Nonexistence of Upper Bound to Inferencing Level in Decentralized Discrete Event Control

Shigemasa Takai, *Senior Member, IEEE*, and Ratnesh Kumar, *Fellow, IEEE*

***Abstract*—In the authors' earlier work, the notion of inference-observability was introduced to characterize the existence of decentralized supervisors that perform multilevel inferencing against self-ambiguity and the ambiguities of others to jointly arrive at a correct control decision. When the property of N-inference-observability holds, N-levels of inferencing are needed. We show in this article that the class of N-inference-observable languages increases *strictly* monotonically as the parameter N is increased. We further show that, in general, there is no upper bound on the number of levels of inferencing required.**

***Index Terms*—Decentralized supervisory control, discrete event system (DES), inference-observability, inferencing.**

I. Introduction

In any decentralized decision-making paradigm, such as decentralized control or diagnosis, multiple decision-makers, each with its own limited sensing capabilities, interact to come up with global decisions. The presence of limited sensing capabilities can lead to *ambiguity* in knowing the system state, and thereby, ambiguity in decision-making. In the context of control of discrete event systems (DESs), a knowledge-based mechanism for assessing self-ambiguity was presented in [7], and later the same architecture was used for assessing self-ambiguity as well as ambiguities of the others in [8]. The process of utilizing the knowledge of self-ambiguity together with ambiguities of the others for the sake of decision-making was referred to as "inferencing" in [8], and for the special case of single-level inferencing, it was called "conditioning" in [14] and [16]. A framework allowing multilevel inferencing over various local control decisions of varying levels of ambiguity was first introduced in [3]. This framework supports inferencing over an *arbitrary* number of levels of ambiguities, and in addition, an a priori partitioning of controllable events into disjunctive/conjunctive classes as in [15] and [16] is not required. An approach for synthesizing a sublanguage (respectively, a superlanguage) of a specification language that can be enforced using the inference-based decentralized supervisors is presented in [9] (respectively, [10]). An arborescent architecture was proposed to realize the inference-based decentralized supervisors in [2]. A similar inference-based framework for the management of ambiguities in the decentralized diagnosis (respectively, prognosis) of failures was reported in [4] and [12] (respectively, [11]).

In the inference-based decentralized control framework of [3], each local supervisor uses its observations of the system behavior to come up with its control decision together with a grade or level of ambiguity for that control decision. A local supervisor issues a control decision with an ambiguity level N when it knows that for each trace that causes ambiguity, there exists another local supervisor, which can issue a control decision with an ambiguity level at most $N-1$. A global control decision is chosen to be the same as a local control decision with the minimal level of ambiguity. The notion of N*-inference-observability* was formulated in [3] to characterize the class of languages achievable using N-levels of inferencing.

In this article, we establish that the class of N-inference-observable languages increases *strictly* monotonically as the parameter N is increased by showing that for any N, in general N-inference-observability is *strictly* stronger than $(N+1)$-inference-observability. This result answers an open question whether or not, even in the setting of finite-state plant and specification models, the number of levels of inferencing required is in general unbounded (and so in general, such a number should be decided a priori based on the available computing resources).

II. Notation and Preliminaries

We consider a DES modeled by an automaton $G = (Q, \Sigma, \delta, q_0, Q_m)$, where Q is the set of states, Σ is the finite set of events, a partial function $\delta : Q \times \Sigma \to Q$ is the transition function, $q_0 \in Q$ is the initial state, and $Q_m \subseteq Q$ is the set of marked states. Let Σ^* be the set of all finite traces of elements of Σ, including the empty trace ε. The function δ can be generalized to $\delta : Q \times \Sigma^* \to Q$ in a usual way. The generated and marked languages of G, denoted by $L(G)$ and $L_m(G)$, respectively, are defined as $L(G) = \{s \in \Sigma^* \mid (\exists q \in Q)\delta(q_0, s) = q\}$ and $L_m(G) = \{s \in \Sigma^* \mid \delta(q_0, s) \in Q_m\}$. Let $K \subseteq \Sigma^*$ be a language. The set of all prefixes of traces in K is denoted by $\overline{K}$. For supervisory control purposes [6], the event set Σ is partitioned into two disjoint subsets Σ_c and Σ_{uc} of controllable and uncontrollable events, respectively. K is said to be controllable if $\overline{K}\Sigma_{uc} \cap L(G) \subseteq \overline{K}$ [6].

Let the set $C = \{0, 1, \phi\}$ be the set of control decisions, where "0" represents a disablement decision, "1" represents an enablement decision, and "ϕ" represents an unsure (or pass) decision. Formally, a supervisor is defined as a map $S : L(G) \times \Sigma \to C$ such that $S(s, \sigma) = 1$ for any $s \in L(G)$ and any $\sigma \in \Sigma_{uc}$. We inductively define the generated language $L(S/G)$ under the control action of S as follows:

1) $\varepsilon \in L(S/G)$;
2) $(\forall s \in L(S/G))(\forall \sigma \in \Sigma)$

$$s\sigma \in L(S/G) \Leftrightarrow [s\sigma \in L(G) \wedge S(s, \sigma) = 1].$$

S is said to be *valid* when for any $s\sigma \in L(G) \cap L(S/G)\Sigma$, $S(s, \sigma) \neq \phi$, i.e., none of the control decisions for feasible events are unsure.

Manuscript received 16 March 2023; revised 14 November 2023 and 3 April 2024; accepted 23 May 2024. Date of publication 6 June 2024; date of current version 25 October 2024. This work was supported in part by the National Science Foundation under Grant NSF-CSSI-2004766 and Grant NSF-PFI-2141084, and in part by JSPS KAKENHI under Grant JP22K04167. Recommended by Associate Editor J. Komenda. *(Corresponding author: Shigemasa Takai.)*

Shigemasa Takai is with the Division of Electrical, Electronic and Infocommunications Engineering, Osaka University, Suita 565-0871, Japan (e-mail: takai@eei.eng.osaka-u.ac.jp).

Ratnesh Kumar is with the Department of Electrical and Computer Engineering, Iowa State University, Ames, IA 50011 USA (e-mail: rkumar@iastate.edu).

Digital Object Identifier 10.1109/TAC.2024.3410832

III. Review of the Inference-Based Decentralized Control Framework

We review the inference-based decentralized control framework introduced in [3]. In the decentralized control setting, there exist n local supervisors, whose decisions are fused to obtain a global control decision so that the controlled behavior satisfies a (global) specification. Let Σ_{ic} be the set of locally controllable events for the ith local supervisor S_i ($i \in I := \{1, 2, \ldots, n\}$), in which case, $\Sigma_c := \bigcup_{i\in I} \Sigma_{ic}$. For each controllable event $\sigma \in \Sigma_c$, the index set of local supervisors for which σ is controllable is denoted by $In(\sigma) = \{i \in I \mid \sigma \in \Sigma_{ic}\}$. The limited sensing capability of the ith local supervisor S_i ($i \in I$) is represented by a local observation mask, $M_i : \Sigma \to \Delta_i \cup \{\varepsilon\}$, where Δ_i is the set of locally observed symbols. The observation mask M_i can be extended to $M_i : \Sigma^* \to \Delta_i^*$ in a usual way. Two traces $s, s' \in \Sigma^*$ with $M_i(s) = M_i(s')$ are said to be M_i-indistinguishable. For any languages $L \subseteq \Sigma^*$ and $L' \subseteq \Delta_i^*$, $M_i(L) \subseteq \Delta_i^*$ and $M_i^{-1}(L') \subseteq \Sigma^*$ are defined as $M_i(L) = \{M_i(s) \in \Delta_i^* \mid s \in L\}$ and $M_i^{-1}(L') = \{s \in \Sigma^* \mid M_i(s) \in L'\}$, respectively.

Each inference-based local supervisor S_i is defined as a map $S_i : M_i(L(G)) \times \Sigma_{ic} \to C \times \mathbb{N}$, where $\mathbb{N}$ denotes the set of nonnegative integers, so that for any $s \in L(G)$ and any $\sigma \in \Sigma_{ic}$,

$$S_i(M_i(s), \sigma) = (c_i(M_i(s), \sigma), n_i(M_i(s), \sigma)).$$

Here, $c_i(M_i(s), \sigma) \in C$ denotes the control decision of S_i for a locally controllable event $\sigma \in \Sigma_{ic}$ following an observation $M_i(s) \in M_i(L(G))$, and $n_i(M_i(s), \sigma) \in \mathbb{N}$ denotes the ambiguity level of the control decision $c_i(M_i(s), \sigma)$. Let $n(s, \sigma)$ be the minimal ambiguity level of local decisions, i.e.,

$$n(s, \sigma) := \min\{n_i(M_i(s), \sigma) \in \mathbb{N} \mid i \in In(\sigma)\}.$$

The decentralized supervisor $\{S_i\}_{i\in I}$ that consists of local supervisors S_i ($i \in I$) issues global decisions on controllable events. For simplicity, with a slight abuse of notation, $\{S_i\}_{i\in I}$ is defined as a map $\{S_i\}_{i\in I} : L(G) \times \Sigma \to C$ (as opposed to the accurate notation, $\{S_i\}_{i\in I} : \prod_{i\in I} M_i(L(G)) \times \Sigma \to C$). For any $s \in L(G)$ and any $\sigma \in \Sigma$, the control decision $\{S_i\}_{i\in I}(s, \sigma)$ is given as follows.

1) If $\sigma \in \Sigma_c$

$$\{S_i\}_{i\in I}(s,\sigma) = \begin{cases} 1, & \text{if } (\forall i \in In(\sigma)) n_i(M_i(s),\sigma) = n(s,\sigma) \\ & \quad \Rightarrow c_i(M_i(s),\sigma) = 1 \\ 0, & \text{if } (\forall i \in In(\sigma)) n_i(M_i(s),\sigma) = n(s,\sigma) \\ & \quad \Rightarrow c_i(M_i(s),\sigma) = 0 \\ \phi, & \text{otherwise.} \end{cases}$$

2) If $\sigma \in \Sigma_{uc}$, $\{S_i\}_{i\in I}(s, \sigma) = 1$.

In other words, for a controllable event, a global control decision is taken to be the same as the minimal ambiguity level local control consensus decision. Such local control consensus decisions are called "winning" decisions.

A useful notion of a decentralized supervisor is the boundedness of the ambiguity level of its winning decisions. A decentralized supervisor is said to be N-inferring if for each controllable event, all winning enablement or all winning disablement decisions have ambiguity levels below N.

Definition 1 ([3]): Given a nonnegative integer $N \in \mathbb{N}$, a decentralized supervisor $\{S_i\}_{i\in I} : L(G) \times \Sigma \to C$ is said to be N-*inferring* if it is valid and for each $\sigma \in \Sigma_c$, either

$$(\forall s \in L(\{S_i\}_{i\in I}/G)) \quad [s\sigma \in L(G) \wedge \{S_i\}_{i\in I}(s,\sigma) = 0] \Rightarrow n(s,\sigma) \leq N$$

or

$$(\forall s \in L(\{S_i\}_{i\in I}/G)) \quad [s\sigma \in L(G) \wedge \{S_i\}_{i\in I}(s,\sigma) = 1] \Rightarrow n(s,\sigma) \leq N.$$

Given a specification language $K \subseteq L(G)$ of the plant G, a pair of sublanguages of $\overline{K}$ is constructed for each controllable event $\sigma \in \Sigma_c$. The set $D_{K,0}(\sigma) \subseteq \overline{K}$ is the set of traces in $\overline{K}$ where σ must be disabled (i.e., $s \in D_{K,0}(\sigma) \Leftrightarrow s\sigma \in L(G) - \overline{K}$), whereas the set $E_{K,0}(\sigma) \subseteq \overline{K}$ is the set of traces where σ must be enabled (i.e., $s \in E_{K,0}(\sigma) \Leftrightarrow s\sigma \in \overline{K}$). Using these as the base step, we inductively define a monotonically decreasing sequence of language pairs $(D_{K,h}(\sigma), E_{K,h}(\sigma))$ as follows:

1) $\begin{cases} D_{K,0}(\sigma) & := & \{s \in \overline{K} | \ s\sigma \in L(G) - \overline{K}\} \\ E_{K,0}(\sigma) & := & \{s \in \overline{K} | \ s\sigma \in \overline{K}\}. \end{cases}$

2) $\begin{cases} D_{K,h+1}(\sigma) \\ \quad := D_{K,h}(\sigma) \cap (\bigcap_{i\in In(\sigma)} M_i^{-1} M_i(E_{K,h}(\sigma))) \\ E_{K,h+1}(\sigma) \\ \quad := E_{K,h}(\sigma) \cap (\bigcap_{i\in In(\sigma)} M_i^{-1} M_i(D_{K,h}(\sigma))). \end{cases}$

Note that $D_{K,h+1}(\sigma)$ is a sublanguage of $D_{K,h}(\sigma)$ consisting of traces for each of which there exists an M_i-indistinguishable trace in $E_{K,h}(\sigma)$ for each $i \in In(\sigma)$. As a result, all the local supervisors that have control over σ will be ambiguous about their control decisions for σ following the execution of a trace in $D_{K,h+1}(\sigma)$. The sublanguage $E_{K,h+1}(\sigma)$ of $E_{K,h}(\sigma)$ can be understood in a similar fashion.

Then, we have the following definition of N-inference-observability.

Definition 2: [3] Given a nonnegative integer $N \in \mathbb{N}$, a language $K \subseteq L(G)$ is said to be N-*inference-observable* if for any $\sigma \in \Sigma_c$, $D_{K,N+1}(\sigma) = \emptyset$ or $E_{K,N+1}(\sigma) = \emptyset$. K is said to be *inference-observable* if there exists $N \in \mathbb{N}$ such that K is N-*inference-observable*.

Remark 1: It was shown in [3] that C&P∨D&A-coobservability [15] and conditional C&P ∨ D&A-coobservability [16] are equivalent to 0-inference-observability and 1-inference-observability, respectively.

The following theorem shows the necessity and sufficiency of N-inference-observability for the existence of an N-inferring decentralized supervisor enforcing the given specification.

Theorem 1 ([3]): For a nonempty language $K \subseteq L(G)$ and a nonnegative integer $N \in \mathbb{N}$, there exists an N-inferring decentralized supervisor $\{S_i\}_{i\in I} : L(G) \times \Sigma \to C$ such that $L(\{S_i\}_{i\in I}/G) = \overline{K}$ if and only if K is controllable and N-inference-observable.

IV. Strict Monotonicity of N-Inference-Observability w.r.t. N

It was shown in [3] that for any $N \in \mathbb{N}$, the property of N-inference-observability implies the property of $(N+1)$-inference-observability (implying that if a specification can be achieved with N-levels of inferencing, then it can also be achieved using larger than N-levels of inferencing). The question whether more levels of inferencing can help achieve a strictly larger class of specifications has remained open. Only in the cases of $N = 0$ and $N = 1$, it was shown that N-inference-observability is strictly stronger than $(N+1)$-inference-observability in general in [16] and [3], respectively, as shown in the following example.

Example 1: We consider the finite automaton G shown in Fig. 1 , where a double circle is used to identify a marked state. It is obtained by adding certain transitions labeled by a or b to the finite automaton presented in Fig. 1(a) of [3] to form cycles by $a'b$ or $b'a$.

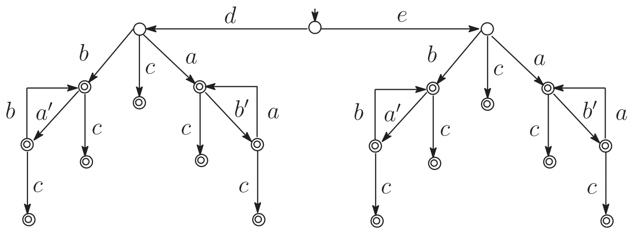

Fig. 1. Automaton G for Example 1 and proof of Theorem 2.

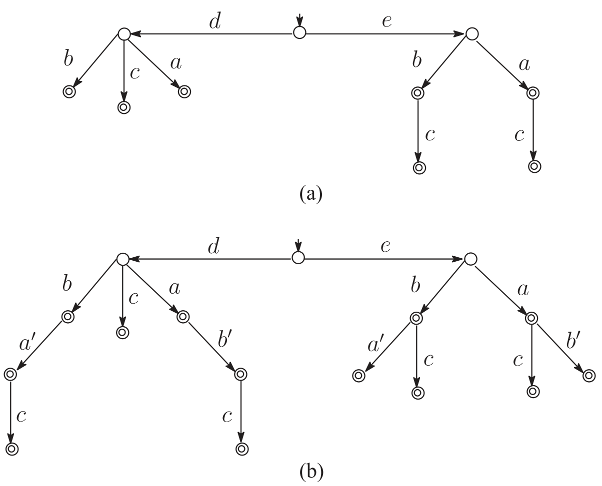

Fig. 2. Automata G_{K_1} and G_{K_2} for Example 1. (a) G_{K_1}. (b) G_{K_2}.

Let $n = 2$, $\Sigma_{1c} = \{a, a', c\}$, $\Sigma_{2c} = \{b, b', c\}$, $\Delta_1 = \{a, a', d, e\}$, $\Delta_2 = \{b, b', d, e\}$, and

$$M_1(\sigma) = \begin{cases} \sigma, & \text{if } \sigma \in \{a, a', d, e\} \\ \varepsilon, & \text{otherwise} \end{cases}$$

$$M_2(\sigma) = \begin{cases} \sigma, & \text{if } \sigma \in \{b, b', d, e\} \\ \varepsilon, & \text{otherwise.} \end{cases}$$

In addition, let K_1 and K_2 be the languages marked by the finite automata G_{K_1} and G_{K_2} shown in Fig. 2(a) and (b), respectively, which are taken from [15] and [16], respectively.

First, we show that K_1 is 1-inference-observable but not 0-inference-observable. For the events a, a', b, and b', we have $D_{K_1,0}(a) = \emptyset$, $E_{K_1,0}(a') = \emptyset$, $D_{K_1,0}(b) = \emptyset$, and $E_{K_1,0}(b') = \emptyset$. For the event c, we have

$$D_{K_1,0}(c) = d(a + b) + e$$
$$E_{K_1,0}(c) = d + e(a + b).$$

Since

$$M_1(D_{K_1,0}(c)) = d(a + \varepsilon) + e$$
$$M_2(D_{K_1,0}(c)) = d(\varepsilon + b) + e$$
$$M_1(E_{K_1,0}(c)) = d + e(a + \varepsilon)$$
$$M_2(E_{K_1,0}(c)) = d + e(\varepsilon + b)$$

we have

$$D_{K_1,1}(c) = D_{K_1,0}(c) \cap \left(\bigcap_{i \in \{1,2\}} M_i^{-1} M_i(E_{K_1,0}(c)) \right)$$
$$= e,$$
$$E_{K_1,1}(c) = E_{K_1,0}(c) \cap \left(\bigcap_{i \in \{1,2\}} M_i^{-1} M_i(D_{K_1,0}(c)) \right)$$
$$= d$$

which imply that K_1 is not 0-inference-observable. Two strings $da, db \in D_{K_1,0}(c)$ (respectively, $ea, eb \in E_{K_1,0}(c)$) which cannot be distinguished from d (respectively, e) by the second and first local supervisors, respectively, are used to show that $E_{K_1,1}(c) = d \neq \emptyset$ (respectively, $D_{K_1,1}(c) = e \neq \emptyset$).

However, since

$$M_1(D_{K_1,1}(c)) = M_2(D_{K_1,1}(c)) = e$$
$$M_1(E_{K_1,1}(c)) = M_2(E_{K_1,1}(c)) = d$$

we have

$$D_{K_1,2}(c) = D_{K_1,1}(c) \cap \left(\bigcap_{i \in \{1,2\}} M_i^{-1} M_i(E_{K_1,1}(c)) \right)$$
$$= \emptyset,$$
$$E_{K_1,2}(c) = E_{K_1,1}(c) \cap \left(\bigcap_{i \in \{1,2\}} M_i^{-1} M_i(D_{K_1,1}(c)) \right)$$
$$= \emptyset,$$

which imply that K_1 is 1-inference-observable.

Next, we show that K_2 is 2-inference-observable but not 1-inference-observable. For the event a, we have

$$D_{K_2,0}(a) = (d + e)ab'$$
$$E_{K_2,0}(a) = d + e.$$

Since $M_1(D_{K_2,0}(a)) = (d + e)a$ and $M_1(E_{K_2,0}(a)) = d + e$, we have

$$D_{K_2,1}(a) = D_{K_2,0}(a) \cap M_1^{-1} M_1(E_{K_2,0}(a)) = \emptyset$$
$$E_{K_2,1}(a) = E_{K_2,0}(a) \cap M_1^{-1} M_1(D_{K_2,0}(a)) = \emptyset.$$

For the event b, we have

$$D_{K_2,0}(b) = (d + e)ba'$$
$$E_{K_2,0}(b) = d + e.$$

Since $M_2(D_{K_2,0}(b)) = (d + e)b$ and $M_2(E_{K_2,0}(b)) = d + e$, we have

$$D_{K_2,1}(b) = D_{K_2,0}(b) \cap M_2^{-1} M_2(E_{K_2,0}(b)) = \emptyset$$
$$E_{K_2,1}(b) = E_{K_2,0}(b) \cap M_2^{-1} M_2(D_{K_2,0}(b)) = \emptyset.$$

For the events a' and b', we have $D_{K_2,0}(a') = D_{K_2,0}(b') = \emptyset$. For the event c, we have

$$D_{K_2,0}(c) = d(a + b) + e(\varepsilon + ab' + ba')$$
$$E_{K_2,0}(c) = d(\varepsilon + ab' + ba') + e(a + b).$$

Compared to $D_{K_1,0}(c)$ (respectively, $E_{K_1,0}(c)$), $D_{K_2,0}(c)$ (respectively, $E_{K_2,0}(c)$) contains eab' and eba' (respectively, dab' and dba') additionally. Since

$$M_1(D_{K_2,0}(c)) = d(a + \varepsilon) + e(\varepsilon + a + a')$$
$$M_2(D_{K_2,0}(c)) = d(\varepsilon + b) + e(\varepsilon + b' + b)$$

$$M_1(E_{K_2,0}(c)) = d(\varepsilon + a + a') + e(a + \varepsilon)$$
$$M_2(E_{K_2,0}(c)) = d(\varepsilon + b' + b) + e(\varepsilon + b)$$

we have

$$\begin{aligned} D_{K_2,1}(c) &= D_{K_2,0}(c) \cap \left(\bigcap_{i\in\{1,2\}} M_i^{-1}M_i(E_{K_2,0}(c)) \right) \\ &= d(a+b) + e \end{aligned}$$

$$\begin{aligned} E_{K_2,1}(c) &= E_{K_2,0}(c) \cap \left(\bigcap_{i\in\{1,2\}} M_i^{-1}M_i(D_{K_2,0}(c)) \right) \\ &= d + e(a+b). \end{aligned}$$

Since $D_{K_2,1}(c) = D_{K_1,0}(c)$ and $E_{K_2,1}(c) = E_{K_1,0}(c)$, we have

$$\begin{aligned} D_{K_2,2}(c) &= D_{K_2,1}(c) \cap \left(\bigcap_{i\in\{1,2\}} M_i^{-1}M_i(E_{K_2,1}(c)) \right) \\ &= D_{K_1,0}(c) \cap \left(\bigcap_{i\in\{1,2\}} M_i^{-1}M_i(E_{K_1,0}(c)) \right) \\ &= D_{K_1,1}(c) \\ &= e \end{aligned}$$

$$\begin{aligned} E_{K_2,2}(c) &= E_{K_2,1}(c) \cap \left(\bigcap_{i\in\{1,2\}} M_i^{-1}M_i(D_{K_2,1}(c)) \right) \\ &= E_{K_1,0}(c) \cap \left(\bigcap_{i\in\{1,2\}} M_i^{-1}M_i(D_{K_1,0}(c)) \right) \\ &= E_{K_1,1}(c) \\ &= d. \end{aligned}$$

Similarly, since $D_{K_2,2}(c) = D_{K_1,1}(c)$ and $E_{K_2,2}(c) = E_{K_1,1}(c)$, we have $D_{K_2,3}(c) = D_{K_1,2}(c) = \emptyset$ and $E_{K_2,3}(c) = E_{K_1,2}(c) = \emptyset$, which imply that K_2 is 2-inference-observable but not 1-inference-observable. Two strings $dab', dba' \in E_{K_2,0}(c)$ (respectively, $eab', eba' \in D_{K_2,0}(c)$) and their prefixes $da, db \in D_{K_2,0}(c)$ (respectively, $ea, eb \in E_{K_2,0}(c)$) are used to show that $E_{K_2,2}(c) = d \neq \emptyset$ (respectively, $D_{K_2,2}(c) = e \neq \emptyset$). The event a' (respectively, b') is introduced so that $dba' \in E_{K_2,0}(c)$ (respectively, $dab' \in E_{K_2,0}(c)$) for which c must be enabled can be distinguished from $da \in D_{K_2,0}(c)$ (respectively, $db \in D_{K_2,0}(c)$) for which c must be disabled by the first (respectively, second) local supervisor.

By extending Example 1, we show in the following theorem that even when the plant language is regular, the property of N-inference-observability is strictly stronger than the property of $(N+1)$-inference-observability for any $N \in \mathbb{N}$, i.e., the classes of N-inference-observable languages form a strictly increasing chain of languages in general.

Theorem 2: For any $N \in \mathbb{N}$, in general, N-inference-observability is strictly stronger than $(N+1)$-inference-observability.

Proof: By [3, Th. 2], N-inference-observability implies $(N+1)$-inference-observability for any $N \in \mathbb{N}$. We need to show that the reverse implication does not hold in general.

We consider the setting of Example 1, where G is the finite automaton shown in Fig. 1. In addition to the languages K_1 and K_2 marked by the finite automata G_{K_1} and G_{K_2} shown in Fig. 2(a) and (b), respectively, for any $N \geq 3$, let $K_N \subseteq L(G)$ be the language marked by the finite automaton G_{K_N} shown in Fig. 3. We show by induction on $l \in \mathbb{N}$ that K_{2l+1} is $(2l+1)$-inference-observable but not $2l$-inference-observable, and K_{2l+2} is $(2l+2)$-inference-observable but not $(2l+1)$-inference-observable, proving the desired strict monotonicity result.

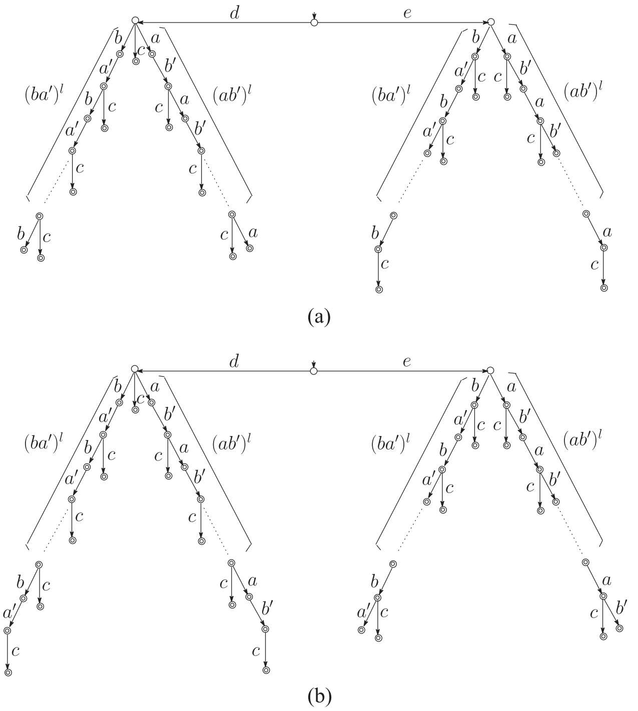

Fig. 3. Automaton G_{K_N} $(N \geq 3)$ for proof of Theorem 2. (a) $G_{K_{2l+1}} (l > 0)$. (b) $G_{K_{2l+2}} (l > 0)$.

The base step where $l = 0$ holds by Example 1. For the induction step, we suppose that, for $l = h \geq 0$, K_{2h+1} is $(2h+1)$-inference-observable but not $2h$-inference-observable, and K_{2h+2} is $(2h+2)$-inference-observable but not $(2h+1)$-inference-observable.

First, we show that $K_{2(h+1)+1}$ is $(2(h+1)+1)$-inference-observable but not $2(h+1)$-inference-observable. The language $K_{2(h+1)+1} \subseteq L(G)$ is marked by the finite automaton $G_{K_{2l+1}}$ shown in Fig. 3(a), where $l = h+1$. For the events a and b, we have $D_{K_{2(h+1)+1},0}(a) = \emptyset$ and $D_{K_{2(h+1)+1},0}(b) = \emptyset$. For the event a', we have

$$D_{K_{2(h+1)+1},0}(a') = (d+e)(ba')^{h+1}b$$

$$E_{K_{2(h+1)+1},0}(a') = (d+e)\left(\sum_{j=0}^{h}(ba')^j b\right).$$

Since

$$M_1(D_{K_{2(h+1)+1},0}(a')) = (d+e)(a')^{h+1}$$

$$M_1(E_{K_{2(h+1)+1},0}(a')) = (d+e)\left(\sum_{j=0}^{h}(a')^j\right)$$

we have

$$\begin{aligned} D_{K_{2(h+1)+1},1}(a') = D_{K_{2(h+1)+1},0}(a') \\ \cap M_1^{-1}M_1(E_{K_{2(h+1)+1},0}(a')) \end{aligned}$$

$$= \emptyset$$
$$E_{K_{2(h+1)+1},1}(a') = E_{K_{2(h+1)+1},0}(a') \cap M_1^{-1} M_1(D_{K_{2(h+1)+1},0}(a')) = \emptyset.$$

For the event b', we have

$$D_{K_{2(h+1)+1},0}(b') = (d+e)(ab')^{h+1}a$$
$$E_{K_{2(h+1)+1},0}(b') = (d+e)\left(\sum_{j=0}^{h}(ab')^j a\right).$$

Since

$$M_2(D_{K_{2(h+1)+1},0}(b')) = (d+e)(b')^{h+1}$$
$$M_2(E_{K_{2(h+1)+1},0}(b')) = (d+e)\left(\sum_{j=0}^{h}(b')^j\right)$$

we have

$$D_{K_{2(h+1)+1},1}(b') = D_{K_{2(h+1)+1},0}(b') \cap M_2^{-1} M_2(E_{K_{2(h+1)+1},0}(b')) = \emptyset$$
$$E_{K_{2(h+1)+1},1}(b') = E_{K_{2(h+1)+1},0}(b') \cap M_2^{-1} M_2(D_{K_{2(h+1)+1},0}(b')) = \emptyset.$$

For the event c, we have

$$D_{K_{2(h+1)+1},0}(c) = d\left(\sum_{j=0}^{h+1}((ab')^j a + (ba')^j b)\right) + e\left(\sum_{j=0}^{h+1}((ab')^j + (ba')^j)\right)$$
$$E_{K_{2(h+1)+1},0}(c) = d\left(\sum_{j=0}^{h+1}((ab')^j + (ba')^j)\right) + e\left(\sum_{j=0}^{h+1}((ab')^j a + (ba')^j b)\right).$$

Since

$$M_1(D_{K_{2(h+1)+1},0}(c)) = d\left(\sum_{j=0}^{h+1}(a^j a + (a')^j)\right) + e\left(\sum_{j=0}^{h+1}(a^j + (a')^j)\right)$$
$$M_2(D_{K_{2(h+1)+1},0}(c)) = d\left(\sum_{j=0}^{h+1}((b')^j + b^j b)\right) + e\left(\sum_{j=0}^{h+1}((b')^j + b^j)\right)$$
$$M_1(E_{K_{2(h+1)+1},0}(c)) = d\left(\sum_{j=0}^{h+1}(a^j + (a')^j)\right) + e\left(\sum_{j=0}^{h+1}(a^j a + (a')^j)\right)$$
$$M_2(E_{K_{2(h+1)+1},0}(c)) = d\left(\sum_{j=0}^{h+1}((b')^j + b^j)\right) + e\left(\sum_{j=0}^{h+1}((b')^j + b^j b)\right)$$

we have

$$D_{K_{2(h+1)+1},1}(c) = D_{K_{2(h+1)+1},0}(c) \cap \left(\bigcap_{i\in\{1,2\}} M_i^{-1} M_i(E_{K_{2(h+1)+1},0}(c))\right) = d\left(\sum_{j=0}^{h}((ab')^j a + (ba')^j b)\right) + e\left(\sum_{j=0}^{h+1}((ab')^j + (ba')^j)\right) = D_{K_{2(h+1)},0}(c)$$
$$E_{K_{2(h+1)+1},1}(c) = E_{K_{2(h+1)+1},0}(c) \cap \left(\bigcap_{i\in\{1,2\}} M_i^{-1} M_i(D_{K_{2(h+1)+1},0}(c))\right) = d\left(\sum_{j=0}^{h+1}((ab')^j + (ba')^j)\right) + e\left(\sum_{j=0}^{h}((ab')^j a + (ba')^j b)\right) = E_{K_{2(h+1)},0}(c).$$

Then, for any $N \in \mathbb{N}$ with $N \geq 1$, we have

$$D_{K_{2(h+1)+1},N}(c) = D_{K_{2(h+1)},N-1}(c)$$
$$E_{K_{2(h+1)+1},N}(c) = E_{K_{2(h+1)},N-1}(c).$$

Since $K_{2(h+1)}$ is $2(h+1)$-inference-observable but not $(2h+1)$-inference-observable by the inductive assumption, we can conclude that $K_{2(h+1)+1}$ is $(2(h+1)+1)$-inference-observable but not $2(h+1)$-inference-observable.

Next, we show that $K_{2(h+1)+2}$ is $(2(h+1)+2)$-inference-observable but not $(2(h+1)+1)$-inference-observable. The language $K_{2(h+1)+2} \subseteq L(G)$ is marked by the finite automaton $G_{K_{2l+2}}$ shown in Fig. 3(b), where $l = h+1$. For the event a, we have

$$D_{K_{2(h+1)+2},0}(a) = (d+e)(ab')^{h+2}$$
$$E_{K_{2(h+1)+2},0}(a) = (d+e)\left(\sum_{j=0}^{h+1}(ab')^j\right).$$

Since

$$M_1(D_{K_{2(h+1)+2},0}(a)) = (d+e)a^{h+2}$$

$$M_1(E_{K_{2(h+1)+2},0}(a)) = (d+e)\left(\sum_{j=0}^{h+1} a^j\right)$$

we have

$$\begin{aligned} D_{K_{2(h+1)+2},1}(a) &= D_{K_{2(h+1)+2},0}(a) \\ &\quad \cap M_1^{-1} M_1(E_{K_{2(h+1)+2},0}(a)) \\ &= \emptyset \\ E_{K_{2(h+1)+2},1}(a) &= E_{K_{2(h+1)+2},0}(a) \\ &\quad \cap M_1^{-1} M_1(D_{K_{2(h+1)+2},0}(a)) \\ &= \emptyset. \end{aligned}$$

For the event b, we have

$$D_{K_{2(h+1)+2},0}(b) = (d+e)(ba')^{h+2}$$

$$E_{K_{2(h+1)+2},0}(b) = (d+e)\left(\sum_{j=0}^{h+1} (ba')^j\right).$$

Since

$$M_2(D_{K_{2(h+1)+2},0}(b)) = (d+e)b^{h+2}$$

$$M_2(E_{K_{2(h+1)+2},0}(b)) = (d+e)\left(\sum_{j=0}^{h+1} b^j\right)$$

we have

$$\begin{aligned} D_{K_{2(h+1)+2},1}(b) &= D_{K_{2(h+1)+2},0}(b) \\ &\quad \cap M_2^{-1} M_2(E_{K_{2(h+1)+2},0}(b)) \\ &= \emptyset \\ E_{K_{2(h+1)+2},1}(b) &= E_{K_{2(h+1)+2},0}(b) \\ &\quad \cap M_2^{-1} M_2(D_{K_{2(h+1)+2},0}(b)) \\ &= \emptyset. \end{aligned}$$

For the events a' and b', we have $D_{K_{2(h+1)+2},0}(a') = \emptyset$ and $D_{K_{2(h+1)+2},0}(b') = \emptyset$. For the event c, we have

$$\begin{aligned} D_{K_{2(h+1)+2},0}(c) &= d\left(\sum_{j=0}^{h+1}((ab')^j a + (ba')^j b)\right) \\ &\quad + e\left(\sum_{j=0}^{h+2}((ab')^j + (ba')^j)\right) \\ E_{K_{2(h+1)+2},0}(c) &= d\left(\sum_{j=0}^{h+2}((ab')^j + (ba')^j)\right) \\ &\quad + e\left(\sum_{j=0}^{h+1}((ab')^j a + (ba')^j b)\right). \end{aligned}$$

Since

$$\begin{aligned} M_1(D_{K_{2(h+1)+2},0}(c)) &= d\left(\sum_{j=0}^{h+1}(a^j a + (a')^j)\right) \\ &\quad + e\left(\sum_{j=0}^{h+2}(a^j + (a')^j)\right) \\ M_2(D_{K_{2(h+1)+2},0}(c)) &= d\left(\sum_{j=0}^{h+1}((b')^j + b^j b)\right) \\ &\quad + e\left(\sum_{j=0}^{h+2}((b')^j + b^j)\right) \\ M_1(E_{K_{2(h+1)+2},0}(c)) &= d\left(\sum_{j=0}^{h+2}(a^j + (a')^j)\right) \\ &\quad + e\left(\sum_{j=0}^{h+1}(a^j a + (a')^j)\right) \\ M_2(E_{K_{2(h+1)+2},0}(c)) &= d\left(\sum_{j=0}^{h+2}((b')^j + b^j)\right) \\ &\quad + e\left(\sum_{j=0}^{h+1}((b')^j + b^j b)\right) \end{aligned}$$

we have

$$\begin{aligned} D_{K_{2(h+1)+2},1}(c) &= D_{K_{2(h+1)+2},0}(c) \\ &\quad \cap \left(\bigcap_{i\in\{1,2\}} M_i^{-1} M_i(E_{K_{2(h+1)+2},0}(c))\right) \\ &= d\left(\sum_{j=0}^{h+1}((ab')^j a + (ba')^j b)\right) \\ &\quad + e\left(\sum_{j=0}^{h+1}((ab')^j + (ba')^j)\right) \\ &= D_{K_{2(h+1)+1},0}(c) \\ E_{K_{2(h+1)+2},1}(c) &= E_{K_{2(h+1)+2},0}(c) \\ &\quad \cap \left(\bigcap_{i\in\{1,2\}} M_i^{-1} M_i(D_{K_{2(h+1)+2},0}(c))\right) \\ &= d\left(\sum_{j=0}^{h+1}((ab')^j + (ba')^j)\right) \\ &\quad + e\left(\sum_{j=0}^{h+1}((ab')^j a + (ba')^j b)\right) \\ &= E_{K_{2(h+1)+1},0}(c). \end{aligned}$$

Then, for any $N \in \mathbb{N}$ with $N \geq 1$, we have

$$D_{K_{2(h+1)+2},N}(c) = D_{K_{2(h+1)+1},N-1}(c)$$

$$E_{K_{2(h+1)+2},N}(c) = E_{K_{2(h+1)+1},N-1}(c).$$

Since $K_{2(h+1)+1}$ is $(2(h+1)+1)$-inference-observable but not $2(h+1)$-inference-observable, we can conclude that $K_{2(h+1)+2}$ is $(2(h+1)+2)$-inference-observable but not $(2(h+1)+1)$-inference-observable. ■

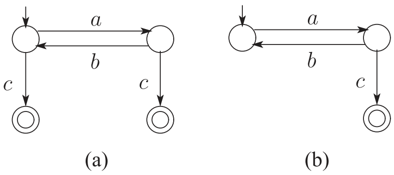

Fig. 4. Automata G and G_K for proof of Theorem 3. (a) G. (b) G_K.

V. Nonexistence of Upper Bound to Inference-Observability

Checking the inference-observability property of a language $K \subseteq L(G)$ requires checking the existence of $N \in \mathbb{N}$ such that for any $\sigma \in \Sigma_c$, $D_{K,N+1}(\sigma) = \emptyset$ or $E_{K,N+1}(\sigma) = \emptyset$. When the specification language is N-inference-observable for some $N \in \mathbb{N}$, the iterative computation of $(D_{K,h}(\sigma), E_{K,h}(\sigma))$ converges in a finite number of steps, i.e., $N+1$ steps. The following theorem shows that a finite-step convergence of this iterative computation is not guaranteed even if the plant and specification languages are regular and there are only two local supervisors.

Theorem 3: For a nonempty language $K \subseteq L(G)$ and a controllable event $\sigma \in \Sigma_c$, in general there does not exist $N \in \mathbb{N}$ such that $D_{K,N}(\sigma) = D_{K,N+1}(\sigma)$ and $E_{K,N}(\sigma) = E_{K,N+1}(\sigma)$.

Proof: We consider the finite automaton G shown in Fig. 4(a). Let $n = 2$, $\Sigma_c = \Sigma_{1c} = \Sigma_{2c} = \{c\}$, $\Delta_1 = \{a\}$, $\Delta_2 = \{b\}$, and

$$M_1(\sigma) = \begin{cases} \sigma, & \text{if } \sigma = a \\ \varepsilon, & \text{otherwise} \end{cases}$$

$$M_2(\sigma) = \begin{cases} \sigma, & \text{if } \sigma = b \\ \varepsilon, & \text{otherwise.} \end{cases}$$

Also, let $K \subseteq L(G)$ be a language marked by the finite automaton G_K shown in Fig. 4(b).

First, we show by induction on $h \in \mathbb{N}$ that

$$D_{K,2h}(c) = (ab)^h (ab)^*$$

$$E_{K,2h}(c) = a(ba)^h (ba)^*.$$

By the definitions, we have $D_{K,0}(c) = (ab)^*$ and $E_{K,0}(c) = a(ba)^*$. This establishes the base step where $h = 0$.

For the induction step, we suppose that $D_{K,2h}(c) = (ab)^h(ab)^*$ and $E_{K,2h}(c) = a(ba)^h(ba)^*$ for some $h \in \mathbb{N}$. Since

$$M_1(D_{K,2h}(c)) = a^h a^*$$

$$M_2(D_{K,2h}(c)) = b^h b^*$$

$$M_1(E_{K,2h}(c)) = a^{h+1} a^*$$

$$M_2(E_{K,2h}(c)) = b^h b^*$$

we have

$$D_{K,2h+1}(c) = D_{K,2h}(c) \cap \left(\bigcap_{i\in\{1,2\}} M_i^{-1} M_i(E_{K,2h}(c)) \right) = (ab)^{h+1}(ab)^*$$

$$E_{K,2h+1}(c) = E_{K,2h}(c) \cap \left(\bigcap_{i\in\{1,2\}} M_i^{-1} M_i(D_{K,2h}(c)) \right) = a(ba)^h(ba)^*.$$

Furthermore, since

$$M_1(D_{K,2h+1}(c)) = a^{h+1} a^*$$

$$M_2(D_{K,2h+1}(c)) = b^{h+1} b^*$$

$$M_1(E_{K,2h+1}(c)) = a^{h+1} a^*$$

$$M_2(E_{K,2h+1}(c)) = b^h b^*$$

we have

$$D_{K,2(h+1)}(c) = D_{K,2h+1}(c) \cap \left(\bigcap_{i\in\{1,2\}} M_i^{-1} M_i(E_{K,2h+1}(c)) \right) = (ab)^{h+1}(ab)^*$$

$$E_{K,2(h+1)}(c) = E_{K,2h+1}(c) \cap \left(\bigcap_{i\in\{1,2\}} M_i^{-1} M_i(D_{K,2h+1}(c)) \right) = a(ba)^{h+1}(ba)^*.$$

This completes the induction step.

Thus, we have $D_{K,2(h+1)}(c) \subset D_{K,2h}(c)$ and $E_{K,2(h+1)}(c) \subset E_{K,2h}(c)$ for any $h \in \mathbb{N}$. For the sake of contradiction, we suppose that there exists $N \in \mathbb{N}$ such that $D_{K,N}(\sigma) = D_{K,N+1}(\sigma)$ and $E_{K,N}(\sigma) = E_{K,N+1}(\sigma)$. Then, we have $D_{K,N}(\sigma) = D_{K,N+l}(\sigma)$ and $E_{K,N}(\sigma) = E_{K,N+l}(\sigma)$ for any $l \geq 0$, which is a contradiction to the first statement of the paragraph. ■

VI. Conclusion

This article settles an open problem of inference-based decentralized control by showing that, in general, there does not exist an upper bound on the number of levels of inferencing required to arrive at a correct control decision as the iterative computation used to arrive at the control decision need not terminate in general. From a design perspective, this suggests that such a number N should be decided as a design parameter based on the available computing resource, and a sensor selection algorithm should be developed for ensuring N-inference-observability (e.g., by extending the work in [1]) or the specification should be relaxed/constrained to a computed N-inference-observable super/sublanguage as in [9] and [10]. The nonexistence of an upper bound to inference-observability suggests that in general there may not exist a finite-time procedure to check for inference-observability (like the property of joint observability [13] and the solvability of the distributed supervisor synthesis problem [5], which are known to be undecidable). Identifying a special case where verifying inference-observability is decidable is a topic of future research.

References

[1] S. Jiang, R. Kumar, and H. E. Garcia, “Optimal sensor selection for discrete-event systems with partial observation,” *IEEE Trans. Autom. Control*, vol. 48, no. 3, pp. 369–381, Mar. 2003.

[2] A. Khoumsi and H. Chakib, “Decentralized supervisory control of discrete event systems: An arborescent architecture to realize inference-based control,” *IEEE Trans. Autom. Control*, vol. 63, no. 12, pp. 4278–4285, Dec. 2018.

[3] R. Kumar and S. Takai, “Inference-based ambiguity management in decentralized decision-making: Decentralized control of discrete event systems,” *IEEE Trans. Autom. Control*, vol. 52, no. 10, pp. 1783–1794, Oct. 2007.

[4] R. Kumar and S. Takai, “Inference-based ambiguity management in decentralized decision-making: Decentralized diagnosis of discrete-event systems,” *IEEE Trans. Autom. Sci. Eng.*, vol. 6, no. 3, pp. 479–491, Jul. 2009.
[5] L. Lin, A. Stefanescu, and R. Su, “On distributed and parameterized supervisor synthesis problems,” *IEEE Trans. Autom. Control*, vol. 61, no. 3, pp. 777–782, Mar. 2016.
[6] P. J. Ramadge and W. M. Wonham, “Supervisory control of a class of discrete event processes,” *SIAM J. Control Optim.*, vol. 25, no. 1, pp. 206–230, Jan. 1987.
[7] S. L. Ricker and K. Rudie, “Know means no: Incorporating knowledge into discrete-event control systems,” *IEEE Trans. Autom. Control*, vol. 45, no. 9, pp. 1656–1668, Sep. 2000.
[8] S. L. Ricker and K. Rudie, “Knowledge is a terrible thing to waste: Using inference in discrete-event control problems,” *IEEE Trans. Autom. Control*, vol. 52, no. 3, pp. 428–441, Mar. 2007.
[9] S. Takai and R. Kumar, “Synthesis of inference-based decentralized control for discrete event systems,” *IEEE Trans. Autom. Control*, vol. 53, no. 2, pp. 522–534, Mar. 2008.
[10] S. Takai and R. Kumar, “Synthesis of over-approximating inference-based decentralized supervisors for discrete event systems,” *IEEE Trans. Autom. Control*, vol. 55, no. 8, pp. 1881–1887, Aug. 2010.
[11] S. Takai and R. Kumar, “Inference-based decentralized prognosis in discrete event systems,” *IEEE Trans. Autom. Control*, vol. 56, no. 1, pp. 165–171, Jan. 2011.
[12] S. Takai and R. Kumar, “A generalized framework for inference-based diagnosis of discrete event systems capturing both disjunctive and conjunctive decision-making,” *IEEE Trans. Autom. Control*, vol. 62, no. 6, pp. 2778–2793, Jun. 2017.
[13] S. Tripakis, “Undecidable problems of decentralized observation and control on regular languages,” *Inform. Process. Lett.*, vol. 90, no. 1, pp. 21–28, Apr. 2004.
[14] Y. Wang, T.-S. Yoo, and S. Lafortune, “Diagnosis of discrete event systems using decentralized architectures,” *Discrete Event Dyn. Syst.: Theory Appl.*, vol. 17, no. 2, pp. 233–263, Jun. 2007.
[15] T.-S. Yoo and S. Lafortune, “A general architecture for decentralized supervisory control of discrete-event systems,” *Discrete Event Dyn. Syst.: Theory Appl.*, vol. 12, no. 3, pp. 335–377, Jul. 2002.
[16] T.-S. Yoo and S. Lafortune, “Decentralized supervisory control with conditional decisions: Supervisor existence,” *IEEE Trans. Autom. Control*, vol. 49, no. 11, pp. 1886–1904, Nov. 2004.