Automatica 133 (2021) 109838

Contents lists available at ScienceDirect

Automatica

journal homepage: www.elsevier.com/locate/automatica

Brief paper

Verification and enforcement of strong infinite- and k-step opacity using state recognizers[☆]

Ziyue Ma[a,*], Xiang Yin[b], Zhiwu Li[a,c]

[a] School of Electro-Mechanical Engineering, Xidian University, Xi'an 710071, China
[b] Department of Automation and the Key Laboratory of System Control and Information Processing, Shanghai Jiao Tong University, Shanghai 201108, China
[c] Institute of Systems Engineering, Macau University of Science and Technology, Taipa, Macau, China

ARTICLE INFO

Article history:
Received 3 May 2020
Received in revised form 2 June 2021
Accepted 14 June 2021
Available online xxxx

Keywords:
Discrete event system
Infinite-step opacity
k-step opacity
State recognizer

ABSTRACT

In this paper, we study the verification and enforcement problems of *strong infinite-step opacity* and *k-step opacity* for partially observed discrete-event systems modeled by finite state automata. Strong infinite-step opacity is a property such that the visit of a secret state cannot be inferred by an intruder at any instance along the entire observation trajectory, while strong k-step opacity is a property such that the visit of a secret state cannot be inferred within k steps after the visit. We propose two information structures called an ∞-step recognizer and a k-step recognizer to verify these two properties. The complexities of our algorithms to verify strong infinite- and k-step opacity are $O(2^{2\cdot|X|}\cdot|E_o|)$ and $O(2^{(k+2)\cdot|X|}\cdot|E_o|)$, respectively, which are lower than that of existing methods in the literature ($|X|$ and $|E_o|$ are the numbers of states and observable events in a plant, respectively). We also derive an upper bound for the value of k in strong k-step opacity, and propose an effective algorithm to determine the maximal value of k for a given plant. Finally, we note that enforcement of strong infinite- and k-step opacity can be transformed into a language specification enforcement problem and hence be solved using supervisory control.

1. Introduction

Opacity (Lafortune, Lin, & Hadjicostis, 2018) is a property that characterizes the situation whether some key information of a system can be inferred by an external intruder or not. In the last decades, abundant results have been done on the issue of opacity in discrete event dynamic models. In automata models, various notions of opacity have been proposed and analyzed, including *current state opacity*, *initial-state opacity* (Saboori & Hadjicostis, 2008), *language-based opacity* (Lin, 2011), *infinite-step opacity* and *k-step opacity* (Falcone & Marchand, 2015; Saboori & Hadjicostis, 2011, 2012). These works are later extended to coordinated architectures (Wu & Lafortune, 2013) and modular systems (Tong & Lan, 2019). Opacity problems are also studied in the framework of Petri nets (Bryans, Koutny, Mazaré, & Ryan, 2008) by checking language containment/equivalence. Recently, the work in Tong, Li, Seatzu, and Giua (2017) studies the verification of current-state and the initial-state opacity using *basis reachability graphs* (Ma, Tong, Li, & Giua, 2017) that improve the computational efficiency. Moreover, for a non-opaque plant, additional measures must be made to guarantee that the secret is not leaked, which is called *opacity enforcement*. These methods include *supervisory control* (Lan, Tong, & Seatzu, 2020; Tong, Li, Seatzu, & Giua, 2018) and *event edition* (Barcelos & Basilio, 2018; Behinaein, Lin, & Rudie, 2019; Ji, Wu, & Lafortune, 2018; Ji, Yin, & Lafortune, 2019; Ji, Yin, & Lafortune, 2019; Mohajerani, Ji, & Lafortune, 2019; Wu & Lafortune, 2014; Yin & Li, 2020).

Among various notions of opacity, infinite-step opacity and k-step opacity have drawn much attention in recent years. Infinite-step opacity is a property such that the visit of a secret state cannot be inferred by an intruder at any instance along the entire observation trajectory. Similarly, k-step opacity is a property such that the visit of a secret state cannot be inferred within k steps after leaving the secret state. In Saboori and Hadjicostis (2011, 2012), the notions of weak infinite- and k-step opacity are first proposed, and two algorithms based on *k-delay state estimators* are developed to verify them. The complexity of verifying weak infinite- and k-step opacity using delay state estimators is $O(2^{|X|}\cdot 2^{|X|^2}\cdot|E_o|)$, where $|X|$ and $|E_o|$ are the numbers of states and observable events in a plant, respectively. In Yin and Lafortune (2017), a more efficient method to verify weak infinite- and k-step opacity using a *two-way observer* is developed

[☆] The material in this paper was not presented at any conference. This paper was recommended for publication in revised form by Associate Editor Christoforos Hadjicostis under the direction of Editor Christos G. Cassandras.

[*] Corresponding author.

E-mail addresses: maziyue@xidian.edu.cn (Z. Ma), yinxiang@sjtu.edu.cn (X. Yin), zhwli@xidian.edu.cn (Z. Li).

https://doi.org/10.1016/j.automatica.2021.109838
0005-1098/© 2021 Elsevier Ltd. All rights reserved.

whose structural complexity is $O(2^{2|X|} \cdot |E_o|)$. Since the weak version of opacity has some limitations in practice, in Falcone and Marchand (2015) a new notion of opacity called *strong k-step opacity* is introduced. An algorithm is also proposed to verify strong k-step opacity in Falcone and Marchand (2015) using *k-delay trajectory estimators* and *R-verifiers* whose complexity is $O(2^{|X|} \cdot 2^{|X|^2} \cdot |E_o|)$.

In this paper, we are interested in the verification problem for *strong infinite-step opacity* (∞-SSO) and *strong k-step opacity* (k-SSO) in partially observed finite state automata.[1] These two notions of opacity are first proposed in Falcone and Marchand (2015) and Saboori and Hadjicostis (2011). However, in the literature there is no method to verify ∞-SSO as far as we know. On the other hand, for k-SSO, the work in Falcone and Marchand (2015) proposes a method to determine k-SSO of a plant for a given value of k. However, such a method cannot determine the maximal value of k for a given plant. Besides, other contributions of this paper are summarized as follows.

- For ∞-SSO, we propose a novel information structure called an ∞-*Step Recognizer* in which the information of passing the secret states is encoded. This approach results in a new algorithm that has complexity of $O(2^{2\cdot|X|} \cdot |E_o|)$.
- We show that the ∞-step recognizer is not suitable for the verification of k-SSO. Hence, we propose another information structure called a *k-Step Recognizer* and develop an algorithm to verify k-SSO of a plant for a certain value of k. This algorithm has complexity of $O(2^{(k+2)\cdot|X|} \cdot |E_o|)$. The previous algorithm reported in Falcone and Marchand (2015) for verifying k-SSO has complexity of $O(2^{|X|} \cdot 2^{|X|^2} \cdot |E_o|)$. Therefore, our new algorithm leads to considerable improvements in verification complexity when the size of a plant is relatively large.
- We propose an upper bound for the value of k in k-SSO. Precisely speaking, we prove that a system is ∞-SSO if and only if it is $(2^{2|X|})$-SSO. Then we propose an iterative algorithm to determine the maximal value of k for a given plant, whose complexity is $O(2^{(k+2)\cdot|X|} \cdot |E_o|)$.
- Finally, we develop an algorithm that enforces strong infinite- and k-step opacity by supervisory control.

This paper is organized in seven sections. Basic notions of automata, partial observation, and notions of opacity are recalled in Sections 2 and 3. In Section 4, the notion of ∞-step recognizer is proposed, based on which a new approach for the verification of ∞-SSO is developed. In Section 5, the notion of k-step recognizer is proposed, as well as an iterative algorithm to determine the maximal value of k for a given plant. In Section 6, the enforcement of ∞-SSO and k-SSO are studied. Section 7 draws the conclusion.

2. Preliminaries

We consider infinite-step and k-step opacity problems in a deterministic *finite state automaton* (*automaton* for short)

$$G = (X, E, \delta, x_0),$$

where X is a set of *states*; E is a set of *events*; $\delta : X \times E \to X$ is the partial transition function; and $x_0 \in X$ is the initial state. We use E^* to denote the *Kleene closure* of E, consisting of all finite sequences composed of the events in E and the *empty sequence* ε. The *language* of G, denoted by $L(G)$, is defined as $L(G) = \{s \in E^* \mid \delta(x_0, s) \in X\}$. Given a sequence $s \in E^*$, $|s|$ denotes the *length* of s. A sequence $\bar{s} \in E^*$ is a *prefix* of a sequence $s \in E^*$ if $s = \bar{s}s'$ where $s' \in E^*$, which is denoted by $\bar{s} \preceq s$. The *prefix closure* of $s \in E^*$ is defined as $Pr(s) = \{\bar{s} \in E^* \mid \bar{s} \preceq s\}$.

A plant G is partially observable in general. Hence, the event set E is partitioned into the *set of observable events* E_o and the *set of unobservable events* E_{uo}. Given a sequence $s \in E^*$, its observable projection $\sigma = P(s)$ is the output of the natural projection $P : E^* \to E_o^*$ recursively defined as (1) $P(\varepsilon) = \varepsilon$, (2) $P(e) = e$ if $e \in E_o$ and $P(e) = \varepsilon$ if $e \in E_{uo}$, (3) $P(se) = P(s)P(e)$, where $s \in E^*$ and $e \in E$. The inverse projection $P^{-1} : E_o^* \to 2^{E^*}$ is defined as:$P^{-1}(\sigma) = \{s \in L(G) \mid P(s) = \sigma\}$, i.e., $P^{-1}(\sigma)$ consists of all sequences s in $L(G)$ whose observations are σ. The *observed language* of G, denoted by $P[L(G)]$, is defined as $P[L(G)] = \{\sigma \in E_o^* \mid (\exists s \in L(G))\ \sigma = P(s)\}$.

3. Strong infinite-step and *k*-step opacity

Given a plant $G = (X, E, \delta, x_0)$, part of its state set is the set of *secret states* that is denoted by X_S with $X_S \subset X$. There is an intruder who knows the structure of G but only observes the observable events (i.e., the events in E_o) generated by G. Hence, for each observation $s \in E_o^*$, the intruder tries to infer if G has visited $x \in X_S$ some time before, by using the knowledge of the structure of G and observation s. This motivates the notion of strong opacity that will be studied in this work.

Definition 3.1 (*Strong k-Step Opacity Falcone & Marchand, 2015*)**.** Given a plant $G = (X, E, \delta, x_0)$, a set of observable events E_o, and a set of secret states X_S, G is *strongly k-step opaque* (k-SSO) with respect to X_S (where $k \in \mathbb{N}$) if for all sequences $st \in L(G)$ such that $\delta(x_0, s) \in X_S$ and $|P(t)| \leq k$, there exists a sequence $w \in L(G)$ such that:

$$(\forall \bar{w} \preceq w, |P(w)| - |P(\bar{w})| \leq k)\ \delta(x_0, \bar{w}) \notin X_S \\ \wedge P(w) = P(st). \tag{1}$$

The physical interpretation of k-SSO is that for any string that leads to a secret state, an intruder cannot necessarily determine that the system is/was in a secret state at that point using up to k observations thereafter. According to its definition, k-SSO reduces to the current-state opacity (Saboori & Hadjicostis, 2008) when $k = 0$. On the other hand, by letting $k \to \infty$ such that conditions "$|P(t)| \leq k$" and "$|P(w)| - |P(\bar{w})| \leq k$" in Eq. (1) are trivially satisfied, the notion of k-SSO is extended to *strong infinite-step opacity* (∞-SSO) below. Such a notion of opacity is first proposed in this work.

Definition 3.2 (*Strong Infinite-Step Opacity*)**.** Given a plant $G = (X, E, \delta, x_0)$, a set of observable events E_o, and a set of secret states X_S, G is *strongly infinite-step opaque* (∞-SSO) with respect to X_S if for all sequence $st \in L(G)$ such that $\delta(x_0, s) \in X_S$, there exists a sequence $w \in L(G)$ such that:

$$(\forall \bar{w} \preceq w)\ \delta(x_0, \bar{w}) \notin X_S \ \wedge\ P(w) = P(st). \tag{2}$$

In plain words, ∞-SSO requires that for any secret sequence that passes a secret state, there should exist another sequence that looks like the former and does not pass any secret state, which implies that an intruder can never determine that the system is/was in a secret state using further observations.

Proposition 3.1. *The following statements are true:*

1. *the fact that G is k-SSO with respect to X_S implies that G is k'-SSO with respect to X_S for any $0 \leq k' < k$;*
2. *the fact that G is ∞-SSO with respect to X_S implies that G is k-SSO with respect to X_S for any $k \in \mathbb{N}$.*

Proof. By Eq. (1), if G is not k'-SSO, then for any $k > k'$, G is not k'-SSO. By contrapositive, statement 1 is true. Since ∞-SSO is k-SSO for $k \to \infty$, statement 2 is also true. □

[1] To simplify the presentation, the term "∞-SSO" (resp. k-SSO) is used as the abbreviation of both "strong infinite-step opacity" (resp. "strong k-step opacity") and "strongly infinite-step opaque" (resp. "strongly k-step opaque"), which depends on the context.

It is worth noting that strong opacity (Definitions 3.1 and 3.2) and weak opacity (see Saboori & Hadjicostis, 2011, 2012; Yin & Lafortune, 2017) are incomparable, i.e., ∞- or k-step strong opacity does not imply ∞- or k-step weak opacity, and vice versa. Hence, the approaches for the verification of one type of opacity verification cannot be applied to the verification of the other.

4. Verification of strong infinite-step opacity using infinite-step recognizers

In this section we focus on the verification problem of ∞-SSO in Definition 3.1. We first define an X_S-*secret language* and then define an (X_S,∞)-*estimation* that characterize the knowledge of the intruder.

Definition 4.1. Given a plant $G=(X,E,\delta,x_0)$ and a set of secret states X_S, we define the X_S-*secret language* of state x as:

$$L_S(G,x)=\{s\in E^* \mid [\delta(x,s)\in X]\wedge \\ [(\exists \bar{s}\preceq s)\ \delta(x,\bar{s})\in X_S]\}. \tag{3}$$

The ∞-*step test function* $S_\infty(x,s): X\times E^*\rightarrow\{0,\infty\}$ is defined as:

$$S_\infty(x,s)=\begin{cases}0, & \text{if } s\notin L_S(G,x)\\ \infty, & \text{if } s\in L_S(G,x)\end{cases}$$

Remark 1. We denote the codomain of function S_∞ as $\{0,\infty\}$ (instead of $\{0,1\}$) to avoid possible confusions for later developments. In Section 5, the flag set in k-SSO is denoted as $\{0,1,\ldots,k+1\}$. The interpretation of the "∞" flag in an ∞-SR is completely different with that of the "1" flag in a k-SR.

Definition 4.2. Given a plant $G=(X,E,\delta,x_0)$, a set of observable events E_o, and a set of secret states X_S, for an observation $\sigma\in P[L(G)]$, the (X_S,∞)-*estimation* of σ is defined as:

$$\mathcal{E}(\sigma,X_S,\infty)=\{(x,\gamma)\mid(\exists s\in P^{-1}(\sigma)) \\ \delta(x_0,s)=x,\gamma=S_\infty(x_0,s)\}. \tag{4}$$

Given an observation $\sigma\in P[L(G)]$, the (X_S,∞)-estimation of σ (i.e., $\mathcal{E}(\sigma,X_S,\infty)$) is a compact representation that characterizes the knowledge of an intruder who observes σ. Precisely speaking, each pair (x,γ) in $\mathcal{E}(\sigma,X_S,\infty)$ represents a trajectory from initial state x_0 such that (i) the plant is currently at state x by executing a sequence $s\in E^*$ that looks like σ, and (ii) during the execution of s from x_0 to x, some secret state has been reached/passed (if $\gamma=\infty$) or no secret state has been reached/passed (if $\gamma=0$). In other words, the second component γ in (x,γ) is a *flag* that denotes if a secret has passed ($\gamma=\infty$) or not ($\gamma=0$).

The (X_S,∞)-estimation of empty observation ε, i.e., $\mathcal{E}(\varepsilon,X_S,\infty)$, can be computed according to Definition 4.2. On the other hand, if $\mathcal{E}(\sigma,X_S,\infty)$ is known for some $\sigma\in P[L(G)]$, then for any event e such that $\sigma e\in P[L(G)]$, a set $\mathcal{E}(\sigma e,X_S,\infty)$ can be computed from set $\mathcal{E}(\sigma,X_S,\infty)$. Now we introduce some useful notations before showing how it can be done. The *unobservable reach* of a pair (x,γ), where $x\in X$ and $\gamma\in\{0,1\}$, is defined as:

$$UR(x,\gamma)=\{(x',\gamma')\mid(\exists s\in E_{uo}^*) \\ \delta(x,s)=x',\gamma'=\max\{\gamma,S_\infty(x,s)\}\}.$$

We also denote by $Next((x,\gamma),e)$ the set of pairs that can be reached immediately upon the occurrence of observable event $e\in E_o$, i.e.,

$$Next((x,\gamma),e)=\{(x',\gamma')\mid\delta(x,e)=x', \\ \gamma'=\max\{\gamma,S_\infty(x,e)\}\}.$$

Proposition 4.1. *Given a plant $G=(X,E,\delta,x_0)$, a set of observable events E_o, and a set of secret states X_S, for an observation $\sigma\in P[L(G)]$ and event $e\in E_o$ such that $\sigma,\sigma e\in P[L(G)]$, it holds:*

$$\mathcal{E}(\sigma e,X_S,\infty)=\bigcup_{(x,\gamma)\in\mathcal{E}(\sigma,X_S,\infty)} UR(Next((x,\gamma),e))$$

Proof. This result is directly from the definition of functions *Next* and *UR*. □

By the definition of $\mathcal{E}(\sigma,X_S,\infty)$, it is not difficult to understand that an intruder can infer that a plant necessarily has reached/passed some secret state by observing $\sigma\in P[L(G)]$ if and only if the flags of all pairs (x,γ) in his/her estimation $\mathcal{E}(\sigma,X_S,\infty)$ are ∞. Now we are ready to introduce an information structure called an ∞-*step recognizer* (∞-SR) in which all $\mathcal{E}(\sigma,X_S,\infty)$ of all $\sigma\in P[L(G)]$ are encoded. In plain words, each state of an ∞-SR is a macro-state d that consists of one or more pairs:

$$d=\{(x_{i_1},\gamma_1),(x_{i_2},\gamma_2),\ldots,(x_{i_n},\gamma_n)\},$$

where each $x_{i_j}\in X$ is a plant state and $\gamma_j\in\{0,\infty\}$ is a *flag* that records if a secret state has been reached ($\gamma=\infty$) or not ($\gamma=0$). The formal definition of the ∞-SR is the following.

Definition 4.3. Given a plant $G=(X,E,\delta,x_0)$, a set of observable events E_o, and a set of secret states X_S, the ∞-*step recognizer* (∞-SR) of G is a deterministic automaton $G_D=(D,E_o,\delta_\infty,d_0)$ such that: (i) the state set is $D\subseteq 2^{X\times\{0,\infty\}}$; (ii) the event set is E_o; (iii) the transition function $\delta_\infty: D\times E_o\rightarrow D$ is recursively defined as:

$$\delta_\infty(d,e)=\bigcup_{(x,\gamma)\in d} UR(Next((x,\gamma),e));$$

(iv) the initial state is $d_0=UR(x_0,0)$.

Proposition 4.2. *Given a plant $G=(X,E,\delta,x_0)$, a set of observable events E_o, and a set of secret states X_S, let $G_D=(D,E_o,\delta_\infty,d_0)$ be the ∞-SR of G. It holds:*

$$\sigma\in P[L(G)] \quad\Rightarrow\quad \mathcal{E}(\sigma,X_S,\infty)=\delta_\infty(d_0,\sigma).$$

Proof. Directly from the definition of ∞-SR and Proposition 4.1. □

Given a plant G and its ∞-SR G_D, we classify the states in G_D into two types: (i) **leaking states**: for all $(x_{i_j},\gamma_j)\in d$, $\gamma_j=\infty$ holds; (ii) **non-leaking states**: there exists at least one $(x_{i_j},\gamma_j)\in d$ with $\gamma_j=0$. Now we are ready to introduce the first main result of this work that is: a plant is ∞-SSO if and only if its corresponding ∞-SR does not contain any leaking state.

Theorem 4.1. *Given a plant $G=(X,E,\delta,x_0)$, a set of observable events E_o, and a set of secret states X_S, let $G_D=(D,E_o,\delta_\infty,d_0)$ be the corresponding ∞-SR. Plant G is ∞-SSO with respect to X_S if and only if there is no leaking state in its ∞-SR.*

Proof. ($\Rightarrow$) By contrapositive, suppose that there exists a leaking state d in G_D. Thus, there exists an observation $\sigma\in P[L(G)]$ such that $\delta_\infty(d_0,\sigma)=d$. By Proposition 4.2, $\mathcal{E}(\sigma,X_S,\infty)=d$ holds. Since all pairs $(x,\gamma)\in\mathcal{E}(\sigma,X_S,\infty)$ satisfy $\gamma=\infty$, the execution of any sequence $s\in P^{-1}(\sigma)$ from x_0 necessarily passes some secret state in X_S. Let s_0 be an arbitrary sequence in $P^{-1}(\sigma)$ that passes X_S. Sequence s_0 can be written as $s_0=st$ such that $\delta(x_0,s)\in X_S$. Then, since all sequences in $P^{-1}(\sigma)$ necessarily pass a secret states, we can conclude that for all sequences $w\in L(G)$ such that $P(w)=P(st)$, none of them satisfies $\delta(x_0,\bar{w})\notin X_S$ for all $\bar{w}\preceq w$. Hence, by Definition 3.1, G is not ∞-SSO with respect to X_S.

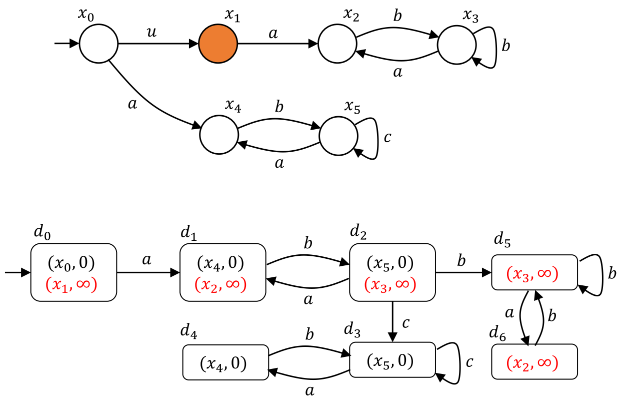

Fig. 1. The automaton for in Example 4.1.

($\Leftarrow$) By contrapositive, suppose that G is not ∞-SSO. By Definition 3.1, there necessarily exists a sequence st such that (1) $\delta(x_0, s) \in X_S$, and (2) there does not exist a sequence $w \in L(G)$ such that $P(w) = P(st)$ and $\delta(x_0, \bar{w}) \notin X_S$ for all $\bar{w} \preceq w$. The second condition implies that all sequences that look like st necessarily pass X_S, i.e., all pairs $(x, \gamma) \in \mathcal{E}(\sigma, X_S, \infty)$ have $\gamma = \infty$. By the construction of G_D and Proposition 4.2, all pairs (x, γ) in state $d = \delta_\infty(d_0, P(s)) = \mathcal{E}(\sigma, X_S, \infty)$ have $\gamma = \infty$. Hence, d is a leaking state in G_D. □

Example 4.1. Consider the automaton G in Fig. 1 in which $E_o = \{a, b, c\}$, $E_{uo} = \{u\}$, and the secret set is $X_S = \{x_1\}$. The corresponding ∞-SR G_D is depicted in the same figure. In G_D there exist two leaking states $d_5 = \{(x_3, \infty)\}$ and $d_6 = \{(x_2, \infty)\}$. By Theorem 4.1, G is not ∞-SSO with respect to X_S. In fact, if the plant executes sequence $s = uabb$ whose observation is $P(s) = abb$, the intruder can infer that secret state x_1 is necessarily passed.

Since the state set D in an ∞-SR satisfies $D \subseteq 2^{X \times \{0, \infty\}}$, the structural complexity of G_D is $O(2^{2|X|} \cdot |E_o|)$. Hence, our approach has the same complexity of the verification of the weak ∞-step opacity using two-way observers (Yin & Lafortune, 2017).

5. Verification of strong k-step opacity using k-step recognizers

Since for ∞-SSO the passage of secret states is leaked when an ∞-SR reaches a leaking state, one may intuitively conjecture that the property of k-SSO can also be determined by inspecting the length of ambiguous paths (i.e., a path on which all states have both 0 and ∞ flags) in the ∞-SR. Unfortunately, such a conjecture is false, since the ambiguous path in ∞-SR does not contain essential information of trajectories of the plant states on the path. Hence, in this section we first solve the verification problem of k-SSO in Definition 3.2 using a different information structure called k-SR.

5.1. Verification of k-SSO for a given k

We introduce some useful notions that will be used to design an information structure called a *k-step recognizer*. We propose (X_S, k)-estimation that characterizes the knowledge of an intruder if a secret state has reached/passed during the last k observations in σ.

Definition 5.1. Given a plant $G = (X, E, \delta, x_0)$, a set of observable events E_o, and a set of secret states X_S, for a sequence $s \in L_S(G, x)$, the *maximal non-secret suffix index* of s is defined as

$$N(x, s) = |P(s)| - \max\{i \mid (\bar{s} \preceq s, |P(\bar{s})| = i) \; \delta(x, \bar{s}) \in X_S\}.$$

In particular, we define $N(x, s) = +\infty$ for $s \in L(G, x) \setminus L_S(G, x)$. The *k-step test function* $S_k : X \times E^* \to \{0, 1, \ldots, k\}$ is defined as:

$$S_k(x, s) = \max\{k - N(x, s) + 1, 0\}.$$

Definition 5.2. Given a plant $G = (X, E, \delta, x_0)$, a set of observable events E_o, a set of secret states X_S, and an integer $k \in \mathbb{N}$, for an observation $\sigma \in P[L(G)]$, the (X_S, k)*-estimation* of σ is defined as:

$$\mathcal{E}(\sigma, X_S, k) = \{(x, \gamma) \mid (\exists s \in P^{-1}(\sigma)) \; \delta(x_0, s) = x, \gamma = S_k(x_0, s)\}. \tag{5}$$

An (X_S, k)-estimation is a compact representation that characterizes the knowledge of an intruder who memorizes the visit of secret states for at most k steps afterwards. In a pair $(x, \gamma) \in \mathcal{E}(\sigma, X_S, k)$, if state x is a secret state, the value of flag γ is set to $k+1$. Moreover, such a flag has a lifespan $k+1$: for each observed event thereafter, the value of γ is decreased by 1 until it reaches zero. Hence, after observing $k+1$ events thereafter, if during this period the trajectory does not pass another secret state, γ is reduced to zero, i.e., the intruder "forgets" that the plant has visited secret state x.

Similar to the result in Section 3, for an observation $\sigma \in P[L(G)]$, set $\mathcal{E}(\sigma, X_S, k)$ can be obtained by iteratively computing the prefix of σ, starting from $\mathcal{E}(\varepsilon, X_S, k)$ (which can be computed according to its definition). We define the *k-unobservable reach* of a pair (x, γ), where $x \in X$ and $\gamma \in \{0, 1, \ldots, k\}$, as:

$$UR_k(x, \gamma) = \{(x', \gamma') \mid (\exists s \in E_{uo}^*) \; \delta(x, s) = x', \gamma' = \max\{\gamma, S_k(x, s)\}\}.$$

We denote by $Next_k((x, \gamma), e)$ the set of states that can be reached immediately upon the occurrence of observable event $e \in E_o$, i.e.,

$$Next_k((x, \gamma), e) = \{(x', \gamma') \mid \delta(x, e) = x', \gamma' = \max\{\gamma, S_k(x, e)\}\}.$$

Proposition 5.1. *Given a plant $G = (X, E, \delta, x_0)$, a set of observable events E_o, a set of secret states X_S, and an integer $k \in \mathbb{N}$, for an observation $\sigma \in P[L(G)]$ and event $e \in E_o$ such that $\sigma, \sigma e \in P[L(G)]$, it holds:*

$$\mathcal{E}(\sigma e, X_S, k) = \bigcup_{(x, \gamma) \in \mathcal{E}(\sigma, X_S, k)} UR_k(Next_k((x, \gamma), e))$$

Proof. This result directly follows from the definition of $Next_k$ and UR_k functions. □

By the definition of (X_S, ∞)-estimation, it is not difficult to understand that an intruder can infer that the plant necessarily has reached/passed some secret state in the last k steps by observing $\sigma \in P[L(G)]$ if and only if the flags of all pairs (x, γ) in his/her estimation $\mathcal{E}(\sigma, X_S, k)$ are nonzero. Now we introduce an information structure called a *k-Step Recognizer* (k-SR) in which all $\mathcal{E}(w, X_S, k)$ of all $\sigma \in P[L(G)]$ are encoded.

Definition 5.3. Given a plant $G = (X, E, \delta, x_0)$, a set of observable events E_o, and a set of secret states X_S, the *k-step recognizer* (k-SR) of G is a deterministic automaton $G_{D,k} = (D_k, E_o, \delta_k, d_0)$ such that: (i) the state set is $D_k \subseteq 2^{X \times \{0, 1, \ldots, k\}}$; (ii) the event set is E_o;

(iii) the transition function $\delta_k : D_k \times E_o \to D_k$ is recursively defined as:

$$\delta_k(d, e) = \bigcup_{(x,\gamma)\in d} UR_k(Next_k((x, \gamma), e));$$

(iv) the initial state is $d_0 = UR_k(x_0, 0)$.

Proposition 5.2. *Given a plant $G = (X, E, \delta, x_0)$, a set of observable events E_o, and a set of secret states X_S, let $G_{D,k} = (D, E_o, \delta_k, d_0)$ be the k-SR of G. For any sequence $\sigma \in P[L(G)]$, it holds:*

$$\mathcal{E}(\sigma, X_S, k) = \delta_k(d_0, \sigma).$$

Proof. Directly from the definition of k-SR and Proposition 4.1. □

Given a plant G and its k-SR $G_{D,k}$, we classify the states in $G_{D,k}$ into two types: (i) **leaking states**: for all $(x_{i_j}, \gamma_j) \in d$, $\gamma_j > 0$ holds; (ii) **non-leaking states**: there exists at least one $(x_{i_j}, \gamma_j) \in d$ with $\gamma_j = 0$. Similar to the result in Section 4, the following theorem indicates that a plant is k-SSO if and only if its corresponding k-SR does not contain any leaking state.

Theorem 5.1. *Given a plant $G = (X, E, \delta, x_0)$, a set of observable events E_o, and a set of secret states X_S, let $G_{D,k} = (D, E_o, \delta_k, d_0)$ be the corresponding k-SR. Plant G is k-SSO with respect to X_S if and only if there is no leaking state in $G_{D,k}$.*

Proof. ($\Rightarrow$) By contrapositive, suppose that there exists a leaking secret state d in G_D, i.e., there exists an observation $\sigma \in P[L(G)]$ such that $\delta_\infty(d_0, \sigma) = d$. By Proposition 5.1, $\mathcal{E}(\sigma, X_S, k) = d$ holds. Since all pairs $(x, \gamma) \in \mathcal{E}(\sigma, X_S, k)$ satisfy $\gamma \geq 1$, the execution of any sequence $s \in P^{-1}(\sigma)$ from x_0 necessarily passes some secret state in X_S in the last k observed events. Let s_0 be an arbitrary sequence in $P^{-1}(\sigma)$. Sequence s_0 can be written as $s_0 = st$ such that $\delta(x_0, s) \in X_S$ and $|t| \leq k$. Then we can conclude that for all sequences $s' \in L(G)$ such that $P(s') = P(st)$, none of them satisfies $\delta(x_0, \bar{s}') \notin X_S$ for all $\bar{s}' \preceq s'$, $|P(s')| - |P(\bar{s}')| \leq k$. Hence, G is not k-SSO with respect to X_S.

($\Leftarrow$) By contrapositive, suppose that G is not k-SSO. By Definition 3.2, there necessarily exists a sequence st such that (1) $\delta(x_0, s) \in X_S$, and (2) there does not exist a sequence $w \in L(G)$ such that $P(w) = P(st)$ and $\delta(x_0, \bar{w}) \notin X_S$ for all $\bar{w} \preceq w$, $|P(w)| - |P(\bar{w})| \leq k$. The second condition implies that all sequences w that look like st must have passed X_S during the last k observed events in w. By the construction of $G_{D,k}$ and Proposition 5.1, all pairs (x, γ) in state $d = \delta(d_0, P(s))$ have $\gamma \geq 1$. Hence, d is a leaking state in $G_{D,k}$. □

Example 5.1. Consider the automaton G in Fig. 2 in which $E_o = \{a, b, c\}$ and $E_{uo} = \{u\}$. Suppose that we want to verify if G is 1-SSO with respect to $X_S = \{x_5, x_7\}$. The corresponding 1-SR $G_{D,1}$ is depicted in the same figure. Since in $G_{D,1}$ there is no leaking state, by Theorem 5.1, G is 1-SSO with respect to X_S, i.e., an intruder can never infer that G was at a secret state one step before.

By the state set $D_k \subseteq 2^{X\times\{0,1,\ldots,k+1\}}$, for a given $k \in \mathbb{N}$, the structural complexity of $G_{D,k}$ is $O(2^{|X|} \cdot 2^{k+2} \cdot |E_o|)$. In comparison, the existing algorithm proposed in Falcone and Marchand (2015) to verify k-SSO has complexity $O(2^{|X|} \cdot 2^{|X|^2} \cdot |E_o|)$. Hence, the complexity of our k-SR-based verification algorithm is smaller than that in Falcone and Marchand (2015) when the size of the plant is large (i.e., $|X|^2 \gg k + 2$).

5.2. Determining the maximal k for k-SSO

In this subsection, we propose an iterative algorithm to determine the maximal value of k such that G is k-SSO. The integrated algorithm is sketched in Algorithm 1. First, we check if G is ∞-SSO using ∞-SR. If G is ∞-SSO then it is k-SSO for all $k \in \mathbb{N}$, and hence Algorithm 1 outputs "∞". Otherwise, we sequentially verify 0-SO, 1-SO, … until encounter a $\bar{k}$ such that the plant G is not $\bar{k}$-SSO. If $\bar{k} \geq 1$, Algorithm 1 outputs "$\bar{k} - 1$", which means that the maximal value of k such that G is k-SSO with respect to X_S is $k = \bar{k} - 1$. On the other hand, if $\bar{k} = 0$, then Algorithm 1 outputs "*NA*", which means that k does not exist.

Algorithm 1 Determining maximal k for k-SSO

Input: A plant $G = (X, E, \delta, x_0)$, a set of observable events E_o, and a set of secret states X_S

```
compute $\infty$-SR $G_D$;
if in $G_D$ there does not exist any leaking state, then
    output "$\infty$" and exit;
end if
let $i = 0$;
while true, do
    compute $i$-SR $G_{D,k}$;
    if in $G_{D,i}$ there exists any leaking state, then
        if $k \neq 0$, then
            output "$i - 1$" and exit;
        else
            output "$NA$" and exit;
        end if
    end if
    let $i = i + 1$;
end while
```

Proposition 5.3. *Algorithm 1 is correct, i.e., it outputs the maximal value of k such that G is k-SSO with respect to X_S.*

Proof. First, if G is ∞-SSO, by Proposition 3.1, G is k-SSO for all $k \in \mathbb{N}$, i.e., the maximal value of k is ∞. Otherwise, there necessarily exists an integer $\hat{k}$ such that G is k-SSO for all $k < \hat{k}$ and not k-SSO for all $k \geq \hat{k}$. Then by testing the existence of leaking states in each $G_{D,i}$ with $i = 0, 1, \ldots$ sequentially we will eventually reach the condition $i = \hat{k}$. If $\hat{k} \geq 1$, G is $(\hat{k} - 1)$-SSO with X_S and is not k-SSO for any $k \geq \hat{k}$. On the other hand, $\hat{k} = 0$ indicates that G is not k-SSO for any $k \in \mathbb{N}$. □

Although Algorithm 1 seems enumerating all k's starting from $k = 0$ in a brute-force way, we point out that the gross computational load of Algorithm 1 falls into the same class of that of k-SR. In fact:

- if G is ∞-SSO (and hence is k-SSO for any $k > 0$), Algorithm 1 is of complexity $O(2^{2\cdot|X|} \cdot |E_o|)$ (since only ∞-SR is constructed);
- if G is not 0-SSO (and hence is not k-SSO for any $k > 0$), Algorithm 1 is of complexity $O([2^{2\cdot|X|} + 2^{|X|}] \cdot |E_o|)$ (since only ∞-SR and 0-SR are constructed);
- if G is k-SSO for some maximal $k > 0$, the complexity of Algorithm 1 is:

$$\begin{aligned} O(\text{Algorithm } 1) &= O([2^{|X|} + 2^{2\cdot|X|} + \cdots + 2^{(k+2)\cdot|X|}] \cdot |E_o|) \\ &= O(|E_o| \cdot \sum_{i=0}^{k} 2^{(|X|\cdot(i+2))}) \\ &= O(\frac{2^{(k+2)\cdot|X|} \cdot 2^{|X|} - 2^{|X|}}{2^{|X|} - 1}) \\ &\approx O(2^{(k+2)\cdot|X|} \cdot |E_o|) \quad (\text{assume } 2^{|X|} \gg 1). \end{aligned}$$

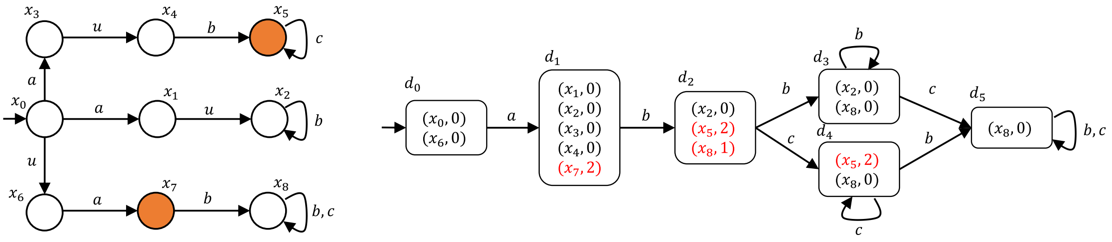

Fig. 2. (Left) The automaton in Example 5.1, and (right) the corresponding 1-SR with $X_S = \{x_5, x_7\}$, $E_o = \{a, b, c\}$, $E_{uo} = \{u\}$.

In other words, the computation on 0-SR, 1-SR, ..., $(k-1)$-SR are negligible comparing with the computation of k-SR.

Example 5.2. Again consider the automaton G in Fig. 2 in which $E_o = \{a, b, c\}$, $E_{uo} = \{u\}$, and secret $X_S = \{x_5, x_7\}$. By applying Algorithm 1, the corresponding ∞-SR G_D is depicted in Fig. 3(a). Since there exists a leaking state $d_4 = \{(x_5, \infty), (x_8, \infty)\}$ in G_D, by Theorem 4.1, G is not ∞-SSO with respect to X_S.

Since G is not ∞-SSO, we construct and examine i-SR for $i = 0, 1, \ldots$ sequentially. The structure of 0-SR, 1-SR, and 2-SR is shown in Fig. 3(b–d). In both 0-SR and 1-SR there is no leaking state, while in 2-SR there is a leaking state $d_6 = \{(x_5, 3), (x_8, 1)\}$. Hence, by Theorem 5.1, G is not 2-SSO with respect to X_S. In fact, for trajectory

$$x_0 \xrightarrow{a} x_3 \xrightarrow{u} x_4 \xrightarrow{b} x_5 \xrightarrow{c} x_5$$

whose observation is abc, the intruder knows that the plant must have been visited a secret state (x_5 in this case) one step before. In other words, the maximal k that makes G be k-SSO with respect to X_S is $k = 1$.

It is worth noting that the "while" loop in Algorithm 1 (Step 6) always terminates in finite iterations, although the complexity of Algorithm 1 has a factor k. Moreover, in the next subsection, we derive an upper bound for k by showing that a system is ∞-SSO if and only if it is $(|X| \cdot (2^{|X|} - 1))$-SSO. Hence, the "while" loop in Algorithm 1 will terminate in at most $|X| \cdot (2^{|X|} - 1)$ iterations.

5.3. An upper bound of k in k-SSO

In this subsection we show that for any $k' > k \geq |X| \cdot (2^{|X|} - 1)$, k-SSO and k'-SSO are equivalent, which indicates that $|X| \cdot (2^{|X|} - 1)$ is an upper bound of delay k for k-SSO.

Theorem 5.2. *Given a plant $G = (X, E, \delta, x_0)$, a set of observable events E_o, and a set of secret states X_S, for $k' > k \geq |X| \cdot (2^{|X|} - 1)$, plant G is k-SSO if and only if G is k'-SSO.*

Proof. By Proposition 3.1, G is k-SSO if G is k'-SSO. Hence, we only need to prove that G is k'-SSO if G is k-SSO. Without loss of generality, we assume $k' = k + 1$, since the proof can be straightforwardly generalized to $k' > k + 1$. In the following, we prove that if G is not n-SSO, then G is not $(n - 1)$-SSO by contrapositive, where $n \geq |X| \cdot (2^{|X|} - 1) + 1$. The roadmap is illustrated in Fig. 4.

Suppose that G is not n-SSO. There necessarily exists a sequence $st \in L(G)$ such that $|P(t)| = n$, $\sigma(x_0, s) \in X_S$ and for all sequences $w \in L(G)$ that satisfy $P(w) = P(st)$, there exists $\bar{w} \preceq w$ with $|w| - |\bar{w}| \leq n$ such that $\sigma(x_0, \bar{w}) \in X_S$. Now consider $G_{d,n} = (D, E_0, \delta_n, d_{0,n})$ that is the n-SR of G. Let sequence t be denoted as $t = e_1 e_2 \cdots e_n$ where $e_i \in E$ for $i = 1, \ldots, n$. In $G_{d,n}$ there necessarily exists the following trajectory

$$d_1 \xrightarrow{e_1} d_2 \xrightarrow{e_2} \cdots \xrightarrow{e_{n-1}} d_n \xrightarrow{e_n} d_{n+1} \tag{6}$$

where $d_1 = \delta_k(d_{0,k}, P(s))$ and d_{n+1} is a leaking state, i.e., all pairs (x, γ) in d_{n+1} are with $\gamma > 0$.

From Eq. (6) we can extract a sequence:

$$\begin{aligned}(x_1, \gamma_1) \xrightarrow{e_1} (x_2, \gamma_2) \xrightarrow{e_2} \cdots \xrightarrow{e_{n-1}} & (x_n, \gamma_n) \\ & \xrightarrow{e_n} (x_{n+1}, \gamma_{n+1}).\end{aligned} \tag{7}$$

where $\gamma_1 = n$ and $1 \leq \gamma_i \leq n$ for $i = 2, \ldots, n + 1$. On the other hand, from Eq. (6) we can extract another sequence:

$$d_{NS,1} \xrightarrow{e_1} d_{NS,2} \xrightarrow{e_2} \cdots \xrightarrow{e_{n-1}} d_{NS,n} \xrightarrow{e_n} d_{NS,n+1}. \tag{8}$$

where $d_{NS,i}$ denotes the set of pairs with zero flags in d_i, i.e., $d_{NS,i} = \{(x, \gamma) \in d_i \mid \gamma = 0\}$. Note that $d_{NS,n+1} = \emptyset$ since G is not n-SSO. Since the lengths of the sequences in Eqs. (7) and (8) are both $n + 1$ that is greater than $|X| \cdot (2^{|X|} - 1) + 1$, and G does not have any unobservable cycle, there necessarily exist two indices $j_1, j_2 \in \{1, \ldots, n\}, j_1 \neq j_2$, such that $x_{j_1} = x_{j_2}$ and $d_{NS,j_1} = d_{NS,j_2}$. As a result, sequence t can be partitioned as $t = uvz$, where $|u| = j_1 - 1$ and $|v| = j_2 - j_1 \geq 1$. By removing the parts associated with subsequence v from Eqs. (7) and (8):

- from Eq. (7), we have $(x_1, \gamma_1) \xrightarrow{uz} (x_{n+1}, \gamma')$. Since $\gamma_1 = n$ and $|P(uz)| \leq |P(uvz)| = |P(t)| = n$, $\gamma' \geq 1$ holds;
- from Eq. (8), we have $d_{NS,1} \xrightarrow{P(uz)} d'_{NS,n+1}$ where $d'_{NS,n+1} = \emptyset$.

Hence, for sequence $suz \in L(G)$, we have $|P(uz)| \leq n-1$, $\sigma(x_0, s) \in X_S$, and for all sequences $w \in L(G)$ that satisfy $P(w) = P(suz)$, there exists $\bar{w} \preceq w$ with $|w| - |\bar{w}| \leq n-1$ such that $\sigma(x_0, \bar{w}) \in X_S$. Therefore, G is not $(n - 1)$-SSO, which concludes the proof. □

Corollary 5.1. *Given a plant $G = (X, E, \delta, x_0)$, a set of observable events E_o, and a set of secret states X_S, G is ∞-SSO if and only if G is $(|X| \cdot (2^{|X|} - 1))$-SSO.*

Proof. Directly from Theorem 5.2 and Proposition 3.1. □

Remark 2. At the end of this section we make the following comments. First, one may have noticed that the propagation rule of flag "∞" in an ∞-SR is similar to the propagation rule of the fault flag in a *diagnoser automaton* (Cassandras & Lafortune, 2008). In fact, since in ∞-SSO an intruder never forgets the visit of a secret state, the secret states in an ∞-SSO problem can be viewed as *state faults*. The modeling power of state faults and event faults are proved to be equivalent in automata (Kumar & Takai, 2010). However, the necessary and sufficient conditions of fault diagnosability (which is related to the existence of *indeterminate cycles* in diagnosers) and ∞-SSO property (the condition in Theorem 4.1) are completely different.

Second, we note that the k-SR designed in Section 5 can be viewed as the conventional *observer automaton* (Cassandras & Lafortune, 2008) when $k = 0$. In fact, for all pairs (x, γ) in all macro-states in $G_{D,0}$, flag $\gamma = 0$ (resp., $\gamma = 1$) if and only if $x \notin X_S$ (resp., $x \in X_S$). Thus, a plant is current-state opacity if and only if there exists a macro-state d in $G_{D,0}$ such that all plant states in d are secret states.

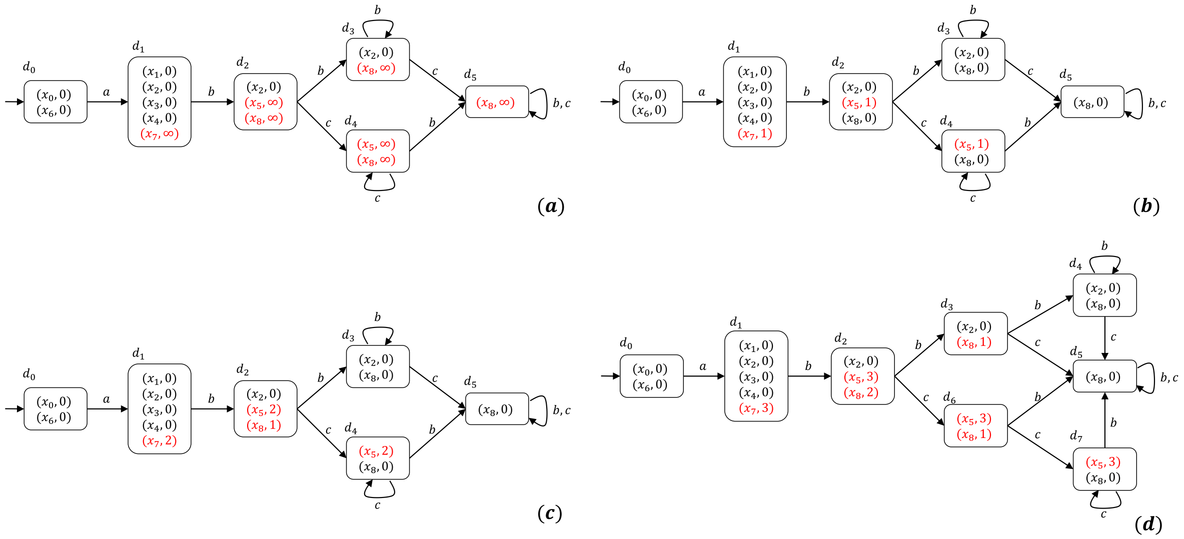

Fig. 3. The SRs for the automaton in Example 5.1: (a) ∞-SR, (b) 0-SR, (c) 1-SR, and (d) 2-SR.

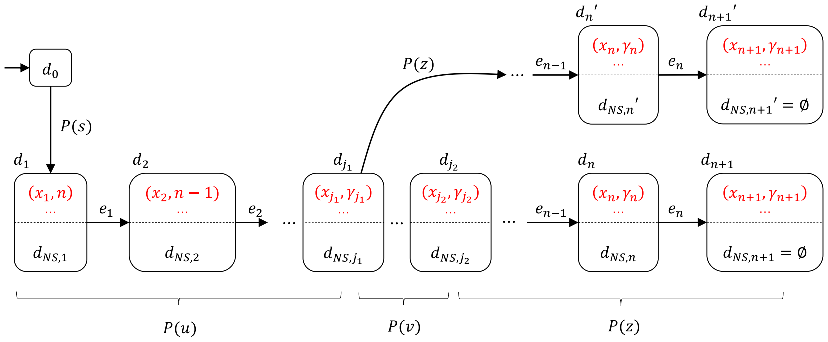

Fig. 4. Illustration of the proof of Theorem 5.2.

6. Strong opacity enforcement using supervisory control

In this section we study the ∞-SSO and k-SSO enforcing problem using supervisory control theory (Cassandras & Lafortune, 2008; Ramadge & Wonham, 1989). We briefly recall some notions on supervisory control in automata. For a plant automaton $G = (X, E, \delta, x_0)$, the event set E is partitioned into two disjoint subsets $E = E_c \dot{\cup} E_{uc}$ where E_c is the set of *controllable events* and E_{uc} is the set of *uncontrollable events*. In Ramadge and Wonham (1989), the control objective, called a *(language) specification*, is defined by a regular language $K \subseteq E^*$. In partially observed systems ($E_o \subset E$), a supervisor Sup can be viewed as a function $Sup : P[L(G)] \to 2^{E_c}$ such that the closed-loop language of Sup over G (denoted as $L(Sup/G)$) is restricted within K. In plain words, Sup runs in parallel with the plant such that for each observed sequence $w = P(s) \in P[L(G)]$, Sup disables some controllable events accordingly.

Now we define the ∞*-leaking language* and the k*-leaking language* as the following.

Definition 6.1. Given an ∞-SR $G_D = (D, E_o, \delta_\infty, d_0)$, the ∞*-leaking language* of G_d is defined as:

$$LL(G_D) = \{\sigma \in E_o^* \mid \delta_\infty(d_0, \sigma) \text{ is a leaking state}\}.$$

Definition 6.2. Given a k-SR $G_{D,k} = (D_k, E_o, \delta_k, d_0)$, the k*-leaking language* of $G_{d,k}$ is defined as:

$$LL(G_{D,k}) = \{\sigma \in E_o^* \mid \delta_k(d_0, \sigma) \text{ is a leaking state}\}.$$

Theorem 6.1. *Given a plant $G = (X, E, \delta, x_0)$, a set of observable events E_o, a set of secret states X_S, and a set of controllable events $E_c \subseteq E$, let G_D (resp., $G_{D,k}$) be the corresponding ∞-SR (resp., k-SR for a given $k \in \mathbb{N}$). A supervisor Sup that enforces a maximal controllable and observable sublanguage of $L(G) \setminus P^{-1}[LL(G_D)]$ (resp. $L(G) \setminus P^{-1}[LL(G_{D,k})]$) enforces ∞-SSO (resp. k-SSO).*

Proof. We only prove the case for ∞-SSO by contrapositive. The case for k-SSO can be analogously proved.

We prove by contrapositive that for any supervisor Sup such that $L(Sup/G) \subseteq L(G) \setminus P^{-1}[LL(G_D)]$, Sup/G is ∞-SSO. Suppose that Sup/G is not ∞-SSO. Following the argument "$\Leftarrow$" part of the proof of Theorem 4.1, there necessarily exists a sequence $s \in L(Sup/G)$ such that $P(s) \in P[LL(G_D)]$, which indicates that $L(Sup/G) \not\subseteq L(G) \setminus P^{-1}[LL(G_D)]$. Hence, $L(Sup/G) \subseteq L(G) \setminus P^{-1}[LL(G_D)]$ implies that Sup/G is ∞-SSO. Then we can conclude that a supervisor Sup that enforces a maximal controllable and observable sublanguage of $L(G) \setminus P^{-1}[LL(G_D)]$ enforces ∞-SSO. □

Corollary 6.1. *Both the ∞-SSO enforcing problem and the k-SSO enforcing problem can be reduced to the supervisory control*

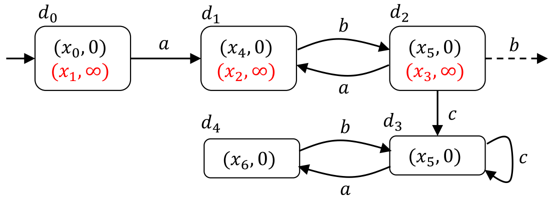

Fig. 5. The supervisor Sup_∞ in Example 6.1.

problem of G with language specification $K = E^* \setminus P^{-1}[LL(G_D)]$ and $K = E^* \setminus P^{-1}[LL(G_{D,k})]$.

Proof. This corollary directly follows from Theorem 6.1. □

As a result, we can enforce ∞-SSO and/or k-SSO for a plant G by enforcing a specification $K_\infty = E^* \setminus P^{-1}[LL(G_D)]$ (for ∞-SSO) or $K_k = E^* \setminus P^{-1}[LL(G_{D,k})]$ (for k-SSO). Such languages K_∞ and K_k can be obtained by manipulating the ∞-SR or k-SR in polynomial complexity (Cassandras & Lafortune, 2008), since both ∞-SR and k-SR are deterministic automata. Note that in partially observed systems, there may not exist a unique maximally permissive supervisor but several incomparable locally maximally solutions. Such locally maximally solutions can be obtained by using synthesis techniques in Hadj-Alouane, Lafortune, and Lin (1996), Takai (2020), Ushio (1999) and Yin and Lafortune (2016). For the limit of space, we do not address this in detail.

On the other hand, if $E_c \subseteq E_o$ holds, then for both ∞-SSO and k-SSO there exists a unique maximally permissive supervisor Sup: it recognizes the *supremal normal sublanguage* of K_∞ or K_k with respect to $L(G)$ (Cassandras & Lafortune, 2008). Therefore, a supervisor can be easily obtained by removing from G_D or $G_{D,k}$ all leaking states and those states from which there exists a path to some leaking state(s) labeled by uncontrollable events.

Example 6.1. Again consider the automaton G and the corresponding ∞-SR G_D in Fig. 1 in which $E_o = \{a, b, c\}$, $E_{uo} = \{u\}$, and the secret set $X_S = \{x_1\}$. Now, let us assume $E_c = \{a, b\}$. Since $E_c \subset E_o$, a maximally permissive supervisor can be obtained by introducing state specification $D_F = \{d_5, d_6\}$. By applying a standard trimming procedure we obtain the supervisor Sup_∞ depicted in Fig. 5. This supervisor disables event b whenever observing $ab(ab)^n$ such that the closed-loop system is ∞-SSO with respect to X_S.

Remark 3. Besides supervisory control, other opacity enforcement approaches such as *event edition* (Ji et al., 2018, 2019; Ji et al., 2019; Mohajerani et al., 2019; Wu & Lafortune, 2014; Yin & Li, 2020) may also be applied to enforce ∞-SSO and k-SSO. However, event edition requires to intrusively revise the plant structure by physically modifying the observation structure or the output information (by implementing new sensors and/or communication protocols), which may not be possible in some cases.

7. Conclusions

In this paper, we have developed a method to verify strong infinite-step opacity and k-step opacity using ∞-step recognizer and k-step recognizer, respectively. The complexities of our algorithms are $O(2^{2\cdot|X|} \cdot |E_o|)$ and $O(2^{(k+2)\cdot|X|} \cdot |E_o|)$, respectively, that are lower than that of previously-proposed methods in the literature. We have also proposed an upper bound for the value of k in strong k-step opacity. The enforcement of both strong infinite- and k-step opacity can be solved using supervisory control. In the future, we will combine this approach with Hu, Ma, and Li (2020) to explore the synthesis of SSO enforcing live supervisors.

Acknowledgments

This work was supported in part by the National Natural Science Foundation of China under Grant Nos. 61873342, 62061136004, 61803259, 61703321, Shaanxi Provincial Natural Science Foundation, China under Grant No. 2019JQ-022, the Fundamental Research Funds for the Central Universities, China under Grant Nos. JB210413, JB190407, and the Science and Technology Development Fund, MSAR, China, under Grant No. 0012/2019/A1.

References

Barcelos, R. J., & Basilio, J. C. (2018). Enforcing current-state opacity through shuffle in event observations. *IFAC-PapersOnLine*, *51*(7), 100–105, 14th IFAC Workshop on Discrete Event Systems WODES 2018.

Behinaein, B., Lin, F., & Rudie, K. (2019). Optimal information release for mixed opacity in discrete-event systems. *IEEE Transactions on Automation Science and Engineering*, *16*(4), 1960–1970.

Bryans, J. W., Koutny, M., Mazaré, L., & Ryan, P. Y. (2008). Opacity generalised to transition systems. *International Journal of Information Security*, *7*(6), 421–435.

Cassandras, C. G., & Lafortune, S. (2008). *Introduction to discrete event systems*. Springer.

Falcone, Y., & Marchand, H. (2015). Enforcement and validation (at runtime) of various notions of opacity. *Discrete Event Dynamic Systems*, *25*, 531–570.

Hadj-Alouane, N. B., Lafortune, S., & Lin, F. (1996). Centralized and distributed algorithms for on-line synthesis of maximal control policies under partial observation. *Discrete Event Dynamic Systems*, *6*(4), 379–427.

Hu, Y., Ma, Z., & Li, Z. (2020). Design of supervisors for active diagnosis in discrete event systems. *IEEE Transactions on Automatic Control*, *65*(12), 5159–5172.

Ji, Y., Wu, Y.-C., & Lafortune, S. (2018). Enforcement of opacity by public and private insertion functions. *Automatica*, *93*(7), 369–378.

Ji, Y., Yin, X., & Lafortune, S. (2019). Enforcing opacity by insertion functions under multiple energy constraints. *Automatica*, *108*, Article 108476.

Ji, Y., Yin, X., & Lafortune, S. (2019). Opacity enforcement using nondeterministic publicly known edit functions. *IEEE Transactions on Automatic Control*, *64*(10), 4369–4376.

Kumar, R., & Takai, S. (2010). Decentralized prognosis of failures in discrete event systems. *IEEE Transactions on Automatic Control*, *55*(1), 48–59.

Lafortune, S., Lin, F., & Hadjicostis, C. N. (2018). On the history of diagnosability and opacity in discrete event systems. *Annual Reviews in Control*, *45*, 257–266.

Lan, H., Tong, Y., & Seatzu, C. (2020). Verification of infinite-step opacity using labeled Petri nets. In *Proceedings of the 21st IFAC World Congress* (pp. 1729–1734).

Lin, F. (2011). Opacity of discrete event systems and its applications. *Automatica*, *47*(3), 496–503.

Ma, Z., Tong, Y., Li, Z., & Giua, A. (2017). Basis marking representation of Petri net reachability spaces and its application to the reachability problem. *IEEE Transactions on Automatic Control*, *62*(3), 1078–1093.

Mohajerani, S., Ji, Y., & Lafortune, S. (2019). Compositional and abstraction-based approach for synthesis of edit functions for opacity enforcement. *IEEE Transactions on Automatic Control*, *65*(8), 3349–3364.

Ramadge, P. J., & Wonham, W. M. (1989). The control of discrete event systems. *Proceedings of IEEE*, *77*(1), 81–98.

Saboori, A., & Hadjicostis, C. N. (2008). Verification of initial-state opacity in security applications of DES. In *Proceedings of the 9th international workshop on discrete event systems* (pp. 328–333).

Saboori, A., & Hadjicostis, C. N. (2011). Verification of K-step opacity and analysis of its complexity. *IEEE Transactions on Automation Science and Engineering*, *8*(3), 549–559.

Saboori, A., & Hadjicostis, C. N. (2012). Opacity-enforcing supervisory strategies via state estimator constructions. *IEEE Transactions on Automatic Control*, *57*(5), 1155–1165.

Takai, S. (2020). Synthesis of maximally permissive supervisors for nondeterministic discrete event systems with nondeterministic specifications. *IEEE Transactions on Automatic Control*, *66*(7), 3197–3204.

Tong, Y., & Lan, H. (2019). Current-state opacity verification in modular discrete event systems. In *Proceedings of the 2019 IEEE international conference on decision and control* (pp. 7665–7670).

Tong, Y., Li, Z. W., Seatzu, C., & Giua, A. (2017). Verification of state-based opacity using Petri nets. *IEEE Transactions on Automatic Control*, *62*(6), 2823–2837.

Tong, Y., Li, Z. W., Seatzu, C., & Giua, A. (2018). Current-state opacity enforcement in discrete event systems under incomparable observations. *Discrete Event Dynamic Systems*, *28*(2), 161–182.

Ushio, T. (1999). On-line control of discrete event systems with a maximally controllable and observable sublanguage. *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences*, *82*(9), 1965–1970.

Wu, Y.-C., & Lafortune, S. (2013). Comparative analysis of related notions of opacity in centralized and coordinated architectures. *Discrete Event Dynamic Systems*, *23*(3), 307–339.

Wu, Y.-C., & Lafortune, S. (2014). Synthesis of insertion functions for enforcement of opacity security properties. *Automatica*, *50*(5), 1336–1348.

Yin, X., & Lafortune, S. (2016). A uniform approach for synthesizing property-enforcing supervisors for partially-observed discrete-event systems. *IEEE Transactions on Automatic Control*, *61*(8), 2140–2154.

Yin, X., & Lafortune, S. (2017). A new approach for the verification of infinite-step and K-step opacity using two-way observers. *Automatica*, *80*, 162–171.

Yin, X., & Li, S. (2020). Synthesis of dynamic masks for infinite-step opacity. *IEEE Transactions on Automatic Control*, *65*(4), 1429–1441.

Ziyue Ma received the B.S. degree and the M.S. degree from Peking University, China, in 2007 and 2011, respectively. In 2017 he got the Ph.D degree in co-tutorship between the School of Electro-Mechanical Engineering of Xidian University, China (in Mechatronic Engineering), and the Department of Electrical and Electronic Engineering of University of Cagliari, Italy (in Electronics and Computer Engineering). He joined Xidian University in 2011, where he is currently an Associate Professor in the School of Electro-Mechanical Engineering. His research interests include control theory in discrete event systems, automaton and Petri net theories, fault diagnosis/prognosis, resource optimization, and information security.

Dr. Ma is a member of Technical Committee Member of IEEE Control System Society (IEEE-CSS) on Discrete Event Systems. He is serving/has served as the Associate Editor of the IEEE Conference on Automation Science and Engineering (CASE'17–'21), European Control Conference (ECC'19–'21), and IEEE International Conference on Systems, Man, and Cybernetics (SMC'19–'21). He is/was the Track Committee Member of the International Conference on Emerging Technologies and Factory Automation (ETFA'17–'21).

Xiang Yin was born in Anhui, China, in 1991. He received the B.Eng. degree from Zhejiang University in 2012, the M.S. degree from the University of Michigan, Ann Arbor, in 2013, and the Ph.D. degree from the University of Michigan, Ann Arbor, in 2017, all in electrical engineering. Since 2017, he has been with the Department of Automation, Shanghai Jiao Tong University, where he is an Associate Professor. His research interests include formal methods, discrete-event systems, and cyber–physical systems.

Dr. Yin is serving as the co-chair of the *IEEE CSS Technical Committee on Discrete Event Systems*, an Associate Editor for the *Journal of Discrete Event Dynamic Systems: Theory & Applications*, and a member of the *IEEE CSS Conference Editorial Board*. Dr. Yin received the IEEE Conference on Decision and Control (CDC) Best Student Paper Award Finalist in 2016. He is the co-chair of the IEEE CSS Technical Committee on Discrete Event Systems.

Zhiwu Li received the B.S., M.S., and Ph.D. degrees from Xidian University in 1989, 1992, and 1995, respectively. He joined Xidian University in 1992. His interests include discrete event systems and Petri nets. He published two monographs in Springer and CRC Press and 150+ papers in Automatica and IEEE Transactions (mostly regular). He was a Visiting Professor at the University of Toronto, Technion (Israel Institute of Technology), Martin-Luther University, Conservatoire National des Arts et Métiers (Cnam), Meliksah Universitesi, and King Saud University. His work was cited by engineers and researchers from more than 50 countries and areas, including prestigious R&D institutes such as IBM, Volvo, HP, GE, GM, ABB, and Huawei. Now, he is also with the Institute of Systems Engineering, Macau University of Science and Technology, Taipa, Macau.

Dr. Li serves (served) an Associate Editor of the IEEE Trans. Automation Science and Engineering, IEEE Trans. Systems, Man, and Cybernetics, Part A: Systems and Human Beings, IEEE Trans. Systems, Man, and Cybernetics: Systems, IEEE Access (also a Senior Editor), Scientific Reports, and Information Sciences (Elsevier). He is a recipient of Alexander von Humboldt Research Grant and Research in Paris.