Automatica 99 (2019) 266–274

Contents lists available at ScienceDirect

Automatica

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

Brief paper

Infinite-step opacity and K-step opacity of stochastic discrete-event systems☆

Xiang Yin [a,b,*], Zhaojian Li [c], Weilin Wang [d], Shaoyuan Li [a,b]

[a] Department of Automation, Shanghai Jiao Tong University, Shanghai 200240, China
[b] Key Laboratory of System Control and Information Processing, Ministry of Education of China, Shanghai 200240, China
[c] Department of Mechanical Engineering, Michigan State University, East Lansing, MI 48824, USA
[d] Faculty of Engineering, Monash University, Clayton, VIC 3800, Australia

ARTICLE INFO

Article history:
Received 25 January 2018
Received in revised form 8 August 2018
Accepted 8 September 2018

Keywords:
Discrete-event systems
Security
Infinite-step opacity
K-step opacity

ABSTRACT

Opacity is an important information-flow property that arises in security and privacy analysis of cyber–physical systems. Among many different notions of opacity, K-step opacity requires that the intruder can never determine unambiguously that the system was at a secret state for any specific instant within K steps prior to that particular instant. This notion becomes infinity-step opacity when K goes to infinity. Existing works on the analysis of infinite-step opacity and K-step opacity only provide a binary characterization, i.e., a system is either opaque or non-opaque. To analyze infinite-step and K-step opacity more quantitatively, in this paper, we investigate the verification of infinite-step and K-step opacity in the context of stochastic discrete-event systems. A new notion of opacity, called almost infinite-step opacity (respectively, almost K-step opacity), is proposed to capture whether or not the probability of violating infinite-step opacity (respectively, K-step opacity) is smaller than a given threshold. We also provide effective algorithms for the verification of almost infinite-step opacity and almost K-step opacity.

1. Introduction

In this paper, we investigate the verification of opacity for Discrete-Event Systems (DES). We assume that the system is monitored by a (potentially malicious) intruder that is modeled as a passive observer. A system is said to be opaque if the intruder can never determine the system's "secret" unambiguously based on its limited observation. In the past years, opacity has drawn considerable attention in the DES literature; see, e.g., Badouel, Bednarczyk, Borzyszkowski, Caillaud, and Darondeau (2007), Bourouis, Klai, Ben Hadj-Alouane, and El Touati (2017), Bryans, Koutny, Mazaré, and Ryan (2008), Cassez, Dubreil, and Marchand (2012), Chédor, Morvan, Pinchinat, and Marchand (2015), Darondeau, Marchand, and Ricker (2014), Dubreil, Darondeau, and Marchand (2010), Falcone and Marchand (2015), Lin (2011), Mullins and Yeddes (2014), Paoli and Lin (2012), Saboori and Hadjicostis (2011), Saboori and Hadjicostis (2012), Saboori and Hadjicostis (2013), Takai and Oka (2008), Tong, Li, Seatzu, and Giua (2017), Wu and Lafortune (2013), Yin and Lafortune (2016), Yin and Lafortune (2017) and Zhang, Shu, and Lin (2015) and a recent survey (Jacob, Lesage, & Faure, 2016) for more references.

In the context of DES, different notions of opacity have been proposed to capture different types of security requirements. Particularly, in Saboori and Hadjicostis (2011), *K-step opacity* was proposed to capture whether or not the intruder may know that the system was/is at a secret state for some specific instant within K (observation) steps prior to that particular instant. When K goes to infinity, K-step opacity becomes *infinite-step opacity* (Saboori & Hadjicostis, 2012). The verification of infinite-step opacity and K-step opacity was originally studied in Saboori and Hadjicostis (2011) and Saboori and Hadjicostis (2012); an improved approach was proposed recently by using a structure called the two-way observer (Yin & Lafortune, 2017).

The definitions of infinite-step opacity and K-step opacity only provide a binary characterization, i.e., a system is either opaque or not. However, a non-opaque system may only have a small probability of violation; this may be still tolerable in many applications. Therefore, recently, many works have considered how to quantitatively evaluate opacity by using probabilistic models (stochastic DES), e.g., Bryans, Koutny, and Mu (2012), Bérard, Mullins, and Sassolas (2015), Bérard, Chatterjee, and Sznajder (2015), Chen, Ibrahim, and Kumar (2017), Keroglou and Hadjicostis

☆ This work is supported by the National Natural Science Foundation of China (61803259, 61833012). The material in this paper was partially presented at the 11th Asian Control Conference, December 17–20, 2017, Gold Coast, Australia. This paper was recommended for publication in revised form by Associate Editor Rong Su under the direction of Editor Christos G. Cassandras.

* Corresponding author at: Department of Automation, Shanghai Jiao Tong University, Shanghai 200240, China.
E-mail address: yinxiang@sjtu.edu.cn (X. Yin).

https://doi.org/10.1016/j.automatica.2018.10.049
0005-1098/

(2013), Keroglou and Hadjicostis (2017a) and Saboori and Hadjicostis (2014). By precisely capturing the transition probability of the system, one is able to evaluate the possibility of being not secure, rather than simply providing a binary answer. In Keroglou and Hadjicostis (2015), Keroglou and Hadjicostis (2017b) and Yin (2017), a related property called detectability was also investigated in the context of stochastic DES.

All existing works on opacity analysis of stochastic DES only consider *current-state-type* opacity. (Note that initial-state opacity is also a current-state-type property if we augment the state space of the system by encoding the initial states.) However, for infinite-step opacity, we also need to consider how future information can affect our knowledge about the current status of the system, which is much more challenging. To the best of our knowledge, how to evaluate infinite-step opacity and K-step opacity, where delayed information is involved, has still not yet been investigated in the context of stochastic DES.

In this paper, we investigate the analysis of infinite-step opacity and K-step opacity in the context of stochastic DES. The main contributions of this paper are as follows. First, we define the notion of *almost infinite-step opacity* (respectively, *almost K-step opacity*) to capture whether or not the cumulated probability of violating infinite-step opacity (respectively, K-step opacity) is smaller than a given threshold. Then we propose effective approaches for the verification of almost infinite-step opacity and almost K-step opacity. Our definitions of almost infinite-step opacity and almost K-step opacity are motivated by the definitions of almost current-state opacity (Saboori & Hadjicostis, 2014) and almost initial-state opacity (Keroglou & Hadjicostis, 2013). However, the proposed verification algorithms are quite different from those in Keroglou and Hadjicostis (2013) and Saboori and Hadjicostis (2014).

The main difficulty in our problem is how to compute the violation languages for infinite-step and K-step opacity. This problem cannot be addressed directly by existing approaches. In Saboori and Hadjicostis (2011), an information structure was proposed for the verification of K-step opacity. The basic idea is to remember the last K events together with the current-state estimate; hence, we can recover the trajectory in the past K steps. However, when K goes to infinity, this structure requires infinite memory. In Saboori and Hadjicostis (2012), an approach was proposed for the verification of infinite-step opacity. The idea is to consider, for each possible current-state estimate, whether or not the current secret can be revealed in the future. However, this approach still fails to capture the violation language since it will provide a set of violation languages with different initial configurations. Our approach addresses the above difficulties by proposing a new *single and finite* information structure. Specifically, the proposed new structure precisely captures all possible delayed state estimates along the trajectory. Moreover, the state of the structure can be updated recursively, via a set of initial-state-estimators, upon the occurrence of a new observable event.

Preliminary and partial version of this paper is presented in Yin, Li, Wang, and Li (2017) without proofs; herein, we provide all proofs omitted and provide more detail explanations. Moreover, Yin, Li et al. (2017) only investigates almost infinite-step opacity and this paper extends the result in Yin, Li et al. (2017) to also include almost K-step opacity.

2. Preliminaries

2.1. System model

Let Σ be a set of events. A string over Σ is a finite sequence of events $s = \sigma_1 \dots \sigma_n$, $\sigma_i \in \Sigma$. We denote by Σ^* the set of all strings over Σ including the empty string ϵ. For any string $s \in \Sigma^*$, we denote by $|s|$ its length with $|\epsilon| = 0$. A language $L \subseteq \Sigma^*$ is a set of strings; we denote by $\overline{L}$ the prefix-closure of L, i.e., $\overline{L} := \{s \in \Sigma^* : \exists t \in \Sigma^* \text{ s.t. } st \in L\}$. For any string $s \in \Sigma^*$, we denote by $t \leq s$ if $t \in \overline{\{s\}}$ and denote by $t < s$ if $t \in \overline{\{s\}} \setminus \{s\}$. For any prefix $t \leq s$ of s, we denote by s/t the post string of t in s, i.e., $t(s/t) = s$.

We consider a DES modeled as a deterministic finite-state automaton (DFA) $G = (X, \Sigma, \delta, x_0)$, where X is the finite set of states, Σ is the finite set of events, $\delta : X \times \Sigma \to X$ is a (partial) deterministic transition function and x_0 is the unique initial state. The transition function δ is also extended to $X \times \Sigma^* \to X$ in the usual manner; see, e.g., Cassandras and Lafortune (2008). For the sake of simplicity, we denote $\delta(x, s)$ by $\delta(s)$ if $x = x_0$. We denote by $\mathcal{L}(G) = \{s \in \Sigma^* : \delta(s)!\}$ the language generated by G, where "!" means "is defined".

A stochastic discrete-event system is modeled as a probabilistic finite-state automaton (PFA) (G, p), where $G = (X, \Sigma, \delta, x_0)$ is a DFA and $p : X \times \Sigma \to [0, 1]$ is the transition probability function. Specifically, for any $x \in X, \sigma \in \Sigma$, we write $p(\sigma \mid x)$ the probability that event σ occurs from state x. We assume that (i) $\forall x \in X : \sum_{\sigma \in \Sigma} p(\sigma \mid x) = 1$; and (ii) $\forall x \in X, \sigma \in \Sigma : p(\sigma \mid x) > 0 \Leftrightarrow \delta(x, \sigma)!$. For any string $s \in \mathcal{L}(G)$, we denote by $Pr(s)$ the probability that s occurs, i.e., $Pr(\epsilon) = 1$ and $Pr(s\sigma) = Pr(s)p(\sigma \mid \delta(s))$, $s \in \Sigma^*, \sigma \in \Sigma$.

2.2. Intruder model

In opacity analysis of DES, we assume that the intruder is modeled as a *passive observer* that can observe partial behavior of the system and then *infer* the secret of the system based on its imperfect information. To this end, we assume that the event set Σ is partitioned into two disjoint sets: $\Sigma = \Sigma_o \dot{\cup} \Sigma_{uo}$, where Σ_o is the set of observable events and Σ_{uo} is the set of unobservable events. The natural projection $P : \Sigma^* \to \Sigma_o^*$ is defined recursively by: for any $s \in \Sigma^*, \sigma \in \Sigma$, we have

$$P(\epsilon) = \epsilon \quad \text{and} \quad P(s\sigma) = \begin{cases} P(s)\sigma & \text{if } \sigma \in \Sigma_o \\ P(s) & \text{if } \sigma \in \Sigma_{uo} \end{cases} \tag{1}$$

The natural projection is also extended to $P : 2^{\Sigma^*} \to 2^{\Sigma_o^*}$ by $P(L) = \{P(s) \in \Sigma_o^* : s \in L\}$ for any $L \subseteq \Sigma^*$.

Based on its observation, the intruder can *infer* which state the system could be in at some specific instant. Formally, let $\alpha \in P(\mathcal{L}(G))$ be an observed string. Then the *current state estimate* upon the occurrence of α is defined by

$$\hat{X}_G(\alpha) = \{x \in X : \exists s \in \mathcal{L}(G) \text{ s.t. } P(s) = \alpha, x = \delta(s)\}$$

The current state estimate can be computed by building the observer automaton (current state estimator); see, e.g., Cassandras and Lafortune (2008).

In some situations, the intruder is also interested in knowing which state the system could be in for some particular previous instant. Suppose that $\alpha \in P(\mathcal{L}(G))$ is observed and we are interested in the state estimate of the system for the instant when only $\beta \leq \alpha$ was executed. Then we define the *delayed state estimate* for β given α been observed by

$$\hat{X}_G(\beta \mid \alpha) = \{x \in X : \exists st \in \mathcal{L}(G) \text{ s.t. } P(s) = \beta, P(st) = \alpha, x = \delta(s)\}$$

Intuitively, $\hat{X}_G(\beta \mid \alpha)$ estimates the state of the system $|\alpha| - |\beta|$ steps prior to the instant when α is observed. Clearly, we have that $\hat{X}_G(\alpha \mid \alpha) = \hat{X}_G(\alpha)$.

Finally, we define the following operators that will be used later. Let $r \in 2^X$ be a set of states and $\sigma \in \Sigma_o$ be an observable event. The unobservable reach is defined by:

$$UR(r) = \{x \in X : \exists x' \in r, \exists s \in \Sigma_{uo}^* \text{ s.t. } \delta(x', s) = x\}$$

The observable transition is defined by:

$$Next_\sigma(r) = \{x \in X : \exists x' \in r \text{ s.t. } \delta(x', \sigma) = x\}$$

Let $\alpha \in \Sigma_o^*$ be an observed string. Operator $\Xi : \Sigma_o^* \to 2^{X\times X}$ is defined by:

$$\Xi(\alpha) = \{(x, x') \in X \times X : \exists s \in \Sigma^* \text{ s.t. } P(s) = \alpha, x' = \delta(x, s)\}$$

Let $\tilde{r}_1, \tilde{r}_2 \in 2^{X\times X}$ be two sets of state pairs. Operator $\circ : 2^{X\times X} \times 2^{X\times X} \to 2^{X\times X}$ is defined by:

$$\tilde{r}_1 \circ \tilde{r}_2 = \{(x_1, x_3) \in X \times X : \exists x_2 \in X \text{ s.t. } (x_1, x_2) \in \tilde{r}_1, (x_2, x_3) \in \tilde{r}_2\}$$

Let $r \in 2^X$ be a set of states. We define operator $\odot : 2^X \to 2^{X\times X}$ by:

$$\odot(r) = \{(x, x') \in r \times X : \exists s \in \Sigma_{uo}^* \text{ s.t. } \delta(x, s) = x'\}$$

3. Infinite-step and K-step opacity in stochastic DES

In this section, we propose the notions of almost infinite-step opacity and almost K-step opacity for stochastic DES.

In opacity analysis, we assume that the system has a "secret". Specifically, we model the "secret" of the system as a set of states $X_S \subseteq X$. Then, within this setting, K-step opacity (respectively, infinite-step opacity) requires that once a secret state is visited, then in the next K steps (respectively, infinite steps), the intruder cannot infer for sure that the system was at a secret state for that particular instant. We define

$$L_{\mathcal{KS}} = \left\{ s \in \mathcal{L}(G) : \begin{array}{c} \exists \alpha \leq P(s) \text{ s.t.} \\ |P(s)/\alpha| \leq K, \hat{X}_G(\alpha \mid P(s)) \subseteq X_S \end{array} \right\} \tag{2}$$

$$L_{\mathcal{IF}} = \{s \in \mathcal{L}(G) : \exists \alpha \leq P(s) \text{ s.t. } \hat{X}_G(\alpha \mid P(s)) \subseteq X_S\} \tag{3}$$

Then K-step opacity and infinity-step opacity require that $L_{\mathcal{KS}} = \emptyset$ and $L_{\mathcal{IF}} = \emptyset$, respectively. The reader is referred to Yin and Lafortune (2017) for the verification of these two properties.

However, infinite-step opacity and K-step opacity only provide binary characterizations, i.e., a system is either opaque or non-opaque. These notions do not consider the system's transition probability into account. In many applications, a small probability of violation may still be tolerable. Therefore, to quantitatively evaluate K-step opacity and infinite-step opacity, it may also be useful to consider the transition probability of the system into account. This motivates the definition of almost infinite-step/K-step opacity.

Recall that $L_{\mathcal{IF}}$ is the set of strings whose occurrences violate infinite-step opacity for some instant. In order to consider the cumulated probability of the violation of infinite-step opacity, we only need to consider those strings violating infinite-step opacity for the first time. Therefore, we define the following language:

$$L_{\mathcal{IF}}^{\mathcal{P}} = \{s \in L_{\mathcal{IF}} : \forall t < s \text{ s.t. } t \notin L_{\mathcal{IF}}\} \tag{4}$$

Now, we are already to define almost infinite-step opacity.

Definition 1 (*Almost Infinite-Step Opacity*). Let (G, p) be a PFA, $\Sigma_o \subseteq \Sigma$ be the set of observable events, $X_S \subseteq X$ be a set of secret state and $\theta < 1$ be a threshold value. We say that (G, p) is almost infinite-step opaque (w.r.t. Σ_o, X_S and θ) if $\sum_{s\in L_{\mathcal{IF}}^{\mathcal{P}}} Pr(s) < \theta$.

Essentially, almost infinite-step opacity requires that the cumulated probability of strings that violate infinite-step opacity in the logic sense is smaller than a given threshold θ. The reason why we consider language $L_{\mathcal{IF}}^{\mathcal{P}}$ rather than language $L_{\mathcal{IF}}$ is that, once the secret of the system is revealed by some string, any of its continuation will also reveal the secret. Therefore, we only need to consider strings in $L_{\mathcal{IF}}^{\mathcal{P}}$ to avoid counting the probability of violation duplicately.

Similar to almost infinite-step opacity, almost K-step opacity can be defined analogously. We define language $L_{\mathcal{KS}}^{\mathcal{P}}$ as the set of strings that violate K-step opacity for the first time, i.e.,

$$L_{\mathcal{KS}}^{\mathcal{P}} = \{s \in L_{\mathcal{KS}} : \forall t < s \text{ s.t. } t \notin L_{\mathcal{KS}}\} \tag{5}$$

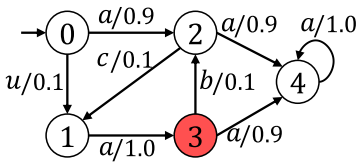


Fig. 1. System (G, p) with $\Sigma_o = \{a, b, c\}$ and $X_S = \{3\}$.

Then almost K-step opacity is defined as follows.

Definition 2 (*Almost K-Step Opacity*). Let (G, p) be a PFA, $\Sigma_o \subseteq \Sigma$ be the set of observable events, $X_S \subseteq X$ be a set of secret state, $K \in \mathbb{N}$ be a non-negative integer and $\theta < 1$ be a threshold value. We say that (G, p) is almost K-step opaque (w.r.t. Σ_o, X_S and θ) if $\sum_{s\in L_{\mathcal{KS}}^{\mathcal{P}}} Pr(s) < \theta$.

We illustrate almost infinite-step opacity and almost K-step opacity by the following example.

Example 1. Let us consider system (G, p) shown in Fig. 1, where $\Sigma_o = \{a, b, c\}$ and $X_S = \{3\}$. Let us consider $K = 1$. Then we have $L_{\mathcal{IF}} = L_{\mathcal{KS}} = \{uab, aca\}\Sigma^* \cap \mathcal{L}(G)$ and $L_{\mathcal{IF}}^{\mathcal{P}} = L_{\mathcal{KS}}^{\mathcal{P}} = \{uab, aca\}$. Since $\sum_{s\in L_{\mathcal{IF}}^{\mathcal{P}}} Pr(s) = \sum_{s\in L_{\mathcal{KS}}^{\mathcal{P}}} Pr(s) = 0.01 + 0.09 = 0.1$, we know that this system is almost infinite-step opaque and almost 1-step opaque for any threshold $\theta > 0.1$.

4. Verification of almost infinite-step opacity

In this section, we show how to formally verify almost infinite-step opacity.

To verify almost infinite-step opacity, the main idea is to construct a new automaton that (i) recognizes $L_{\mathcal{IF}}^{\mathcal{P}}$; and (ii) tracks the original transition probability of (G, p). When we consider current-state opacity or initial-state opacity, these requirements can be simply fulfilled by taking the product composition of G and its current state estimator or its initial-state estimator. However, this task is much more challenging for infinite-step opacity as $L_{\mathcal{IF}}^{\mathcal{P}}$ involves delayed information. Therefore, we need a new information structure that recognizes $L_{\mathcal{IF}}^{\mathcal{P}}$. This is detailed next.

Let G be a DFA. We define a new automaton

$$V_G = (Q, \Sigma, f, q_0) \tag{6}$$

where

- $Q \subseteq X \times 2^X \times 2^{2^{X\times X}}$ is the set of states;
- Σ is the set of events;
- $f : Q \times \Sigma \to Q$ is the transition function defined by: for any $q = (x, r, R) \in X \times 2^X \times 2^{2^{X\times X}}$ and $\sigma \in \Sigma$, we have

$$f(q, \sigma) = \begin{cases} (\delta(x, \sigma), r, R) & \text{if } \sigma \in \Sigma_{uo} \\ (\delta(x, \sigma), r', R') & \text{if } \sigma \in \Sigma_o \end{cases} \tag{7}$$

  where

$$r' = UR(Next_\sigma(r)) \tag{8}$$

$$R' = \{\rho \circ \Xi(\sigma) \in 2^{X\times X} : \rho \in R\} \cup \{\odot(r')\} \tag{9}$$

- $q_0 = (x_0, r_0, R_0)$ is the unique initial state, where $r_0 = UR(\{x_0\})$ and $R_0 = \{\odot(r_0)\}$.

Let us explain the intuition of the above construction. Note that each state (x, r, R) in V_G consists of three components, which are used as follows. The first component x simply tracks the current state in the original system G. Hence, the transition of f is consistent with δ for this component and we have that $\mathcal{L}(V_G) = \mathcal{L}(G)$.

The second component r tracks the current state estimate of the original system. That is, $r = \hat{X}_G(P(s))$, where s is a string leading to (x, r, R). The transition function of this component is the same as the transition function of the standard observer automaton, i.e., $UR(Next_\sigma(\cdot))$. The third component R is used to track the following information. Note that, R is a set of sets of state pairs. Each $\rho \in R$ essentially represents the delayed state estimate of the system for some (current or previous) instant. Specifically, for any $(x, x') \in \rho$, x is a state the system could be in at that instant and x' is a state the system could be in currently from x, and ρ consists of all such pairs for that specific instant. Although the number of previous instants may be infinite, R is a finite set as there are only finite such configurations for all possible delayed state estimates. Therefore, upon the occurrence of a new observable event, say σ, we need to update the delayed state estimate for all previous instants captured by $\{\rho \circ \Xi(\sigma) : \rho \in R\}$, and, at the same time, remember the current state estimate captured by $\{\odot(r')\}$.

Example 2. Again, let us consider system (G, p) shown in Fig. 1. Its corresponding automaton V_G is shown in Fig. 2. First, V_G starts with the initial state $(x_0, r_0, R_0) = (0, \{0, 1\}, \{\{(0, 0), (0, 1), (1, 1)\}\})$. The second component means that the current state estimate is $\hat{X}_G(\epsilon) = \{0, 1\}$ and the third component means that, within the initial step, the system may start from state 0 and end up with state 0, or start from state 0 and end up with state 1, or start from state 1 and end up with state 1. When observable event a occurs, we move to a new state $(x_1, r_1, R_1) = (2, \{2, 3\}, \{\{(0, 2), (0, 3), (1, 3)\}, \{(2, 2), (3, 3)\}\})$, where $\{2, 3\}$ is the current state estimate of the system, and $\{(0, 0), (0, 1), (1, 1)\} \in R_0$ is updated to

$$\begin{aligned}&\{(0, 0), (0, 1), (1, 1)\} \circ \Xi(a)\\ =&\{(0, 0), (0, 1), (1, 1)\} \circ \left\{\begin{array}{c}(0, 2), (0, 3), (1, 3),\\ (2, 4), (3, 4), (4, 4)\end{array}\right\}\\ =&\{(0, 2), (0, 3), (1, 3)\}\end{aligned}$$

which is the updated knowledge about the system for the instant one step ago, i.e., the delayed state estimate. At the same time, we need to add $\odot(\{2, 3\}) = \{(2, 2), (3, 3)\}$ to R_1 in order to remember the new state estimate for the current instant.

Next, we formally summarize the properties of V_G. First, for any $\rho \in 2^{X \times X}$, we define

$$I_1(\rho) = \{x \in X : \exists x' \in X \text{ s.t. } (x, x') \in \rho\} \tag{10}$$

as the set of states in the first component of ρ. Also, for any $R \in 2^{2^{X \times X}}$, we define

$$I_1(R) = \{I_1(\rho) \in 2^X : \rho \in R\} \tag{11}$$

Then we have the following result, which essentially states the intuition of V_G explained earlier.

Lemma 1. *For any string $s \in \mathcal{L}(V_G) = \mathcal{L}(G)$, let $f(q_0, s) = (x_s, r_s, R_s)$ be the state reached in V_G via s, then we have:*

(i) $r_s = \hat{X}_G(P(s))$;
(ii) $I_1(R_s) = \{\hat{X}_G(\alpha \mid P(s)) \in 2^X : \alpha \leq P(s)\}$.

Proof. (i) follows from the fact that $UR(Next(\cdot))$ is actually the well-known *observer* (or current-state estimator) construction; see, e.g., Cassandras and Lafortune (2008). Hereafter, we focus on the proof of (ii). First, we claim that

$$R_s = \{\rho_{\alpha, P(s)} \in 2^{X \times X} : \alpha \leq P(s)\} \tag{12}$$

where

$$\begin{aligned}&\rho_{\alpha, P(s)} = \{(x, x') \in X \times X : \exists t w \in \mathcal{L}(G) \text{ s.t.}\\ &\delta(t) = x, \delta(tw) = x', P(t) = \alpha, P(tw) = P(s)\}\end{aligned} \tag{13}$$

We prove this claim by induction on the length of $P(s)$.

Induction Basis: Suppose that $|P(s)| = 0$, i.e., $P(s) = \epsilon$. Then we know that $R_s = R_0$ and

$$\begin{aligned}&\odot(UR(\{x_0\}))\\ =&\{(x, x') \in X \times X : \exists x \in UR(\{x_0\}), w \in \Sigma_{uo}^* \text{ s.t. } \delta(x, w) = x'\}\\ =&\left\{(x, x') \in X \times X : \begin{array}{c}\exists t \in \Sigma_{uo}^*, \exists w \in \Sigma_{uo}^* \text{ s.t.}\\ \delta(x_0, t) = x, \delta(x, w) = x'\end{array}\right\}\\ =&\left\{(x, x') \in X \times X : \begin{array}{c}\exists t w \in \mathcal{L}(G) \text{ s.t.}\\ \delta(t) = x, \delta(tw) = x', P(tw) = \epsilon\end{array}\right\}\\ =&\rho_{\epsilon, \epsilon}\end{aligned}$$

Therefore, we know that $R_0 = \{\odot(UR(\{x_0\}))\} = \{\rho_{\epsilon, \epsilon}\}$.

Induction Step: Now, let us assume that, for $|P(s)| = k$, Eq. (12) holds. We need to prove that Eq. (12) still holds for $|P(s)| = k + 1$. To this end, we write s in the form of $s = t\sigma\xi$, where $|P(t)| = k$, $\sigma \in \Sigma_o$ and $\xi \in \Sigma_{uo}^*$.

By the construction of V_G, we know that

$$R_s = \{\rho \circ \Xi(\sigma) \in 2^{X \times X} : \rho \in R_t\} \cup \{\odot(\hat{X}_G(P(s)))\} \tag{14}$$

By the induction hypothesis, we know that $R_t = \{\rho_{\alpha, P(t)} \in 2^{X \times X} : \alpha \leq P(t)\}$. Also, by the definition of Ξ and $\circ$, we know that $\rho_{\alpha, P(t)\sigma} = \rho_{\alpha, P(t)} \circ \Xi(\sigma)$. Therefore, we have

$$\begin{aligned}\{\rho \circ \Xi(\sigma) \in 2^{X \times X} : \rho \in R_t\} =&\{\rho_{\alpha, P(t)} \circ \Xi(\sigma) \in 2^{X \times X} : \alpha \leq P(t)\}\\ =&\{\rho_{\alpha, P(s)} \in 2^{X \times X} : \alpha \leq P(t)\}\end{aligned} \tag{15}$$

Moreover, we have

$$\odot(\hat{X}_G(P(s))) = \rho_{P(s), P(s)} \tag{16}$$

By combining Eqs. (14), (15) and (16), we know that

$$\begin{aligned}R_s =&\{\rho_{\alpha, P(s)} \in 2^{X \times X} : \alpha \leq P(t)\} \cup \{\rho_{P(s), P(s)}\}\\ =&\{\rho_{\alpha, P(s)} \in 2^{X \times X} : \alpha \leq P(s)\}\end{aligned}$$

This completes the induction step, i.e., Eq. (12) holds.

Then, by Eq. (12), we know that

$$\begin{aligned}I_1(\rho_{\alpha, P(s)}) =&\left\{x \in X : \begin{array}{c}\exists t w \in \mathcal{L}(G) \text{ s.t.}\\ \delta(t) = x, P(t) = \alpha, P(tw) = P(s)\end{array}\right\}\\ =&\hat{X}_G(\alpha \mid P(s))\end{aligned} \tag{17}$$

This implies that $I_1(R_s) = \{I_1(\rho_{\alpha, P(s)}) \in 2^X : \alpha \leq P(s)\} = \{\hat{X}_G(\alpha \mid P(s)) \in 2^X : \alpha \leq P(s)\}$, which completes the proof for (ii). □

By Lemma 1, we know that, for any string $s \in \mathcal{L}(V_G)$ such that $f(q_0, s) = (x, r, R)$, $I_1(R)$ is actually the set of all delayed state estimates for all previous instants. Therefore, to determine whether or not $s \in L_{\mathcal{IF}}$, it suffices to determine whether or not there exists $\rho \in R$ such that $I_1(\rho) \subseteq X_S$. More specifically, let

$$Q_{\mathcal{IF}} = \{(x, r, R) \in Q : \exists \rho \in R \text{ s.t. } I_1(\rho) \subseteq X_S\} \tag{18}$$

Then we have the following result.

Lemma 2. *For any string $s \in \mathcal{L}(V_G) = \mathcal{L}(G)$, we have*

$$L_{\mathcal{IF}}^{\mathcal{P}} = \{s \in \mathcal{L}(G) : [f(s) \in Q_{\mathcal{IF}}] \wedge [\forall t < s : f(s) \notin Q_{\mathcal{IF}}]\}$$

Proof. By Lemma 1, we know that

$$\begin{aligned}L_{\mathcal{IF}} =&\{s \in \mathcal{L}(G) : \exists \alpha \leq P(s) \text{ s.t. } \hat{X}(\alpha \mid P(s)) \subseteq X_S\}\\ =&\{s \in \mathcal{L}(G) : \exists \rho \in R \text{ s.t. } I_1(\rho) \subseteq X_S\}\\ =&\{s \in \mathcal{L}(G) : f(s) \in Q_{\mathcal{IF}}\}\end{aligned}$$

Therefore, $L_{\mathcal{IF}}^{\mathcal{P}} = \{s \in L_{\mathcal{IF}} : \forall t < s \text{ s.t. } t \notin L_{\mathcal{IF}}\} = \{s \in \mathcal{L}(G) : [f(s) \in Q_{\mathcal{IF}}] \wedge [\forall t < s : f(s) \notin Q_{\mathcal{IF}}]\}$. □

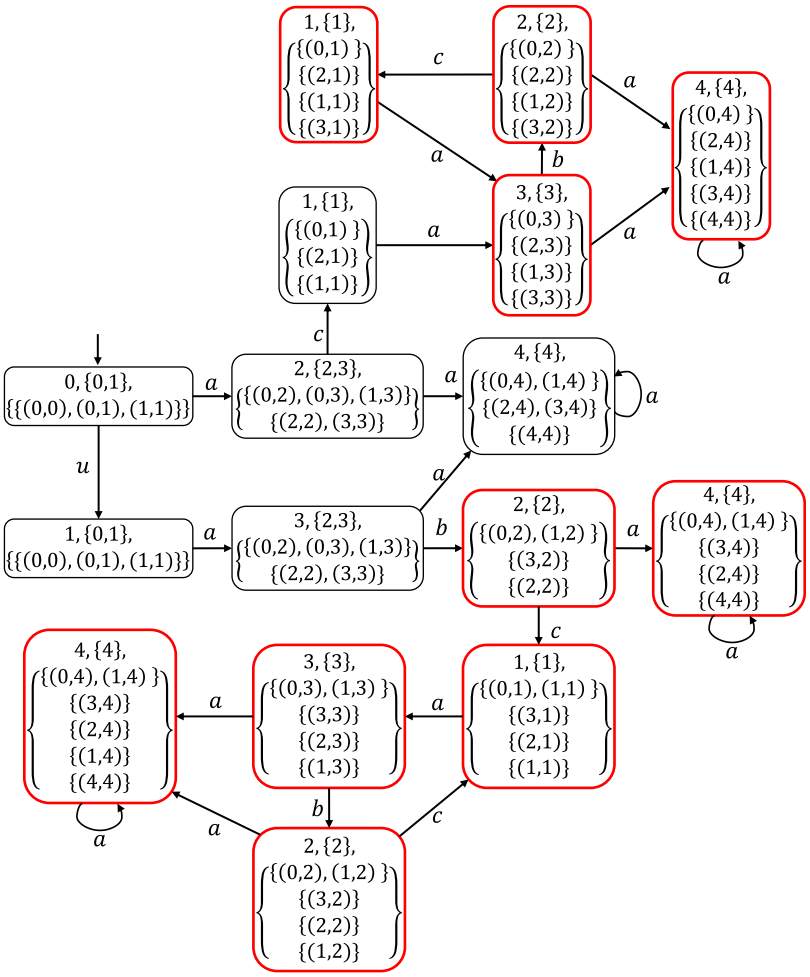


Fig. 2. Automaton V_G for the system in Fig. 1.

By Lemma 2, it is clear that, to compute $\sum_{s\in L^{\mathcal{P}}_{\mathcal{IF}}} Pr(s)$, it suffices to compute the probability of hitting a state in $Q_{\mathcal{IF}}$ in a Markov chain associated to V_G with transition probability reflecting the original system (G, p). This is formalized as follows. First, we denote by $\tilde{V}_G = (\tilde{Q}, \Sigma, \tilde{f}, q_0)$ the accessible part of the automaton obtained by removing all outgoing transitions from states in $Q_{\mathcal{IF}}$. Then we define a Markov Chain (MC) $\mathcal{M} = (\tilde{Q}, p_{\mathcal{M}}, \pi_0)$, where the state space of the MC is the same as the state space of $\tilde{V}_G$ and the transition probability function $p_{\mathcal{M}} : \tilde{Q} \times \tilde{Q} \to [0, 1]$ is defined by: for any $q = (x, r, R)$, $q' = (x', r', R') \in \tilde{Q}$,

$$p_{\mathcal{M}}(q' \mid q) = \begin{cases} \sum_{\sigma\in\Sigma: f(q,\sigma)=q'} p(\sigma \mid x) & \text{if } q \notin Q_{\mathcal{IF}} \\ 1 & \text{if } q = q' \in Q_{\mathcal{IF}} \\ 0 & \text{otherwise} \end{cases}$$

and $\pi_0 : \tilde{Q} \to [0, 1]$ is the initial state distribution such that $\pi_0(q_0) = 1$ and $\forall q \neq q_0 : \pi_0(q) = 0$.[1] Therefore, for $\mathcal{M}$, all states in $Q_{\mathcal{IF}} \cap \tilde{Q}$ are absorbing, i.e., once we reach a state in $Q_{\mathcal{IF}} \cap \tilde{Q}$, we will stay in it forever. This absorbing probability, denoted by $p^{abs}_{\mathcal{M}}(Q_{\mathcal{IF}})$, can be computed in a backward recursive manner by Norris (1998):

$$p^{abs}_{\mathcal{M}}(Q_{\mathcal{IF}}) = \sum_{q\in\tilde{Q}} \pi_0(q)\mathbb{P}(q) \tag{19}$$

where $\mathbb{P} : \tilde{Q} \to \mathbb{R}^+_0$ is the vector of minimal non-negative solution[2] to the following equation:

$$\mathbb{P}(q) = \begin{cases} \sum_{q'\in\tilde{Q}} p_{\mathcal{M}}(q' \mid q)\mathbb{P}(q') & \text{if } q \notin Q_{\mathcal{IF}} \\ 1 & \text{if } q \in Q_{\mathcal{IF}} \end{cases} \tag{20}$$

Since $\mathcal{L}(V_G) = \mathcal{L}(G)$ and $\mathcal{M}$ is constructed by tracking the transition probability of the original system, we have $p^{abs}_{\mathcal{M}}(Q_{\mathcal{IF}}) = \sum_{s\in L^{\mathcal{P}}_{\mathcal{IF}}} Pr(s)$. Hence, we have the following main theorem.

Theorem 3. *(G, p) is almost infinite-step opaque w.r.t. threshold θ if and only if $p^{abs}_{\mathcal{M}}(Q_{\mathcal{IF}}) < \theta$.*

Example 3. Still, let us consider system (G, p) shown in Fig. 1 and V_G is shown in Fig. 2. States in $Q_{\mathcal{IF}}$ are marked by red lines. For example, we know that $(2, \{2\}, \{\{(0, 2), (1, 2)\}, \{(3, 2)\}, \{(2, 2)\}\}) \in Q_{\mathcal{IF}}$, since $I_1(\{(3, 2)\}) = \{3\} \subseteq X_S$. Then its associated MC $\mathcal{M}$ is shown in Fig. 3. For the sake of simplicity, each state in $\mathcal{M}$ is renamed from M_1 to M_8. To compute the absorbing probability in

[1] We assume w.l.o.g. that $q_0 \in \tilde{Q}$; otherwise it implies that the system is not infinite-step opacity even for threshold $\theta = 1$.

[2] In general, the solution to Eq. (20) is not unique since states that cannot reach absorbing states are free-terms in the equations. Therefore, we need to assign zero to those states in order to find a minimal non-negative solution.

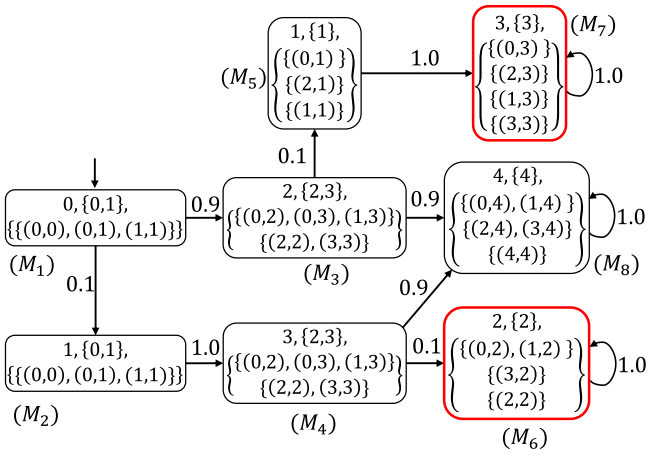


Fig. 3. Markov chain $\mathcal{M}$ for the system in Fig. 1.

$Q_{\mathcal{IF}} \cap \tilde{Q}$, we need to solve the following equations:

$$\begin{cases} \mathbb{P}(M_1) &= 0.1 \times \mathbb{P}(M_2) + 0.9 \times \mathbb{P}(M_3) \\ \mathbb{P}(M_2) &= \mathbb{P}(M_4) \\ \mathbb{P}(M_3) &= 0.1 \times \mathbb{P}(M_5) + 0.9 \times \mathbb{P}(M_8) \\ \mathbb{P}(M_4) &= 0.1 \times \mathbb{P}(M_6) + 0.9 \times \mathbb{P}(M_8) \\ \mathbb{P}(M_8) &= \mathbb{P}(M_8) \\ \mathbb{P}(M_5) &= \mathbb{P}(M_6) = \mathbb{P}(M_7) = 1 \end{cases} \tag{21}$$

Note that, the solution to Eq. (21) is not unique as $\mathbb{P}(M_8)$ is a free term. To obtain the minimal solution, we need to set $\mathbb{P}(M_8) = 0$ and we have $\mathbb{P} = [0.1\ 0.1\ 0.1\ 0.1\ 1\ 1\ 1\ 0]$. Therefore, we know that $\sum_{s \in L^{\mathcal{P}}_{\mathcal{IF}}} Pr(s) = p^{abs}_{\mathcal{M}}(Q_{\mathcal{IF}}) = \pi_0(M_1) \times \mathbb{P}(M_1) = 0.1$, i.e., the system is almost infinite-step opaque for any $\theta > 0.1$, and this is consistent with our result in Example 1.

5. Verification of almost K-step opacity

In this section, we show how to verify almost K-step opacity. To this end, we need to construct an automaton that recognizes $L^{\mathcal{P}}_K$. Our approach is similar to the idea of V_G by tracking delayed state estimates of previous instants. However, instead of tracking all possible previous instants, for K-step opacity, we only need to track delayed state estimates for previous instants within K steps prior to the current instant.

Formally, let G be a DFA. We define a new automaton

$$V^K_G = (Q_K, \Sigma, f_K, q_{0,K}) \tag{22}$$

where

- $Q_K \subseteq X \times 2^X \times \underbrace{2^{X\times X} \times \cdots \times 2^{X\times X}}_{K\text{-times}}$ is the set of states;
- Σ is the set of events;
- $f_K : Q_K \times \Sigma \to Q_K$ is the transition function defined by: for any $q = (x, r, \rho_1, \ldots, \rho_K) \in X \times 2^X \times \underbrace{2^{X\times X} \times \cdots \times 2^{X\times X}}_{K\text{-times}}$ and $\sigma \in \Sigma$, we have

$$f_K(q, \sigma) = \begin{cases} (\delta(x, \sigma), r, \rho_1, \ldots, \rho_K) & \text{if } \sigma \in \Sigma_{uo} \\ (\delta(x, \sigma), r', \rho'_1, \ldots, \rho'_K) & \text{if } \sigma \in \Sigma_o \end{cases} \tag{23}$$

where

$$r' = UR(Next_\sigma(r)) \tag{24}$$

$$\rho'_i = \{\tilde{r} \circ \Xi(\sigma) \in 2^{X\times X} : \tilde{r} \in \rho_{i-1}\} \tag{25}$$

where $\rho_0 = \odot(r)$.

- $q_{0,K} = (x_0, UR(\{x_0\}), \emptyset, \ldots, \emptyset)$ is the initial state.

Compared with V_G whose state space is $2^{2^{X\times X}}$, the state space of V^K_G is $(2^{X\times X})^K$ since we only need to remember the delayed state estimate of the system for instants in the previous K steps. Specifically, upon the occurrence of each observable event, ρ'_i is obtained by updating ρ_{i-1} with the new observable event and the information in ρ_K will not be used since it is about state of system $K+1$ steps ago. Note that ρ'_1 is obtained by taking $\rho_0 = \odot(r)$ since r is the current state estimate for the instant one step ago. Still, since the transition of f_K is consistent with δ, we also have $\mathcal{L}(V^K_G) = \mathcal{L}(G)$. It is worth remarking that state estimates in each state of V_G are unordered in the sense that we do not distinguish between different instants. However, state estimates in V^K_G are ordered since we need to precisely remember the associated instant for each delayed state estimate.

Lemma 4. *For any string $s \in \mathcal{L}(V^K_G) = \mathcal{L}(G)$, let $f_K(q_{0,K}, s) = (x, r, \rho_1, \ldots, \rho_K)$ be the state reached via string s, then we have:*

(i) $r = \hat{X}_G(P(s))$;
(ii) *For any $\alpha < P(s)$ such that $i := |P(s)| - |\alpha| \leq K$, we have $I_1(\rho_{|P(s)|-|\alpha|}) = \hat{X}_G(\alpha \mid P(s))$.*

Proof. (i) still follows from the fact that $UR(Next(\cdot))$ is the well-known observer construction. Hereafter, we focus on the proof of (ii). We claim that, for any $s \in \mathcal{L}(G)$, we have

$$\begin{aligned} &\rho_{|P(s)|-|\alpha|} = \{(x, x') \in X \times X : \exists uv \in \mathcal{L}(G) \text{ s.t.} \\ &\delta(u) = x, \delta(uv) = x', \wedge P(u) = \alpha, P(uv) = P(s)\} \end{aligned} \tag{26}$$

We prove this by induction on the length of $P(s)$.

Induction Basis: Suppose that $|P(s)| = 0$, i.e., $P(s) = \epsilon$. Then (ii) holds directly since there is no α such that $\alpha < \epsilon$.

Induction Step: Now, let us assume that, for $|P(s)| = k$, Eq. (26) holds. We need to prove that Eq. (12) still holds for $|P(s)| = k+1$. To this end, we write s in the form of $s = t\sigma\xi$, where $|P(t)| = k$, $\sigma \in \Sigma_o$ and $\xi \in \Sigma^*_{uo}$. Let $f_K(q_{0,K}, t) = (x', r', \rho'_1, \ldots, \rho'_K)$ and $f_K(q_{0,K}, t\sigma) = (x'', r'', \rho''_1, \ldots, \rho''_K)$. Since $\xi \in \Sigma^*_{uo}$, we know that $\rho''_i = \rho_i$. Therefore, it suffices to show that for any $i = 1, \ldots, K$, ρ''_i satisfies Eq. (26).

Let us consider a prefix $\alpha < P(t)\sigma$, i.e., $\alpha \leq P(t)$, such that $|P(s)| - |\alpha| \leq K$. We consider the following two cases for α.

If $\alpha = P(t)$, i.e., $|P(s)| - |\alpha| = 1$, we have

$$\begin{aligned} \rho''_1 &= \odot(r') \circ \Xi(\sigma) = \odot(\hat{X}_G(\alpha)) \circ \Xi(\sigma) \\ &= \left\{(x, x') \in X \times X : \begin{array}{c} \exists uv \in \mathcal{L}(G) \text{ s.t. } P(u) = \alpha, \\ P(uv) = \alpha\sigma, x = \delta(u), x' = \delta(v) \end{array}\right\} \end{aligned}$$

Therefore, $I_1(\rho_1) = \{x \in X : \exists uv \in \mathcal{L}(G) \text{ s.t. } P(u) = \alpha, P(uv) = \alpha\sigma, x = \delta(u)\} = \hat{X}_G(\alpha \mid P(s))$.

If $\alpha \neq P(t)$, i.e., $i := |P(s)| - |\alpha| > 1$, then we have

$$\rho''_{|P(s)|-|\alpha|} = \rho'_{|P(t)|-|\alpha|} \circ \Xi(\sigma) \tag{27}$$

Since $\rho'_{|P(t)|-|\alpha|}$ is reached via string t where $|P(t)| = k$, by the induction hypothesis, we know that

$$\begin{aligned} &\rho''_{|P(t)|-|\alpha|} = \{(x, x') \in X \times X : \exists uv \in \mathcal{L}(G) \text{ s.t.} \\ &\delta(u) = x, \delta(uv) = x', P(u) = \alpha, P(uv) = P(t)\} \end{aligned} \tag{28}$$

By combining Eqs. (27) and (28), we have

$$\begin{aligned} &\rho''_{|P(t)|-|\alpha|} = \{(x, x') \in X \times X : \exists uv \in \mathcal{L}(G) \text{ s.t.} \\ &\delta(u) = x, \delta(uv) = x', P(u) = \alpha, P(uv) = P(t)\sigma = P(s)\} \end{aligned}$$

This completes the induction step, i.e., Eq. (26) holds.

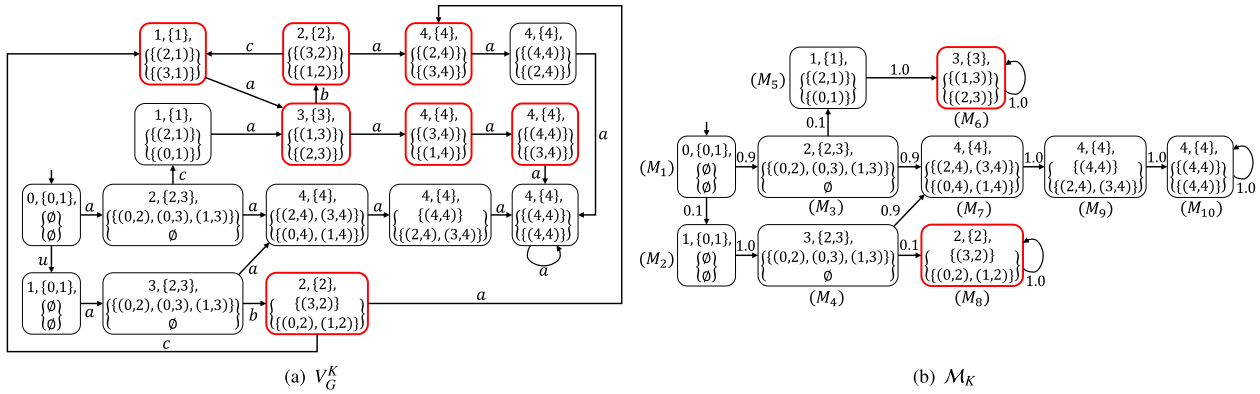


Fig. 4. An example for almost K-step opacity, where $K = 2$.

Then, by Eq. (26), we know that

$$I_1(\rho_{|P(s)|-|\alpha|}) = \left\{ x \in X : \begin{array}{c} \exists uv \in \mathcal{L}(G) \text{ s.t. } \delta(u) = x, \\ P(u) = \alpha, P(uv) = P(s) \end{array} \right\} = \hat{X}_G(\alpha \mid P(s)) \tag{29}$$

This completes the proof for (ii). □

By Lemma 4, we know that, for any string $s \in \mathcal{L}(V_G^K)$ s.t. $f_K(s) = (x, r, \rho_1, \ldots, \rho_K)$, r is still the current state estimate, but $I_1(\rho_i)$ is actually the delayed state estimate for the instant i steps ago. Therefore, to determine whether or not $s \in L_{\mathcal{KS}}$, it suffices to determine whether or not (i) $r \subseteq X_S$; or (ii) there exists $i = 1, \ldots, K$ s.t. $\emptyset \neq I_1(\rho_i) \subseteq X_S$. More specifically, let

$$Q_{\mathcal{KS}} = \{(x, r, \rho_1, \ldots, \rho_K) \in Q_K : [r \subseteq X_S] \vee [\exists i = 1, \ldots, K : \emptyset \neq I_1(\rho_i) \subseteq X_S]\} \tag{30}$$

Then we have the following result.

Lemma 5. *For any string $s \in \mathcal{L}(V_G) = \mathcal{L}(G)$, we have*

$$L_{\mathcal{KS}}^{\mathcal{P}} = \{s \in \mathcal{L}(G) : [f_K(s) \in Q_{\mathcal{KS}}] \wedge [\forall t < s : f_K(s) \notin Q_{\mathcal{KS}}]\}$$

Proof. By Lemma 4, we know that

$$\begin{aligned} L_{\mathcal{KS}} &= \left\{ s \in \mathcal{L}(G) : \begin{array}{c} \exists \alpha \leq P(s) \text{ s.t.} \\ |P(s)/\alpha| \leq K \wedge \hat{X}_G(\alpha \mid P(s)) \subseteq X_S \end{array} \right\} \\ &= \left\{ s \in \mathcal{L}(G) : \begin{array}{c} \hat{X}_G(P(s)) \subseteq X_S \text{ or} \\ (\exists \alpha < P(s) : |P(s)/\alpha| \leq K) \text{ s.t.} \\ \emptyset \neq \hat{X}_G(\alpha \mid P(s)) \subseteq X_S \end{array} \right\} \\ &= \{s \in \mathcal{L}(G) : f_K(s) \in Q_{\mathcal{KS}}\} \end{aligned}$$

Therefore, we have

$$\begin{aligned} L_{\mathcal{KS}}^{\mathcal{P}} &= \{s \in L_{\mathcal{KS}} : \forall t < s \text{ s.t. } t \notin L_{\mathcal{KS}}\} \\ &= \{s \in \mathcal{L}(G) : [f_K(s) \in Q_{\mathcal{KS}}] \wedge [\forall t < s : f_K(s) \notin Q_{\mathcal{KS}}]\} \end{aligned}$$

This completes the proof. □

Similar to the case of almost infinite-step opacity, by Lemma 5, we know that, to compute $\sum_{s \in L_{\mathcal{IF}}^{\mathcal{P}}} Pr(s)$, it suffices to compute the probability of hitting a state in $Q_{\mathcal{KS}}$ in a Markov chain associated to V_G^K with transition probability reflecting the original system (G, p). Still, we denote by $\tilde{V}_G^K = (\tilde{Q}_K, \Sigma, \tilde{f}_K, q_{0,K})$ the accessible part of the automaton obtained by removing all outgoing transitions from states in $Q_{\mathcal{KS}}$. Then we define a Markov chain $\mathcal{M}_K = (\tilde{Q}_K, p_{\mathcal{M},K}, \pi_{0,K})$, where the $p_{\mathcal{M}_K} : \tilde{Q}_K \times \tilde{Q}_K \to [0, 1]$ is defined by: for any $q = (x, r, \rho_1, \ldots, \rho_K)$, $q' = (x', r', \rho_1', \ldots, \rho_K') \in \tilde{Q}_K$,

$$p_{\mathcal{M}_K}(q' \mid q) = \begin{cases} \sum_{\sigma \in \Sigma : f_K(q,\sigma) = q'} p(\sigma \mid x) & \text{if } q \notin Q_{\mathcal{KS}} \\ 1 & \text{if } q = q' \in Q_{\mathcal{KS}} \\ 0 & \text{otherwise} \end{cases}$$

and $\pi_0 : \tilde{Q} \to [0, 1]$ is the initial state distribution defined by $\pi_0(q_0) = 1$. Still, all states in $Q_{\mathcal{KS}} \cap \tilde{Q}_K$ are constructed to be absorbing and we denote by $p_{\mathcal{M}_K}^{abs}(Q_{\mathcal{KS}})$ the absorbing probability of hitting a state in $Q_{\mathcal{KS}}$. Then, according to Lemma 5, we also have the following theorem.

Theorem 6. *(G, p) is almost K-step opaque w.r.t. threshold θ if and only if $p_{\mathcal{M}_K}^{abs}(Q_{\mathcal{KS}}) < \theta$.*

Example 4. Again, let us consider system (G, p) in Fig. 1 with $K = 2$. Then its corresponding automaton V_G^K is shown in Fig. 4(a) and states in $Q_{\mathcal{KS}}$ are marked by red lines. For example, we know that $(3, \{3\}, \{\{(1, 3)\}, \{(2, 3)\}\}) \in Q_{\mathcal{KS}}$ as $\{3\} \subseteq X_S$, and $(2, \{2\}, \{\{(3, 2)\}, \{(1, 2)\}\}) \in Q_{\mathcal{KS}}$ as $I_1(\{\{(3, 2)\}\}) \subseteq X_S$. Then its associated MC $\mathcal{M}_K$ is shown in Fig. 4(b). We can compute the absorbing probability of $Q_{\mathcal{KS}} \cap \tilde{Q}$, which is $\sum_{s \in L_{\mathcal{KS}}^{\mathcal{P}}} Pr(s) = p_{\mathcal{M}}^{abs}(Q_{\mathcal{KS}}) = \pi_0(M_1) \times \mathbb{P}(M_1) = 0.1$, i.e., the system is almost 2-step opaque for any $\theta > 0.1$. Note that, in this example, the thresholds for infinite-step opacity and 2-steps opacity coincide. In general, the threshold for K-step opacity is smaller than or equal to the threshold for infinite-step opacity.

Remark 7. It was shown in Yin and Lafortune (2017) that, in the logical setting, K-step opacity and infinite-step opacity coincide when K is greater than $2^{|X|} - 2$. However, in the stochastic setting, almost K-step opacity is always strictly weaker than almost $(K+1)$-step opacity no matter how large K is. For example, let us consider system (G, p) shown in Fig. 5 with $\Sigma_o = \{a, b\}$ and $X_S = \{3\}$. Then we have $L_{\mathcal{IF}}^{\mathcal{P}} = \{u_1 ab\}\{b\}^*\{a\}$ and, for any $K \geq 2$, $L_{\mathcal{KS}}^{\mathcal{P}} = \{u_1 ab^n a : 1 \leq n \leq K - 1\}$. Clearly, we see that $\sum_{s \in L_{\mathcal{KS}}^{\mathcal{P}}} Pr(s)$ strictly increases when K increases. Although the limit of $\sum_{s \in L_{\mathcal{KS}}^{\mathcal{P}}} Pr(s)$ is $\sum_{s \in L_{\mathcal{IF}}^{\mathcal{P}}} Pr(s)$, these two values are not equal no matter how large K is. This is why we need the proposed approach to compute $\sum_{s \in L_{\mathcal{KS}}^{\mathcal{P}}} Pr(s)$ for a given K.

Remark 8. We conclude this section by discussing the complexity of the proposed verification algorithms. For infinite-step opacity, we need to first construct V_G and $\mathcal{M}$; both contain at most $n_{max} := |X| \times 2^{|X|} \times 2^{2^{|X| \times |X|}}$ states. Then to obtain $p_{\mathcal{M}_K}^{abs}(Q_{\mathcal{IF}})$, it takes $O(n_{max}^3)$ to solve the linear equations. Therefore, the overall worst-case complexity is doubly-exponential in the size of G. Similarly, for

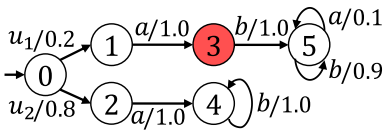


Fig. 5. System (G, p) with $\Sigma_o = \{a, b\}$ and $X_S = \{3\}$.

K-step opacity, $\mathcal{M}_K$ contains at most $\tilde{n}_{max} := |X| \times 2^{|X|} \times 2^{K\times|X|\times|X|}$ states and it also takes $O(\tilde{n}^3_{max})$ to solve the linear equations. Therefore, the overall worst-case complexity is exponential in both K and the size of G. We note that the complexity of verifying almost infinite-step (or K-step opacity) is higher than its logical counterpart. Recall that, using the algorithms in Yin and Lafortune (2017), both infinite-step and K-step opacity can be verified with single-exponential complexity only in the size of G. The higher complexity comes from the fact that, we need to precisely compute languages $L_{\mathcal{IF}}$ and $L_{\mathcal{KS}}$ in the stochastic setting. Therefore, one not only needs to track the current state estimate of the system, but also needs to precisely track all possible delayed state estimates for previous instants.

6. Conclusion

In this paper, we investigated the analysis of infinite-step opacity and K-step opacity in the context of stochastic DES. A new notion called almost infinite-step opacity (respectively, almost K-step opacity) was proposed to quantitatively evaluate the probability that infinite-step opacity (respectively, K-step opacity) is violated. An effective algorithm was provided for the verification of each notion. The complexity for verifying almost infinite-step opacity is doubly-exponential in the size of the original system and complexity for verifying almost K-step opacity is exponential in the size of the original system and K. Recently, an abstraction-based approach was proposed for efficiently verifying logical opacity by constructing a smaller system that preserves opacity (Noori-Hosseini, Lennartson, & Hadjicostis, 2018; Zhang, Yin, & Zamani, 2018). How to use such an abstraction-based technique to reduce the complexity of our verification algorithms is an interesting future direction.

References

Badouel, E., Bednarczyk, M., Borzyszkowski, A., Caillaud, B., & Darondeau, P. (2007). Concurrent secrets. *Discrete Event Dynamic Systems*, *17*(4), 425–446.

Bérard, B., Chatterjee, K., & Sznajder, N. (2015). Probabilistic opacity for Markov decision processes. *Information Processing Letters*, *115*(1), 52–59.

Bérard, B., Mullins, J., & Sassolas, M. (2015). Quantifying opacity. *Mathematical Structures in Computer Science*, *25*(2), 361–403.

Bourouis, A., Klai, K., Ben Hadj-Alouane, N., & El Touati, Y. (2017). On the verification of opacity in web services and their composition. *IEEE Transactions on Services Computing*, *10*(1), 66–79.

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

Bryans, J. W., Koutny, M., & Mu, C. (2012). Towards quantitative analysis of opacity. In *Int. symp. trustworthy global comp.* (pp. 145–163).

Cassandras, C. G., & Lafortune, S. (2008). *Introduction to discrete event systems* (2nd ed.). Springer.

Cassez, F., Dubreil, J., & Marchand, H. (2012). Synthesis of opaque systems with static and dynamic masks. *Formal Methods in System Design*, *40*(1), 88–115.

Chédor, S., Morvan, C., Pinchinat, S., & Marchand, H. (2015). Diagnosis and opacity problems for infinite state systems modeled by recursive tile systems. *Discrete Event Dynamic Systems*, *25*(1–2), 271–294.

Chen, J., Ibrahim, M., & Kumar, R. (2017). Quantification of secrecy in partially observed stochastic discrete event systems. *IEEE Transactions on Automation Science and Engineering*, *14*(1), 185–195.

Darondeau, P., Marchand, H., & Ricker, L. (2014). Enforcing opacity of regular predicates on modal transition systems. *Discrete Event Dynamic Systems*, *25*(1–2), 251–270.

Dubreil, J., Darondeau, P., & Marchand, H. (2010). Supervisory control for opacity. *IEEE Transactions on Automatic Control*, *55*(5), 1089–1100.

Falcone, Y., & Marchand, H. (2015). Enforcement and validation (at runtime) of various notions of opacity. *Discrete Event Dynamic Systems*, *25*(4), 531–570.

Jacob, R., Lesage, J.-J., & Faure, J.-M. (2016). Overview of discrete event systems opacity: Models, validation, and quantification. *Annual Reviews in Control*, *41*, 135–146.

Keroglou, C., & Hadjicostis, C. N. (2013). Initial state opacity in stochastic DES. In *18th IEEE conference on ETFA* (pp. 1–8).

Keroglou, C., & Hadjicostis, C. N. (2015). Detectability in stochastic discrete event systems. *Systems & Control Letters*, *84*, 21–26.

Keroglou, C., & Hadjicostis, C. N. (2017a). Probabilistic system opacity in discrete event systems. *Discrete Event Dynamics and Systms*, 1–26.

Keroglou, C., & Hadjicostis, C. N. (2017b). Verification of detectability in probabilistic finite automata. *Automatica*, *86*, 192–198.

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

Mullins, J., & Yeddes, M. (2014). Opacity with orwellian observers and intransitive non-interference. In *12th int. workshop on discrete event systems* (pp. 344–349).

Noori-Hosseini, M., Lennartson, B., & Hadjicostis, C. (2018). Compositional visible bisimulation abstraction applied to opacity verification. In *14th int. workshop on discrete event systems* (pp. 434–441).

Norris, J. R. (1998). *Markov chains*. Cambridge University Press.

Paoli, A., & Lin, F. (2012). Decentralized opacity of discrete event systems. In *ACC* (pp. 6083–6088).

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). Verification of infinite-step opacity and complexity considerations. *IEEE TAC*, *57*(5), 1265–1269.

Saboori, A., & Hadjicostis, C. N. (2013). Verification of initial-state opacity in security applications of discrete event systems. *Information Sciences*, *246*, 115–132.

Saboori, A., & Hadjicostis, C. N. (2014). Current-state opacity formulations in probabilistic finite automata. *IEEE TAC*, *59*(1), 120–133.

Takai, S., & Oka, Y. (2008). A formula for the supremal controllable and opaque sublanguage arising in supervisory control. *SICE Journal of Control Measurements & Systems Integration*, *1*(4), 307–311.

Tong, Y., Li, Z., Seatzu, C., & Giua, A. (2017). Decidability of opacity verification problems in labeled Petri net systems. *Automatica*, *80*, 48–53.

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.

Yin, X. (2017). Initial-state detectability of stochastic discrete-event systems with probabilistic sensor failures. *Automatca*, *80*, 127–134.

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.

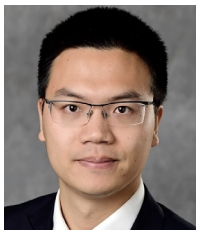
Yin, X., Li, Z., Wang, W., & Li, S. (2017). Infinite-step opacity of stochastic discrete-event systems. In *Asian control conf.* (pp. 102–107).

Zhang, B., Shu, S., & Lin, F. (2015). Maximum information release while ensuring opacity in discrete event systems. *IEEE Transactions on Automation Science and Engineering*, *12*(4), 1067–1079.

Zhang, K., Yin, X., & Zamani, M. (2018). Opacity of nondeterministic transition systems: A (bi)simulation relation approach. arXiv:1802.03321.

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, control of discrete-event systems, model-based fault diagnosis, security and their applications to cyber and cyber–physical systems. He received the Outstanding Reviewer Awards from Automatica, the IEEE Transactions on Automatic Control and the Journal of Discrete Event Dynamic Systems. He also 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.



Dr. Zhaojian Li received his B. Eng. degree from Nanjing University of Aeronautics and Astronautics in 2010. He obtained M.S. (2013) and Ph.D. (2015) in Aerospace Engineering (flight dynamics and control) at the University of Michigan, Ann Arbor. He worked as an algorithm engineer at General Motors from January 2016 to July 2017. Since August 2017, he has been an Assistant Professor in the department of Mechanical Engineering at Michigan State University. His research interests include Learning-based Control, Nonlinear and Complex Systems, and Robotics and Automated Vehicles. He was a recipient of the National Scholarship from China.

Weilin Wang received the M.S. degree in electrical engineering: systems, the M.S.E. degree in industrial engineering, and the Ph.D. degree in electrical engineering: systems from the University of Michigan, Ann Arbor, in 2003, 2006, and 2007, respectively. He is currently a Professor in the Department of Control Science and Engineering at University of Shanghai for Science and Technology. He is also with the Faculty of Engineering at Monash University.

Shaoyuan Li was born in Hebei, China, in 1965. He received the B.S. and M.S. degrees in automation from the Hebei University of Technology, Tianjin, China, in 1987 and 1992, respectively, and the Ph.D. degree from Nankai University, Tianjin, in 1997. Since 1997, he has been with the Department of Automation, Shanghai Jiao Tong University, Shanghai, China, where he is currently a Professor. His current research interests include model predictive control, dynamic system optimization, and cyber–physical systems.