σ^* is finite, i.e., iff the set $\{w^{-1}L \mid w \in \Sigma^*\}$ is finite. The cardinality of the set $\{w^{-1}L \mid w \in \Sigma^*\}$ is called the *index of L* and, in the case of a regular language L is equal to the number of states of the minimal automaton accepting L . In the paper we often represent a regular language by the *regular expression* that describes it.

3 Language observability

Let us consider two languages K and L defined over an alphabet Σ , such that $K \subset L \subseteq \Sigma^*$, and a set of n sub-alphabets $\Sigma_i \subseteq \Sigma$, $i = 1, \dots, n$. The n sub-alphabets Σ_i 's are associated with n sites \mathcal{S}_i , $i = 1, \dots, n$. In particular, Σ_i includes all the events that can be *observed* by \mathcal{S}_i .

The following definition of decentralized observability has been given by Tripakis in [1] for regular languages. We consider here an extension to general languages.

Definition 1 *Let us consider two languages L and $K \subset L$. The language K is jointly observable wrt L and $\{\Sigma_i \mid i = 1, \dots, n\}$, if there exists a function $f : \Sigma_1^* \times \dots \times \Sigma_n^* \rightarrow \{0, 1\}$, such that $\forall w \in L$*

$$w \in K \Leftrightarrow f(P_1(w), \dots, P_n(w)) = 1. \quad (1)$$

Note that if $\varepsilon \in K$ and K is jointly observable wrt L and $\{\Sigma_i \mid i = 1, \dots, n\}$, then there exists no word $w \in L \setminus K$ whose projection over all Σ_i 's is equal to the empty word ε .

It is proved in [1] that even for regular languages, checking the above property requires unbounded memory, and hence it is undecidable because the word w may have arbitrary length. In this paper this definition will be revised in three ways:

- by assuming bounded observation: this leads to the definition of *q-observability* that assumes that the length of the word w in Def. 1 is upper bounded by a finite number q (see Def. 2);
- by assuming that before the observation, a word in K has been generated; this leads to the definition of *uniform observability* (see Def. 3);
- by assuming both bounded observations and repeated synchronizations; this leads to the definition of *uniform q-observability* (see Def. 6).

Considering words of finite length q , Def. 1 can be rewritten as follows.

Definition 2 *Let us consider two languages L and $K \subset L$. The language K is q-observable wrt L and $\{\Sigma_i \mid i = 1, \dots, n\}$, if there exists a function $f : \Sigma_1^* \times \dots \times \Sigma_n^* \rightarrow \{0, 1\}$, such that $\forall w \in L$ with $|w| \leq q$*

$$w \in K \Leftrightarrow f(P_1(w), \dots, P_n(w)) = 1. \quad (2)$$

It is immediate to observe that, given two languages K and L , and an arbitrary value of $q \in \mathbb{N}$, if K is jointly observable wrt L and $\{\Sigma_i \mid i = 1, \dots, n\}$ (Def. 1), then K is also q -observable wrt L and $\{\Sigma_i \mid i = 1, \dots, n\}$ (Def. 2). Obviously, the other implication is not true in general, namely, K could be q -observable for some given q but not jointly observable.

Let us modify Def. 1 to keep into account the possibility that the observation starts after an arbitrary word in K has occurred.

Definition 3 *Let us consider two languages L and $K \subset L$. The language K is uniformly observable wrt L and $\{\Sigma_i \mid i = 1, \dots, n\}$, if $\forall u \in K$ there exists a function $f_u : \Sigma_1^* \times \dots \times \Sigma_n^* \rightarrow \{0, 1\}$, such that $\forall w \in u^{-1}L$ it holds*

$$w \in u^{-1}K \Leftrightarrow f_u(P_1(w), \dots, P_n(w)) = 1. \quad (3)$$

In simple words, uniform observability implies the possibility of establishing if the behavior of a given system is in the reference language K after the occurrence of a word $uw \in L$, knowing its prefix $u \in K$. The following proposition shows that Def. 3 is equivalent to Def. 1 if $\varepsilon \in K$.

Proposition 4 *Let Σ be a finite alphabet, and $\Sigma_i \subseteq \Sigma$, with $i = 1, \dots, n$, be n sub-alphabets of Σ . Let L and K be two languages such that $K \subset L \subseteq \Sigma^*$. If language K is jointly observable wrt L and $\{\Sigma_i \mid i = 1, \dots, n\}$, then K is uniformly observable wrt L and $\{\Sigma_i \mid i = 1, \dots, n\}$. If $\varepsilon \in K$ the converse also holds.*

PROOF. (\implies) *Assume that K is jointly observable wrt L and $\{\Sigma_i \mid i = 1, \dots, n\}$. In order to show that it is also uniformly observable, let consider a word $u \in K$ preceding the observation w . The function f_u in Def. 3 can be defined as follows*

$$f_u(P_1(w), \dots, P_n(w)) = f(P_1(uw), \dots, P_n(uw))$$

where $f(P_1(uw), \dots, P_n(uw))$ is the function in Def. 1 which exists by assumption.

(\impliedby) *By simply taking $u = \varepsilon$ in Def. 3, Def. 1 holds.* □

The following example shows that if $\varepsilon \notin K$, then uniform observability does not imply joint observability.

Example 5 *Let us consider two languages $K = \{ac^i \mid i \geq 0\}$ and $L = K \cup \{b\}$ and one site $\Sigma_1 = \{c\}$. Notice that $\varepsilon \notin K$.*

Language K is not jointly observable wrt L and Σ_1 because there exist two words $w_1 = a \in K$ and $w_2 = b \in L \setminus K$ having the same projections $P_1(w_1) = P_1(w_2) = \varepsilon$. However, K is uniformly observable wrt L and Σ_1 . Indeed, since $u \in K$, then $u = ac^j, j \geq 0$ and $u^{-1}K = \{c^i \mid i \geq 0\}$. Therefore, $f_u(P_1(w)) = 1$ for all $w \in u^{-1}K$. ■

An additional property we consider takes into account both bounded observations and uniform observability.

Definition 6 *Let Σ be a finite alphabet, and $\Sigma_i \subseteq \Sigma$, with $i = 1, \dots, n$, be n sub-alphabets of Σ . Let L and K be two languages such that $K \subset L \subseteq \Sigma^*$.*

Language K is uniformly q -observable wrt L and $\{\Sigma_i \mid i = 1, \dots, n\}$, if $\forall u \in K$ there exists a function $f_u : \Sigma_1^ \times \dots \times \Sigma_n^* \rightarrow \{0, 1\}$ such that $\forall w \in u^{-1}L$ with $|w| \leq q$, it holds*

$$w \in u^{-1}K \iff f_u(P_1(w), \dots, P_n(w)) = 1. \quad (4)$$

In simple words, uniform q -observability implies the possibility of establishing if the behavior of a given system is in the reference language K after the occurrence of a word w of length less or equal to q , knowing the word u in K preceding w .

The following proposition clarifies the relationship between uniform q -observability and q -observability.

(joint) observability (Tripakis [1])	\implies (Prop. 4) \longleftarrow if $\varepsilon \in K$ (Prop. 4)	uniform observability
\Downarrow (trivial)		\Downarrow (trivial)
q -observability	\longleftarrow if $\varepsilon \in K$ (Prop. 7)	uniform q -observability

Table 1
Relationships among different observability notions.

Proposition 7 Let Σ be a finite alphabet, and $\Sigma_i \subseteq \Sigma$, with $i = 1, \dots, n$, be n sub-alphabets of Σ . Let L and K be two languages such that $K \subset L \subseteq \Sigma^*$ and $\varepsilon \in K$. Given a number $q \in \mathbb{N}$, if language K is uniformly q -observable wrt L and $\{\Sigma_i \mid i = 1, \dots, n\}$, then K is also q -observable wrt L and $\{\Sigma_i \mid i = 1, \dots, n\}$.

PROOF. By taking $u = \varepsilon$ in Def. 6, Def. 2 holds. □

The following example shows that q -observability does not imply uniform q -observability.

Example 8 Let $\Sigma_1 = \{a, c\}$ and $\Sigma_2 = \{b\}$ be two alphabets and let $K = \overline{(a b c)^*}$ and $L = \overline{(a b c)^*} + \overline{(a c b)^*}$, where overline denotes prefix closure. Obviously, $K \subset L$. Language K is not uniformly 2-observable. Indeed, let us consider $w = a \in K$ in Def. 6. Let $u_1 = bc \in a^{-1}K$ and $u_2 = cb \in a^{-1}L$. It can be checked immediately that u_1 and u_2 have the same projection on both alphabets. However, $au_1 = abc \in K$ while $au_2 = acb \in L \setminus K$, thus function f_a cannot be defined. ■

Table 1 summarizes the relationships among the observability properties defined above.

4 Uniform q -observability

4.1 General results

The following proposition shows that uniform q -observability implies uniform $(q - 1)$ -observability.

Proposition 9 If K is uniformly q -observable wrt L and $\{\Sigma_i \mid i = 1, \dots, n\}$, then it is also uniformly $(q - 1)$ -observable wrt them.

PROOF. Follows from the fact that the same f_u function used in the case of uniform q -observability can be used in the case of uniform $(q - 1)$ -observability, simply restricting its arguments to words of length $q - 1$. □

This implies that, if a language is uniformly q -observable for some finite $q > 1$, then it is also uniformly 1-observable. The following proposition shows that any language is uniformly 1-observable if any event can be observed by at least one site.

Proposition 10 Let us consider a set of alphabets Σ_i , $i = 1, \dots, n$, such that $\Sigma_1 \cup \dots \cup \Sigma_n = \Sigma$. Any language $K \subset L \subseteq \Sigma^*$ is uniformly 1-observable wrt to L and $\{\Sigma_i \mid i = 1, \dots, n\}$.

PROOF. Since $\Sigma_1 \cup \dots \cup \Sigma_n = \Sigma$, there exists at least one site that can detect any event e that has occurred, i.e., for each $e \in \Sigma$, there exists i such that $P_i(e) = e$. Let $u \in K$, then the function f_u of Def. 6 is defined as follows:

$$f_u(P_1(e), \dots, P_n(e)) = \begin{cases} 1, & \text{if } ue \in K \\ 0, & \text{otherwise} \end{cases}$$

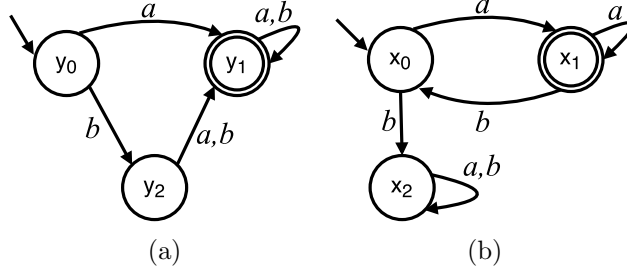


Fig. 1. The DFA used in Example 14.

Being possible to define the function for any observed event, the system is uniformly 1-observable. \square

On the contrary, uniform 1-observability is no more ensured if one or more events in Σ are not observable by all the sites.

4.2 Results for regular languages

If L and K are regular languages, uniform q -observability is decidable for any finite q .

Proposition 11 *Let us consider a set of alphabets Σ_i , $i = 1, \dots, n$, such that $\Sigma_1 \cup \dots \cup \Sigma_n = \Sigma$. Let K and L be two regular languages such that $K \subset L \subseteq \Sigma^*$. Uniform q -observability of K wrt L and $\{\Sigma_i \mid i = 1, \dots, n\}$ is decidable for any finite $q \in \mathbb{N}$.*

PROOF. According to Myhill-Nerode Theorem [20], a regular language K has a finite index, i.e., the set of languages $\{u^{-1}K \mid u \in K\}$ is finite. This implies that it is sufficient to check for a finite number of words u the existence of a function f_u defined over a finite subset of $\Sigma_1^* \times \dots \times \Sigma_n^*$, i.e., the set of projections on Σ_i 's, $i = 1, \dots, n$, of words $w \in u^{-1}L$ of length less than or equal to q . Thus the problem is decidable. \square

Let us now introduce an equivalence relationship among words that allows us to rephrase the definition of uniform observability. This will be used later in Algorithm 1 to perform analysis of q -observability.

Definition 12 *Let Σ be a finite alphabet, and $\Sigma_i \subseteq \Sigma$, with $i = 1, \dots, n$, be n sub-alphabets of Σ . A word $w \in \Sigma^*$ is observation equivalent (or simply equivalent) to $v \in \Sigma^*$, denoted as $w \equiv v$, if $P_i(w) = P_i(v)$ for all $i = 1, \dots, n$. We say that two words that are not equivalent are distinguishable.*

Note that the relation introduced in Def 12 is an equivalence relation which determines a unique minimal partition of a set of words into equivalence classes. We denote $[w]$ the set of words which are observation equivalent to word w . Based on the above definition, an immediate condition to check uniform q -observability is given by the following proposition.

Proposition 13 *Let Σ be a finite alphabet, and $\Sigma_i \subseteq \Sigma$, for $i = 1, \dots, n$, be n sub-alphabets of Σ . Let L and K be two regular languages such that $K \subset L \subseteq \Sigma^*$. Language K is uniformly q -observable wrt L and $\{\Sigma_i \mid i = 1, \dots, n\}$, if $\forall u \in K$, and $\forall w \in u^{-1}L$ with $|w| \leq q$, $[w] \cap u^{-1}K \neq \emptyset \Rightarrow [w] \subseteq u^{-1}K$.*

PROOF. We prove this by contradiction. Let us assume that $[w] \cap u^{-1}K \neq \emptyset$ but $[w] \not\subseteq u^{-1}K$. By Def. 6, there must exist words $v_1, v_2 \in [w]$ such that: $v_1 \in u^{-1}K$ and thus $f_u(P_1(v_1), \dots, P_n(v_1)) = 1$ while $v_2 \notin u^{-1}K$ and thus $f_u(P_1(v_2), \dots, P_n(v_2)) = 0$.

However, this leads to a contradiction because being $v_1, v_2 \in [w]$, then $v_1 \equiv v_2$, i.e., $P_i(v_1) = P_i(v_2)$ for all $i = 1, \dots, n$. \square

Example 14 Given alphabets $\Sigma = \{a, b\}$, $\Sigma_1 = \{a\}$, $\Sigma_2 = \{b\}$, let $K = a(a+ba)^*$ and $L = a(a+b)^* + b(a+b)(a+b)^*$. Language L is accepted by the DFA in Fig. 1(a), while K is the language accepted by DFA in Fig. 1(b).

First, we observe that K is 2-observable wrt L , Σ_1 and Σ_2 . In fact, the set of words of length less than or equal to 2 in L is $\{a, aa, ab, ba, bb\}$ and this set can be partitioned as $\{a, aa\} \subseteq K$ and $\{ab, ba, bb\} \subseteq L \setminus K$. The function $f(P_1(w), P_2(w))$ that takes value 1 if $P_2(w) = \varepsilon$ and value 0 otherwise recognizes words in K .

However, K is not uniformly 2-observable. To show this consider the word $u = a \in K$. The set of words of length less than or equal to 2 in $a^{-1}L$ is $\{\varepsilon, a, b, aa, ab, ba, bb\}$ and this set can be partitioned as $\{\varepsilon, a, aa, ba\} \subseteq a^{-1}K$ and $\{b, ab, bb\} \subseteq a^{-1}L \setminus a^{-1}K$. Obviously, $[ab] = \{ab, ba\}$ since $(P_1(ab), P_2(ab)) = (P_1(ba), P_2(ba)) = (a, b)$, however $[ab] \cap a^{-1}K = \{ba\} \neq \emptyset$ while $[ab] \not\subseteq a^{-1}K$. By Proposition 13 we conclude that K is not uniformly 2-observable wrt L , Σ_1 and Σ_2 .

Incidentally, we also note that by Proposition 10 it follows that K is uniformly 1-observable wrt L , Σ_1 and Σ_2 ■

From the Myhill-Nerode Theorem [20], it follows that with each regular language can be uniquely associated a minimal DFA accepting it. Now, let L and K be two regular languages, where K represents the *reference* behavior and L represents a larger behavior. We define G_L as the minimal DFA accepting $L_m(G_L) = L$ and generating $L(G_L) = Pref(L)$. We also define G_K as the minimal complete DFA accepting $L_m(G_K) = K$; since this automaton is complete it generates $L(G_K) = \Sigma^*$. Starting from G_L and G_K , we construct a DFA H whose set of final states is partitioned in two subsets: the set of states accepting words in K and the set of states accepting words in $L \setminus K$.

Definition 15 Let $G_L = (X_1, \Sigma, \delta_1, x_{0,1}, X_{m,1})$ with $L_m(G_L) = L$ and $L(G_L) = Pref(L)$, and $G_K = (X_2, \Sigma, \delta_2, x_{0,2}, X_{m,2})$ with $L_m(G_K) = K$ and $L(G_K) = \Sigma^*$. Let H be the DFA defined as $H = (X, \Sigma, \delta, x_0, X_m \cup \hat{X}_m)$ where

- $X = X_1 \times X_2$,
- $\delta((x_1, x_2), e) = (\delta_1(x_1, e), \delta_2(x_2, e))$, for all $(x_1, x_2) \in X_1 \times X_2$ and for all $e \in \Sigma$ such that $\delta_1(x_1, e)$ is defined,
- $x_0 = x_{0,1} \times x_{0,2}$,
- $X_m = X_{m,1} \times X_{m,2}$,
- $\hat{X}_m = X_{m,1} \times (X \setminus X_{m,2})$.

The automaton H is the *parallel composition* of G_L and G_K with a set of final states partitioned into two subsets X_m and \hat{X}_m .

One can readily verify that $L(H) = Pref(L)$ and the following implications hold for H :

$$w \in K \iff \delta^*(x_0, w) \in X_m, \quad (5)$$

and

$$w \in L \setminus K \iff \delta^*(x_0, w) \in \hat{X}_m. \quad (6)$$

Algorithm 1 can be used to check q -observability. Its correctness is proved by the following proposition which provides an explanation of the algorithm.

Proposition 16 Let Σ be a finite alphabet, and $\Sigma_i \subseteq \Sigma$, with $i = 1, \dots, n$, be n sub-alphabets of Σ . Let L and K be two regular languages such that $K \subset L \subseteq \Sigma^*$. The language K is uniformly q -observable with respect to L and $\{\Sigma_i \mid i = 1, \dots, n\}$ if Algorithm 1 returns true.

PROOF. Notice that the set of words $\delta^*(x, u)$ that can be generated by H starting from a state $x \in X_m \cup \hat{X}_m$ defines the residual $u^{-1}L$. Therefore, according to Def. 6, in order to check uniform q -observability we have to consider all states in X_m as initial ones (Steps 2 and 3). For each initial state $x \in X_m$, all words of length less than or equal to q that can be generated by H starting from x and reaching a state in $X_m \cup \hat{X}_m$ (namely, words in L), are computed (Step 5) and partitioned into equivalence classes (Step 6). If all words in all equivalence classes are either in $u^{-1}K$ or outside $u^{-1}K$ (where $u \in K$ is such that $\delta^*(x_0, u) = x$), then the algorithm iterates considering a new initial state. Otherwise, the language K is not uniformly q -observable according to Proposition 13. □

The complexity of Algorithm 1 is $O(|X| \cdot |\Sigma|^q)$.

Algorithm 1 Analysis of uniform q -observability

Input: DFA $H = (X, \Sigma, \delta, x_0, X_m \cup \hat{X}_m)$ built according to Def. 15 from languages L and K ; a positive integer q ; a set of alphabets $\{\Sigma_i \mid i = 1, \dots, n\}$

Output: A boolean variable UQO specifying if K is uniformly q -observable wrt L and $\{\Sigma_i \mid i = 1, \dots, n\}$

- 1: Let $UQO = true$.
 - 2: Let $\mathcal{X} = X_m$ be the set of final states of H accepting words in K .
 - 3: **while** $UQO = true$ **and** $\mathcal{X} \neq \emptyset$ **do**
 - 4: Choose a state $x \in \mathcal{X}$.
 - 5: Compute the set of words of length less than or equal to q that can be generated by H starting from x and reaching a state in $X_m \cup \hat{X}_m$.
 - 6: Partition this set into equivalence classes $\Pi = \{\mathcal{W}_1, \dots, \mathcal{W}_r\}$ according to Def. 12.
 - 7: **if** \exists an equivalence class $\mathcal{W} \in \Pi$ such that $v_1, v_2 \in \mathcal{W}$ with $\delta^*(x, v_1) \in X_m$ and $\delta^*(x, v_2) \in \hat{X}_m$ **then**
 - 8: let $UQO = false$
 - 9: **end if**
 - 10: Let $\mathcal{X} = \mathcal{X} \setminus \{x\}$;
 - 11: **end while**
-

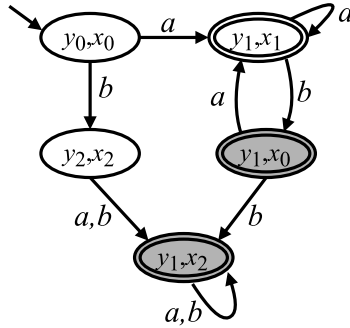


Fig. 2. The DFA H in Example 17.

Example 17 Let us consider again the case of Example 14 where L is the language accepted by the DFA in Fig. 1(a), while K is the language accepted by the DFA in Fig. 1(b). The DFA H built according to Def. 15 is shown in Fig. 2 where $X_m = \{(y_1, x_1)\}$ and $\hat{X}_m = \{(y_1, x_0), (y_1, x_2)\}$. Note that in this figure, and in the following ones, final states in \hat{X}_m are denoted by a gray shade.

Let us assume that $\Sigma_1 = \{a\}$ and $\Sigma_2 = \{b\}$.

We want to study uniform 2-observability using Algorithm 1. In Step 2, \mathcal{X} is initialized at X_m . In Step 4 we select $x = (y_1, x_1)$. The set of words of length less than or equal to 2 that can be generated by H starting from x and reaching a state in $X_m \cup \hat{X}_m$ are: ε , a , b , aa , ab , ba and bb (Step 5). In Step 6 we partition this set into equivalence classes: $\mathcal{W}_1 = \{\varepsilon\}$, $\mathcal{W}_2 = \{a\}$, $\mathcal{W}_3 = \{b\}$, $\mathcal{W}_4 = \{aa\}$, $\mathcal{W}_5 = \{ab, ba\}$, and $\mathcal{W}_6 = \{bb\}$. Obviously, the condition in Step 7 is satisfied for \mathcal{W}_5 : $\delta^*(x, ba) \in X_m$ and $\delta^*(x, ab) \in \hat{X}_m$. Therefore, the algorithm concludes (Step 8) that the language K is not uniformly 2-observable. \blacksquare

5 q -sync observability

In this section we introduce a new property, strictly related to uniform q -observability, that we call q -sync observability. Subsection 5.1 provides general results while Subsection 5.2 considers regular languages.

5.1 General results

The main difference of q -sync observability with respect to joint observability is that, although we are still considering words of arbitrary length, we assume that while they are generated there exist repeated communications from the local sites to the coordinator (i.e., synchronizations). More precisely, at most q events can occur between any two consecutive synchronizations.

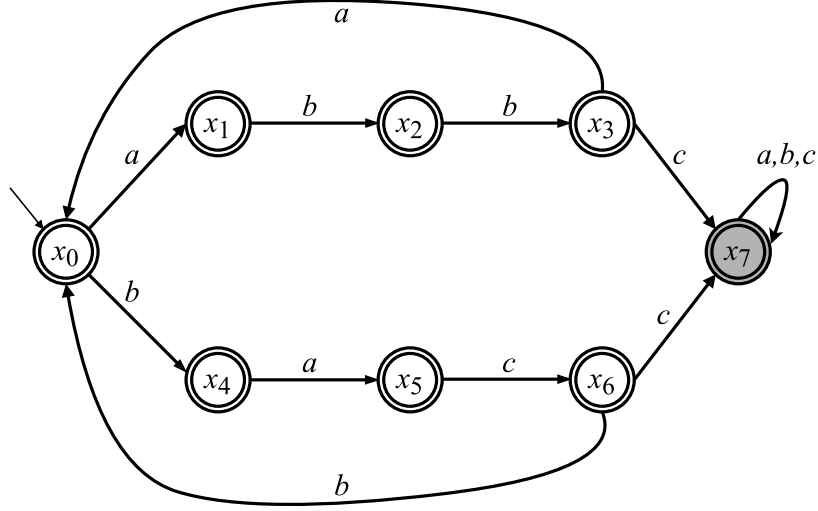


Fig. 3. The DFA in Examples 22 and 25.

Definition 18 Let Σ be a finite alphabet, and $\Sigma_i \subseteq \Sigma$, with $i = 1, \dots, n$, be n sub-alphabets of Σ . Let L and K be two languages such that $K \subseteq L \subseteq \Sigma^*$. The language K is called q -sync observable wrt L and $\{\Sigma_i \mid i = 1, \dots, n\}$, if there exists a function $f : (\Sigma_1^* \times \Sigma_2^* \times \dots \times \Sigma_n^*)^* \rightarrow \{0, 1\}$ such that for all $m \in \mathbb{N}$ and for all sequences of m words (w_1, w_2, \dots, w_m) such that $w_i \in L$ and $|w_i| \leq q$, $\forall i = 1, \dots, m$, it holds

$$w_1 w_2 \dots w_m \in K \iff f(P_1(w_1), \dots, P_n(w_1), \dots, P_1(w_m), \dots, P_n(w_m)) = 1. \quad (7)$$

The notion of observable equivalence can be easily extended to the case of sync observable equivalence.

Definition 19 Let Σ be a finite alphabet, and $\Sigma_i \subseteq \Sigma$, with $i = 1, \dots, n$, be n sub-alphabets of Σ . Consider two sequences w_1, w_2, \dots, w_m and v_1, v_2, \dots, v_m , where $(w_1 w_2 \dots w_m) \in L$ and $(v_1 v_2 \dots v_m) \in L$. The two sequences are sync observable equivalent, or simply equivalent when clear from the context, if $P_i(w_j) = P_i(v_j)$ for all $i = 1, \dots, n$, and all $j = 1, \dots, m$. We denote this $(w_1, w_2, \dots, w_m) \equiv (v_1, v_2, \dots, v_m)$. Finally, we say that two sequences that are not equivalent are distinguishable.

Proposition 20 If K is q -sync observable wrt L and $\{\Sigma_i \mid i = 1, \dots, n\}$, then it is also $(q - 1)$ -sync observable wrt them.

PROOF. Follows by the fact that the same f function used in the case of q -sync observability can be used in the case of $(q - 1)$ -sync observability, simply restricting its arguments to words of length $q - 1$. \square

Proposition 21 If a language K is q -sync observable wrt to a language L and a set of alphabets $\{\Sigma_i \mid i = 1, \dots, n\}$, then it is also uniformly q -observable wrt L and $\{\Sigma_i \mid i = 1, \dots, n\}$.

PROOF. It is a consequence of Def. 6 and 18. Indeed, consider any word $u \in K$ and write it as $u = w_1 w_2 \dots w_k$ where $|w_i| \leq q$ for all i . Then for any word $w \in u^{-1}L$ with $|w| \leq q$, function f_u can be defined as in Def. 6 in terms of function f in Def. 18 as follows:

$$\begin{aligned} f_u(P_1(w), \dots, P_n(w)) &= \\ &= f(P_1(w_1), \dots, P_n(w_1), \dots, P_1(w_k), \dots, P_n(w_k), \dots \\ &\dots P_1(w), \dots, P_n(w)) \end{aligned}$$

showing that K is uniformly q -observable wrt L and $\{\Sigma_i \mid i = 1, \dots, n\}$. □

On the contrary, uniform q -observability does not imply q -sync observability as shown by the following example.

Example 22 Let L be the language accepted by the DFA in Fig. 3, while K is the language accepted by the same DFA but neglecting x_7 . Finally, assume three sites with alphabets $\Sigma_1 = \{a\}$, $\Sigma_2 = \{b\}$ and $\Sigma_3 = \{c\}$, respectively. Using the approach discussed in the previous section one may prove that K is uniformly 3-observable.

However, it is not 3-sync observable. To show this, let us consider $w_1 = ab$, $w_2 = bc$, $v_1 = ba$ and $v_2 = cb$. One can verify that $v_1v_2 \in K$ and we should define:

$$\begin{aligned} f(P_1(v_1), P_2(v_1), P_3(v_1), P_1(v_2), P_2(v_2), P_3(v_2)) = \\ f(a, b, \varepsilon, \varepsilon, b, c) = 1. \end{aligned}$$

However, it holds $w_1w_2 \in L \setminus K$, therefore we should also define

$$\begin{aligned} f(P_1(w_1), P_2(w_1), P_3(w_1), P_1(w_2), P_2(w_2), P_3(w_2)) = \\ f(a, b, \varepsilon, \varepsilon, b, c) = 0. \end{aligned}$$

clearly a contradiction. We conclude that a function f as in Def. 18 does not exist. ■

5.2 Results for regular languages

Interesting results can be proved if K and L are regular languages.

Proposition 23 Let us consider a set of alphabets Σ_i , $i = 1, \dots, n$, such that $\Sigma_1 \cup \dots \cup \Sigma_n = \Sigma$. Let K and L be two regular languages such that $K \subset L \subseteq \Sigma^*$. For any finite $q \in \mathbb{N}$, q -sync observability of K wrt L and $\{\Sigma_i \mid i = 1, \dots, n\}$ is decidable.

PROOF. Since we are taking into account regular languages we can equivalently speak in terms of the DFA $H = (X, \Sigma, \delta, x_0, X_m \cup \hat{X}_m)$ built according to Def. 15. To determine if the property holds for $m = 1$ (first synchronization) we need to check all words w_1 of length less than or equal to q that can be generated by H starting from the initial state x_0 . Among them, we verify that no word which reaches X_m (a word in K) is equivalent to a word which reaches \hat{X}_m (a word in $L \setminus K$).

Consider the case $m = 2$. After the first synchronization is performed when the generated word is w_1 , we do not know the current state of H but we know that it belongs to a given set $S(w_1) \subseteq X$ and the set $\Xi_1 = \{S(w_1) \subseteq X \mid w_1 \in \text{Pref}(L), |w_1| \leq q\}$ is finite. Now, for all possible $S \in \Xi_1$ we consider the language $L(H \mid S) = \cup_{x \in S} L(H \mid x)$ where $L(H \mid x)$ denotes the language generated by the automaton with initial state x and we need to check all words of length less than or equal to q in this language. Again, among them we must verify that no word which reaches X_m starting from a state in S is equivalent to a word which reaches \hat{X}_m starting from a state in S (possibly different from the previous one). As m is increased, one may have different sets Ξ_k , $k = 1, \dots, m - 1$, to check but eventually $\Xi_{k+1} = \cup_{j=1}^k \Xi_j$ because for all $k \geq 1$ it holds $\Xi_k \subseteq 2^X$. Since there exist at most $2^{|X|}$ languages $L(H \mid S)$ to consider the problem is decidable. □

Algorithm 2, which is a generalization of Algorithm 1, can be used to check q -sync observability. Its correctness is proved by the following proposition which provides an explanation of the algorithm.

Proposition 24 Let Σ be a finite alphabet, and $\Sigma_i \subseteq \Sigma$, with $i = 1, \dots, n$, be n sub-alphabets of Σ . Let L and K be two regular languages such that $K \subset L \subseteq \Sigma^*$. The language K is q -sync observable wrt L and $\{\Sigma_i \mid i = 1, \dots, n\}$ if Algorithm 2 returns true.

Algorithm 2 Analysis of q -sync observability

Input: DFA $H = (X, \Sigma, \delta, x_0, X_m \cup \hat{X}_m)$ built according to Def. 15 from languages L and K ; a positive integer q ; a set of alphabets $\{\Sigma_i \mid i = 1, \dots, n\}$

Output: A boolean variable QSO specifying if K is q -sync observable wrt L and $\{\Sigma_i \mid i = 1, \dots, n\}$

```
1: Let  $QSO = true$ .
2: Let  $S = \{x_0\}$ ,  $\Xi = \{S\}$  and label  $S$  new.
3: while  $QSO = true$  and  $\exists$  a set  $S \in \Xi$  labeled new do
4:   Select a set  $S \in \Xi$  labeled new and label it old.
5:   Compute the set of words of length less than or equal to  $q$  that can be generated by  $H$  starting from states in  $S$ .
6:   Partition this set into equivalence classes  $\Pi = \{\mathcal{W}_1, \dots, \mathcal{W}_r\}$  according to Def. 12.
7:   for each class  $\mathcal{W}_g \in \Pi$  do
8:     Let  $\mathcal{S}_g = \{\delta^*(x, w) \mid x \in S, w \in \mathcal{W}_g\}$ .
9:     if  $\mathcal{S}_g \notin \Xi$  then
10:      if  $\mathcal{S}_g \cap X_m \neq \emptyset$  and  $\mathcal{S}_g \cap \hat{X}_m \neq \emptyset$  then
11:        let  $QSO = false$ .
12:      end if
13:      Let  $\Xi = \Xi \cup \mathcal{S}_g$  and label  $\mathcal{S}_g$  as new.
14:    end if
15:  end for
16: end while
```

PROOF. The set Ξ contains subsets of states in which the system can be after an observation. Obviously, if one of these sets contains states in X_m (reached by words in K) and states in \hat{X}_m (reached by words in $L \setminus K$), the system is not q -sync observable. Notice that the algorithm simply computes all possible states of the system starting from the previous possible states and finishes when all possible sets are checked. \square

The complexity of Algorithm 2 is $2^{|X|} \cdot |\Sigma|^q$.

Example 25 Consider again the DFA in Fig. 3. Let it be the DFA H built according to Definition 15 where $X_m = \{x_0, \dots, x_6\}$ and $\hat{X}_m = \{x_7\}$. This means that L is the language accepted by such DFA, while K is the languages accepted by the same DFA, neglecting x_7 . Again assume $\Sigma_1 = \{a\}$, $\Sigma_2 = \{b\}$, $\Sigma_3 = \{c\}$. We want to apply Algorithm 2 to check 3-sync observability.

- Initially, $S = \{x_0\}$ and $\Xi = \{S\}$.
- In Step 5, all words of length less than or equal to 3 terminating in a state in $X_m \cup \hat{X}_m = \{x_0, \dots, x_7\}$ are computed starting from x_0 , i.e., ε , a , b , ab , ba , abb and bac .
- For these words, the equivalence classes are: $\{\varepsilon\}$, $\{a\}$, $\{b\}$, $\{ab, ba\}$, $\{abb\}$ and $\{bac\}$. All such words starting from x_0 lead to a state in X_m , i.e., they are contained in K , so the boolean variable is not updated.
- The corresponding sets of final states computed at Step 8 are: $\{x_0\}$, $\{x_1\}$, $\{x_4\}$, $\{x_2, x_5\}$, $\{x_3\}$ and $\{x_6\}$.
- All these sets minus $\{x_0\}$ are not in Ξ . Moreover, the condition in Step 10 is not satisfied hence are introduced in Ξ and labeled them as new (Step 13).
- The while loop is iterated. Assume $S = \{x_2, x_5\}$ is considered and thus labeled as old.
- The following words are computed in Step 5: ε , b , ba , bc , baa , bab , bca , $bc b$, bcc generated from state x_2 and ε , c , cb , cc , cba , cbb , cca , ccb , ccc generated from state x_5 .
- In the next step, these words are partitioned in equivalence classes: $\{\varepsilon\}$, $\{b\}$, $\{c\}$, $\{ba\}$, $\{bc, cb\}$, $\{cc\}$, $\{baa\}$, $\{bab\}$, $\{bca, cba\}$, $\{bc b, cbb\}$, $\{bcc, ccb\}$, $\{cca\}$, $\{ccc\}$.
- The corresponding sets of final states are: $\{x_2, x_5\}$, $\{x_3\}$, $\{x_6\}$, $\{x_0\}$, $\{x_7, x_0\}$, $\{x_7\}$, $\{x_1\}$, $\{x_4\}$, $\{x_7, x_1\}$, $\{x_7, x_4\}$, $\{x_7\}$, $\{x_7\}$, $\{x_7\}$.
- Since some of the above sets simultaneously contain states in X_m and states in \hat{X}_m , e.g., $\{x_7, x_0\}$ the algorithm returns false. \blacksquare

6 Computation of synchronizing time instants

In this section we focus on regular languages L and K and assume that all the events are observable by at least one site, namely $\Sigma = \cup_{i=1}^n \Sigma_i$ (see the following Remark 1 for a discussion on the requirement of this assumption).

We consider the following problem that may occur in several real applications. We want to determine a set of instants of synchronization of the different sites, so that a word in $L \setminus K$ is identified exactly *as soon as* it occurs. In particular, we propose a solution that is based on the idea of moving off-line most of the calculations. To this aim, as in the previous sections, we associate minimal DFA to languages L and K , denoted as G_L and G_K , respectively, and build the DFA H according to Def. 15. We provide an algorithm that, given an arbitrary state x of H , computes a set of synchronization instants \mathcal{I}_x to guarantee that the occurrence of any word in $L \setminus K$ generated starting from x , is immediately detected. Clearly, if k is the length of the shortest word in $L \setminus K$ generated starting from x , then the last synchronization should occur after k instants, namely after the occurrence of k events, being by assumption $\Sigma = \cup_{i=1}^n \Sigma_i$. The proposed solution also guarantees (as formally proved later on) that, for any state $x \in X_m \cup \hat{X}_m$, where $X_m \cup \hat{X}_m$ is the set of final states of H , the state reached from x after the occurrence of k events is uniquely identified. As a result, the computation of sets \mathcal{I}_x , for all $x \in X_m \cup \hat{X}_m$, can be executed off-line, and the resulting outputs are used on-line (without further computations) to establish when synchronizations should occur.

Before formalizing the proposed solution, we provide a revised definition of sync equivalent sequences (Def. 19) in order to keep into account synchronization instants.

Definition 26 *Let Σ be a finite alphabet, and $\Sigma_i \subseteq \Sigma$, with $i = 1, \dots, n$, be n sub-alphabets of Σ . Consider two words $w, v \in \Sigma^*$ having the same length k . Let $\mathcal{I} = \{p_1, p_2, \dots, p_I\} \subseteq \{1, 2, \dots, k\}$ be a set of synchronizing instants, where $p_I = k$.*

We say that w and v are \mathcal{I} -sync observable equivalent, or simply \mathcal{I} -equivalent where clear from the context, if $P_i(w_j) = P_i(v_j)$ for all $i = 1, \dots, n$, and all $j = 1, \dots, I$, where $w_1 = w(1 : p_1)$, $v_1 = v(1 : p_1)$, $w_j = w(p_{j-1} + 1 : p_j)$, $v_j = v(p_{j-1} + 1 : p_j)$ for all $j = 2, \dots, I$.

In simple words, two sequences w and v with the same length k , are \mathcal{I} -equivalent if the subsequences resulting from synchronizing at the set of instants \mathcal{I} , generate the same projections over all alphabets Σ_i 's.

Finally note that, if \mathcal{I} is a singleton, namely it is $\mathcal{I} = \{k\}$, then two sequences w and v are \mathcal{I} -equivalent if and only if w and v are equivalent according to Def. 12.

Algorithm 3 formalizes the proposed solution.

Algorithm 3 Computation of synchronization instants

Input: DFA $H = (X, \Sigma, \delta, x_0, X_m \cup \hat{X}_m)$ built according to Def. 15, and a state $x \in X_m \cup \hat{X}_m$

Output: The set of synchronization instants \mathcal{I}_x

- 1: Let k be the length of the shortest word starting from x and leading to a state in \hat{X}_m .
 - 2: Let $\mathcal{I}_x = \{k\}$.
 - 3: Compute the set of words \mathcal{L} of length k that can be generated by H starting from x and reaching a state in $X_m \cup \hat{X}_m$.
 - 4: Partition \mathcal{L} into the set of \mathcal{I}_x -equivalence classes according to Def. 26.
 - 5: **while** there exists a class $\mathcal{W} \in \Pi$ such that $w, v \in \mathcal{W}$ and $\delta^*(x_0, w) \neq \delta^*(x_0, v)$ **do**
 - 6: Choose a set $\mathcal{W} \in \Pi$ such that $w, v \in \mathcal{W}$ and $\delta^*(x_0, w) \neq \delta^*(x_0, v)$.
 - 7: Compute an integer $p < k$ such that w and v are not \mathcal{I} -equivalent for $\mathcal{I} = \mathcal{I}_x \cup \{p\}$.
 - 8: Let $\mathcal{I}_x = \mathcal{I}_x \cup \{p\}$.
 - 9: Partition \mathcal{L} into the set of \mathcal{I}_x -equivalence classes according to Def. 26.
 - 10: **end while**
-

Remark 1 *The assumption $\Sigma = \cup_{i=1}^n \Sigma_i$ guarantees that the algorithm terminates in a finite number of steps. In fact, at Step 7, for any two words w and v that are \mathcal{I}_x -equivalent (where clearly it is $w \neq v$), there always exists an integer p such that, if a new synchronization is added at instant p , then the two words w and v become distinguishable. Therefore, a solution at Step 7 always exists, which in turn guarantees that the algorithm terminates in a finite number of steps. In the worst case, \mathcal{I}_x contains all integers from 1 to k .*

The following proposition proves that, after the last synchronization in \mathcal{I}_x , the state of H is uniquely determined. This is clearly not true after intermediate synchronizations.

Proposition 27 *Let H be the DFA built according to Def. 15 and $x \in X_m \cup \hat{X}_m$ be one of its final states. Let \mathcal{I}_x be the set of synchronization instants computed using Algorithm 3, and k be its largest entry. The state reached from x after k events is uniquely determined.*

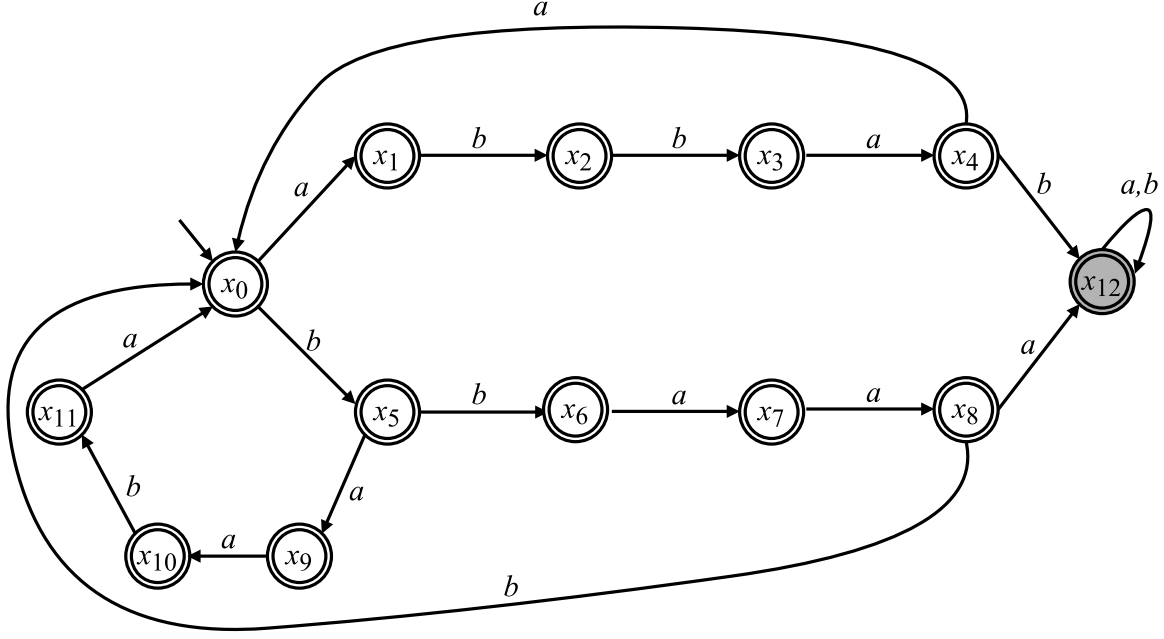


Fig. 4. The DFA H in Example 28.

PROOF. Follows from the fact that by repeatedly iterating Step 7, all words of length k that lead to different states are distinguishable. \square

The complexity of Algorithm 3 is $O(n \cdot |\Sigma|^{2|X|} \cdot |X|)$ where n is the number of sites. Indeed, k is bounded by $|X|$ thus the maximum number of words in \mathcal{L} of length k computed at Step 3 is bounded by $|\Sigma|^{|X|}$. Thus, testing the condition of the while loop (Steps 5) requires a number of operations bounded by $|\Sigma|^{2|X|} \cdot n$ to compare the projections of all pairs strings with respect to the n alphabets Σ_i 's. Finally, each time the while loop is executed the set \mathcal{I}_x increases, thus the loop is executed a number of times upper-bounded by k .

Example 28 Consider the DFA in Fig. 4. Let it be the DFA H built according to Definition 15 where $X_m = \{x_0, \dots, x_{11}\}$ and $\tilde{X}_m = \{x_{12}\}$. This means that L is the language accepted by such DFA, while K is the language accepted by the same DFA, neglecting state x_{12} . Assume $\Sigma_1 = \{a\}$, $\Sigma_2 = \{b\}$ and focus on state x_0 . Let us apply Algorithm 3 to compute the set \mathcal{I}_{x_0} .

- The length of the shortest path from x_0 to state x_{12} is $k = 5$. Hence, we initially take $\mathcal{I}_{x_0} = \{5\}$ (Step 2).
- The set of words of length $k = 5$ starting from x_0 is $\{abbaa, abbab, bbaaa, bbaab, baaba\}$ (Step 3). Therefore, we can define two \mathcal{I}_{x_0} -equivalence classes, i.e., $\Pi = \{\mathcal{W}_1, \mathcal{W}_2\}$ where $\mathcal{W}_1 = \{abbaa, bbaaa, baaba\}$ and $\mathcal{W}_2 = \{abbab, bbaab\}$ (Step 4).
- Condition in Step 5 is true, indeed the two words $w = abbaa$, $v = bbaaa$, both in \mathcal{W}_1 lead to different states, namely it is $\delta^*(x_0, w) = x_0 \neq \delta^*(x_0, v) = x_{12}$.
- We select an \mathcal{I}_{x_0} -equivalence class, e.g., $\bar{\mathcal{W}} = \mathcal{W}_1$ containing two words $w = abbaa$, $v = bbaaa$, which lead to different states (Step 6).
- We compute an index p to distinguish $w = abbaa$ from $v = bbaaa$ (Step 7). One possible solution is $p = 1$. Indeed, $w = w_1w_2$ with $w_1 = a$, $w_2 = bbaa$, $v = v_1v_2$ with $v_1 = b$, $v_2 = baaa$, and $P_1(w_1) = a \neq P_1(v_1) = \varepsilon$.
- We update the set of synchronization instants at $\mathcal{I}_{x_0} = \{1, 5\}$ (Step 8).
- The \mathcal{I}_{x_0} -equivalence classes now are: $\mathcal{W}'_1 = \{abbaa\}$, $\mathcal{W}''_1 = \{bbaaa, baaba\}$, $\mathcal{W}'_2 = \{abbab\}$ and $\mathcal{W}''_2 = \{bbaab\}$.
- We select an \mathcal{I}_{x_0} -equivalence class, e.g., $\bar{\mathcal{W}} = \mathcal{W}''_1$ containing two words that lead to different states: $w = bbaaa$ and $v = baaba$ (Step 6).
- We compute an index p to distinguish $w = bbaaa$ from $v = baaba$ (Step 7). One possible solution is $p = 2$.
- We update the set of synchronization instants at $\mathcal{I}_{x_0} = \{1, 2, 5\}$ (Step 8). Since now, the \mathcal{I}_{x_0} -equivalence classes are singleton, the algorithm stops. \blacksquare

7 Conclusions and future work

This paper deals with the problem of establishing if a given behavior belongs to a reference language K , based on decentralized observation, and a coordinator. Two different properties have been defined: uniform q -observability and q -sync observability, that differ for the criterion used to synchronize the different sites. Finally, an algorithm to compute the instants in which synchronizations should occur, has been given. It guarantees that the occurrence of any word not in K is detected as soon as it has occurred.

Our future research will be devoted to provide a criterion for an optimal choice of synchronizing instants, and address the problem of delays in the detection of words not in K . Investigating what happens in an asynchronous setting will be another object of our future work, e.g., taking inspiration by [21].

We conclude with a final remark. In this work we have studied decentralized observability with synchronizations from the point of view of a set of coordinated agents that locally observe a system with the goal of understanding if its evolution belongs to a reference behavior. The evolution of the system cannot be modified and the only decision variables are the synchronization instants in which the agents communicate with the coordinator. We believe that in this same framework other meaningful problems may be addressed, including supervisory control with the objective of enforcing (or enhancing) observability or, more generally, of enforcing an arbitrary given specification.

References

- [1] S. Tripakis, Undecidable problems of decentralized observation and control on regular languages, *Information Processing Letters* 90 (1) (2004) 21–28 (2004).
- [2] M. Sampath, R. Sengupta, S. Lafortune, K. Sinnamohideen, D. Teneketzis, Diagnosability of Discrete-Event Systems, *IEEE Trans. on Automatic Control* 40 (9) (1995) 1555–1575 (1995).
- [3] M. Cabasino, A. Giua, C. Mahulea, C. Seatzu, On decentralized observability of discrete event systems, in: *Proc. 50th IEEE Conference on Decision and Control*, 2011 (Dec. 2011).
- [4] P. E. Caines, R. Greiner, S. Wang, Dynamical logic observers for finite automata, in: *Proc. 27th IEEE Conference of Decision and Control*, 1988, pp. 226–233 (December 1988).
- [5] P. E. Caines, S. Wang, Classical and logic based regulator design and its complexity for partially observed automata, in: *Proc. of 28th Conference on Decision and Control*, 1989, pp. 132–137 (1989).
- [6] C. M. Özveren, A. S. Willsky, Observability of discrete event dynamic systems, *IEEE Transactions on Automatic Control* 35 (7) (1990) 797–806 (1990).
- [7] R. Kumar, V. Garg, S. I. Markus, Predicates and predicate transformers for supervisory control of discrete event dynamical systems, *IEEE Trans. on Automatic Control* 38 (2) (1993) 232–247 (1993).
- [8] G. Barrett, S. Lafortune, Decentralized supervisory control with communicating controllers, *IEEE Trans. on Automatic Control* 45 (9) (2000) 1620–1638 (Sep. 2000). doi:10.1109/9.880613.
- [9] E. Fabre, A. Benveniste, Partial order techniques for distributed discrete event systems: Why you cannot avoid using them, *Discrete Event Dynamic Systems* 17 (3) (2007) 355–403 (2007).
- [10] A. Giua, C. Seatzu, Observability of Place/Transition Nets, *IEEE Trans. on Automatic Control* 47 (9) (2002) 1424–1437 (2002).
- [11] D. Corona, A. Giua, C. Seatzu, Marking estimation of Petri nets with silent transitions, *IEEE Trans. on Automatic Control* 52 (9) (2007) 1695–1699 (2007).
- [12] L. Li, C. N. Hadjicostis, Reconstruction of transition firing sequences based on asynchronous observations of place token changes, in: *46th IEEE Conference on Decision and Control*, 2007, pp. 1898–1903 (2007).
- [13] C. N. Hadjicostis, C. Seatzu, Decentralized state estimation in discrete event systems under partially ordered observation sequences, in: *WODES 2016: Workshop on Discrete Event Systems*, 2016, pp. 393–398 (2016).
- [14] S. Yokota, T. Yamamoto, S. Takai, Computation of the delay bounds and synthesis of diagnosers for decentralized diagnosis with conditional decisions, *Discrete Event Dynamic Systems* (2016) 1–40 (2016).
- [15] L. K. Carvalho, M. V. Moreira, J. C. Basilio, S. Lafortune, Robust diagnosis of discrete-event systems against permanent loss of observations, *Automatica* 49 (1) (2013) 223 – 231 (2013).
- [16] X. Yin, Z. Li, Decentralized fault prognosis of discrete event systems with guaranteed performance bound, *Automatica* 69 (2016) 375 – 379 (2016).
- [17] S. Takai, Robust prognosability for a set of partially observed discrete event systems, *Automatica* 51 (2015) 123 – 130 (2015).
- [18] R. Zhang, K. Cai, Y. Gan, W. M. Wonham, Distributed supervisory control of discrete-event systems with communication delay, *Discrete Event Dynamic Systems* 26 (2) (2016) 263–293 (2016).
- [19] D. Sears, K. Rudie, Minimal sensor activation and minimal communication in discrete-event systems, *Discrete Event Dynamic Systems* 26 (2) (2016) 295–349 (2016).
- [20] J. Hopcroft, R. Motwani, J. Ullman, *Introduction to Automata Theory, Languages and Computation* (Third Edition), Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 2006 (2006).
- [21] S. Tripakis, Decentralized control of discrete-event systems with bounded or unbounded delay communication, *IEEE Transactions on Automatic Control* 49 (9) (2004) 1489–1501 (2004).