2019 IEEE 58th Conference on Decision and Control (CDC)
Palais des Congrès et des Expositions Nice Acropolis
Nice, France, December 11-13, 2019

Abstraction-Based Synthesis of Opacity-Enforcing Controllers using Alternating Simulation Relations

Junyao Hou, Xiang Yin, Shaoyuan Li and Majid Zamani

Abstract— **Opacity is an important information-flow security property that captures the plausible deniability for some "secret" of a system. In this paper, we investigate the problem of synthesizing controllers that enforce opacity for labeled transition systems (LTS). Most of the existing works on synthesis of opacity-enforcing controllers are based on the original system model, which may contain a large number of states. To mitigate the complexity of the controller synthesis procedure, we propose an abstraction-based approach for controller synthesis. Specifically, we propose notion of *opacity-preserving alternating (bi)simulation relation* for the purpose of abstraction. We show that, if the abstract system is opacity-preserving alternatingly simulated by the original system which may be significantly smaller, then we can synthesize an opacity-enforcing controller based on the abstract system and then refine it back to a controller enforcing opacity of the original system. We investigate both initial-state opacity and infinite-step opacity. We also show the effectiveness of the proposed approach by a set of examples.**

I. Introduction

Opacity is an information-flow property arising in the security analysis of cyber-physical systems. As a confidentiality property, opacity describes the plausible deniability of the system's secret in the presence of an intruder, modeled as a passive observer, that is potentially malicious. The concept of opacity was originally introduced in the computer science literature [1]; later it was extended to discrete-event systems (DES) modeled by transition systems [2]. Due to its importance, opacity has drawn considerable attention in the DES literature in the past years; see, e.g., [3]–[11].

In many situations, the original system may not be opaque directly. Therefore, it is desirable to design a controller (or a supervisor) that restricts the behavior of the system such that the closed-loop system under control is opaque. This problem is referred to as the *opacity-enforcing control synthesis problem* and has also drawn many attentions in the literature [12]–[15]. For example, in [12], the authors investigated the scenario where the intruder's observation is a subset of the controller's observation. In [15], the authors dropped the assumption on the observation of the intruder but assumed that the intruder does not know the implementation of the controller.

Most of the existing works on opacity-enforcing controller synthesis are based on the original model of the system. However, in many real-world applications, the size of the system is very large, which makes synthesizing controllers using original models computationally very challenging. Therefore, one must look into finite abstractions of such systems to mitigate the synthesis complexity. In the context of DES, several abstraction-based techniques have been proposed very recently for the purpose of *verifying* opacity; see, e.g., [16]–[21]. However, these abstraction techniques are not directly applicable for the purpose of synthesis. In [16], the authors proposed an abstraction-based approach for synthesizing edit functions to enforce opacity. However, an edit function can only change the external behavior of the system, while a controller can restrict the internal behavior of the system,

In this paper, we investigate the abstraction-based synthesis of opacity-enforcing controllers. Specifically, we use labeled transitions systems (LTSs), which is a fundamental model for the verification and control of hybrid systems, as the underlying model of the system. We employ the notion of *alternating (bi)simulation relation* to connect original systems with abstract ones. This notion has been shown to be very suitable for computing symbolic models of control systems [22], [23]. However, the standard definition of alternating (bi)simulation relation does not preserve opacity.

We propose a new concept of *opacity-preserving* alternating (bi)simulation relation. In particular, we show that, if there exists an opacity-preserving alternating simulation relation from the abstract system to the original one, then we can synthesize an opacity-enforcing controller based on the abstraction first and then refine it back to the original system to enforce opacity. We investigate this notion for both initial-state opacity and infinite-step opacity. Furthermore, we show that if there exists an opacity-preserving alternating *bisimulation* relation between the abstract system and the original one, then the opacity synthesis problem is solvable for the original system *if and only if* it is solvable for the abstract one.

II. System Model and Opacity

A. Preliminaries

In this paper, we consider a system modeled by a *labeled transition system* (LTS)

$$(X, X_0, U, \longrightarrow, Y, H),$$

where X is a finite set of states, $X_0 \subseteq X$ is the set of initial states, U is a finite set of inputs, $\longrightarrow \subseteq X \times U \times X$

This work was supported by the National Natural Science Foundation of China (61803259, 61833012).

J. Hou, X. Yin S. Li are with Department of Automation and Key Laboratory of System Control and Information Processing, Shanghai Jiao Tong University, Shanghai 200240, China. E-mail: {houjunyao,yinxiang,syli}@sjtu.edu.cn.

M. Zamani is with the Computer Science Department, University of Colorado Boulder, CO 80309, USA. M. Zamani is also with the Computer Science Department, Ludwig Maximilian University of Munich, 80539 Munich, Germany. Email: majid.zamani@colorado.edu.

978-1-7281-1397-5/19/$31.00 ©2019 IEEE

is a transition relation, Y is a finite set of outputs, and $H : X \to Y$ is an output mapping. For the sake of simplicity, we also denote a transition $(x, u, x') \in \longrightarrow$ by $x \xrightarrow{u} x'$, where we say that x' is a u-successor, or simply successor, of x. Note that the u-successor is not unique in general as the transition relation is non-deterministic. For each state $x \in X$, we denote by $U(x)$ the set of all inputs defined at x, i.e., $U(x) = \{u \in U : \exists x' \in X \text{ s.t. } x \xrightarrow{u} x'\}$. We denote by U^* the set of all finite sequences of inputs including the empty sequence ϵ; sets X^* and Y^* are defined analogously. We assume that the system investigated in this paper is non-blocking, i.e., $\forall x \in X : U(x) \neq \emptyset$.

A (finite) internal behavior generated from state $x \in X$ under input sequence $u_1 \cdots u_n \in U^*$ is a sequence of transition $x_0 \xrightarrow{u_1} x_1 \xrightarrow{u_2} \cdots \xrightarrow{u_n} x_n$, where $x_0 = x$. Note that the internal behavior generated may not be unique under the same input as the system is non-deterministic in general. Then the external behavior of the above internal behavior is a sequence of outputs $H(x_0)H(x_1)\cdots H(x_n) \in Y^*$. We only consider finite behaviors throughout the paper.

Let $T_a = (X_a, X_{a0}, U_a, \xrightarrow[a]{}, Y, H_a)$ and $T_b = (X_b, X_{b0}, U_b, \xrightarrow[b]{}, Y, H_b)$ be two LTSs with the same output set. Let $\mathcal{I} \subseteq X_a \times X_b \times U_a \times U_b$ be an interconnection relation such that $\forall (x_a, x_b) \in \pi_X(\mathcal{I}) : H(x_a) = H(x_b)$, where $\pi_X(\cdot)$ denotes the projection to $X_a \times X_b$. The composition of T_a and T_b with interconnection relation $\mathcal{I}$ is a new LTS

$$T_a \times_{\mathcal{I}} T_b = (X_{ab}, X_{ab0}, U_{ab}, \xrightarrow[ab]{}, Y, H_{ab}),$$

where $X_{ab} = \pi_X(\mathcal{I})$, $X_{ab0} = X_{ab} \cap (X_{a0} \times X_{b0})$, $U_{ab} = U_a \times U_b$, $H_{ab}((x_a, x_b)) = H_a(x_a) = H_b(x_b)$ and $(x_a, x_b) \xrightarrow[ab]{(u_a, u_b)} (x'_a, x'_b)$ if (i) $x_a \xrightarrow[a]{u_a} x'_a$; and (ii) $x_b \xrightarrow[b]{u_b} x'_b$; and (iii) $(x_a, x_b, u_a, u_b) \in \mathcal{I}$. The subscript $\mathcal{I}$ will be dropped when it is clear from the context.

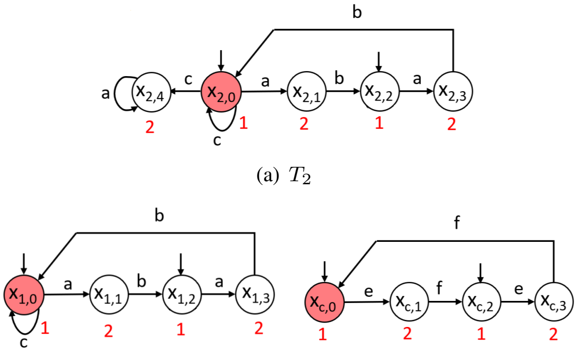
B. Opacity

In this paper, we consider internal behaviors as the information available to the system, while external behaviors are considered as the information available to the outside of the system (for example, an intruder). That is, the information of the system is released by the output mapping $H : X \to Y$.

In many applications, the system may have some "secret" that does not want to be revealed via the external behavior. We adopt a state-based formulation of secret. Specifically, we assume that $S \subseteq X$ is a set of *secret states*, and hereafter, we write an LTS in the form of $(X, X_0, S, U, \longrightarrow, Y, H)$. Then opacity captures the plausible deniability of the system's secret under the information leakage. In this paper, we discuss two important types of opacity called initial-state opacity and infinite-step opacity.

Definition 1: Let $T = (X, X_0, S, U, \longrightarrow, Y, H)$ be an LTS with secret states. We say that T is

- *initial-state opaque* if for any $x_0 \in X_0 \cap S$ and finite sequence $x_0 \xrightarrow{u_1} x_1 \xrightarrow{u_2} \cdots \xrightarrow{u_n} x_n$, there exist $x'_0 \in X_0 \setminus S$ and a finite sequence $x'_0 \xrightarrow{u'_1} x'_1 \xrightarrow{u'_2} \cdots \xrightarrow{u'_n} x'_n$ such that $H(x_i) = H(x'_i), \forall i = 0, 1, \ldots, n$;
- *infinite-step opaque* if for any $x_0 \in X_0$ and finite sequence $x_0 \xrightarrow{u_1} x_1 \xrightarrow{u_2} \cdots \xrightarrow{u_n} x_n$ such that $x_k \in S$ for some $k = 0, \ldots n$, there exist $x'_0 \in X_0$ and finite sequence $x'_0 \xrightarrow{u'_1} x'_1 \xrightarrow{u'_2} \cdots \xrightarrow{u'_n} x'_n$ such that $H(x_i) = H(x'_i), \forall i = 0, 1, \ldots, n$ and $x'_k \notin S$.



(a) T_2

(b) T_1

(c) T_c for T_1

Fig. 1. $T_1 \preceq_{AS}^{IOP} T_2$. Both T_1 and T_2 are not initial-state opaque, but both can be enforced to be opaque.

Intuitively, initial-state opacity requires that the intruder can never determine that the system was initialized from a secret state based on the external behavior, while infinite-step opacity requires that the intruder can never determine that the system was at a secret state at any specific instant. Clearly, infinite-step opacity is strictly stronger than initial-state opacity.

Example 1: Let us consider LTS $T_1 = (X_1, X_{1,0}, S_1, U_1, \xrightarrow[1]{}, Y, H_1)$ shown in Fig. 1(b), where $X_1 = \{x_{1,0}, x_{1,1}, x_{1,2}, x_{1,3}\}, U_1 = \{a, b, c\}$ and $Y = \{1, 2\}$. The output mapping is specified by the value associated with each state. An initial state is marked by a sourceless arrow and we mark a secret state by red color, i.e., $X_{1,0} = \{x_{1,0}, x_{1,2}\}$ and $S_1 = \{x_{1,0}\}$.

Clearly, system T_1 is not initial-state opaque. For example, if output 1 occurs twice successively, then the intruder knows immediately that the system must start from secret state $x_{1,0}$. Therefore, T_1 is also not infinite-step opaque. ■

III. Opacity-Enforcing Controllers

A. Feedback Composition

When an LTS T does not satisfy some property, e.g., opacity, we can synthesize a controller for T such that the closed-loop system meets the specification. There are several (equivalent) definitions for controllers in the literature. In this paper, we adopt the definition of controller in [24], in which a controller is considered also as a system that is composable to the original one with *alternating simulation relation*.

Definition 2: (Alternating Simulation Relation) Let $T_a = (X_a, X_{a0}, U_a, \xrightarrow[a]{}, Y, H_a)$ and $T_b = (X_b, X_{b0}, U_b, \xrightarrow[b]{}, Y, H_b)$ be two LTSs with the same output set. A relation $R \subseteq X_a \times X_b$ is said to be an alternating simulation relation from T_a to T_b if the following conditions hold:

1) $\forall x_{a0} \in X_{a0}, \exists x_{b0} \in X_{b0} : (x_{a0}, x_{b0}) \in R$;

2) $\forall (x_a, x_b) \in R : H_a(x_a) = H_b(x_b)$;

3) $\forall (x_a, x_b) \in R, \forall u_a \in U_a(x_a), \exists u_b \in U_b(x_b)$ such that $\forall x_b \xrightarrow{u_b} x_b', \exists x_a \xrightarrow{u_a} x_a' : (x_a', x_b') \in R$.

We say that T_a is alternatingly simulated by T_b (or T_b alternatingly simulates T_a), denoted by $T_a \preceq_{AS} T_b$, if there exists an alternating simulation relation from T_a to T_b.

An alternating simulation relation $R \subseteq X_a \times X_b$ from T_a to T_b can also be extended to an interconnection relation $R^e \subseteq X_a \times X_b \times U_a \times U_b$ defined by: $(x_a, x_b, u_a, u_b) \in R^e$ if

(i) $(x_a, x_b) \in R$; and

(ii) $u_a \in U_a(x_a), u_b \in U_b(x_b)$; and

(iii) $\forall x_b \xrightarrow{u_b} x_b', \exists x_a \xrightarrow{u_a} x_a' : (x_a', x_b') \in R$.

Intuitively, R^e explicitly specifies which inputs we need to choose in order to maintain the alternating simulation relation.

If $T_a \preceq_{AS} T_b$, then we can consider T_a as a *controller* for T_b if $T_a \preceq_{AS} T_b$. Specifically, assume that $T_a \times T_b$ is at state $(x_a, x_b) \in R$. First, controller T_a offers an input $u_a \in U_a(x_a)$; this input is then transferred to T_b as a matching input $u_b \in U_b(x_b)$ via the interconnection relation R^e. Due to the non-determinism, T_b may go to any successor of u_b. Once T_b measures the successor state, T_a will update its state by matching the successor in T_b, and then offer a new input, and so forth. The above discussion is summarize by the following definition.

Definition 3: (Feedback Composition) An LTS T_c is said to be *feedback composable* with an LTS T if there exists an alternating simulation relation R from T_c to T. When T_c is feedback composable with T, the feedback composition of T_c and T is given by

$$T_c \times_{\mathcal{F}} T = (X_c \times X, X_{c0} \times X, U_c \times U, \xrightarrow[\mathcal{F}]{}, Y, H),$$

where the interconnection relation $\mathcal{F} = R^e$ is the extended alternating simulation relation. For the sake of simplicity, we still denote the output mapping by H as we always have $H_{\times_{\mathcal{F}}}((x_c, x)) = H(x)$.

Therefore, we refer to T_c as a *controller* for T if it is feedback composable. Furthermore, we require that $T_c \times_{\mathcal{F}} T$ should also be non-blocking. In this paper, our interest is to design a controller T_c for T such that $T_c \times_{\mathcal{F}} T$ is opaque. More specifically, we say that T_c *enforces initial-state opacity* for T if for any $(x_{c0}, x_0) \in X_{c0} \times (X_0 \cap S)$ and any finite sequence

$$(x_{0c}, x_0) \xrightarrow[\mathcal{F}]{(u_{1c}, u_1)} (x_{1c}, x_1) \xrightarrow[\mathcal{F}]{(u_{2c}, u_2)} \cdots \xrightarrow[\mathcal{F}]{(u_{nc}, u_n)} (x_{nc}, x_n),$$

there exist $x_0' \in X_0 \setminus S$ and a finite sequence

$$x_0' \xrightarrow{u_1'} x_1' \xrightarrow{u_2'} \cdots \xrightarrow{u_n'} x_n'$$

such that $H(x_i) = H(x_i'), \forall i = 0, 1, \ldots, n$. That T_c enforces infinite-step opacity is defined analogously. Then the *Opacity Enforcing Control Problem* requires to synthesize a controller T_c for T that enforces (initial-state or infinite-step) opacity.

B. Abstraction and Controller Refinement

The opacity-enforcing control problem considered has already been solved in the literature [12]–[15]; the reader is referred to these works for detailed synthesis algorithms. However, the computation complexity is still a big burden when the size of the original system is very large. To mitigate complexity, a natural approach is to construct an abstract system for the purpose of synthesis.

It is well-known that alternating simulation relations can serve as an abstraction relation for control synthesis by capturing control non-determinism [22]. This is formalized by the following result.

Proposition 1: [24] Let T_1, T_2 and T_c be systems with the same output set. Suppose that T_c is feedback composable with T_1 under alternating simulation relation $R_{c1} \subseteq X_c \times X_1$. If there exists an alternating simulation relation $R_{12} \subseteq X_1 \times X_2$ from T_1 to T_2, then $T_c \times_{\mathcal{F}} T_1$ is feedback composable with T_2 under alternating simulation relation defined by:

$$R_{(c1)2} = \left\{ \begin{array}{c|c} ((x_c, x_1), x_2) & (x_c, x_1) \in R_{c1} \\ \in (X_c \times X_1) \times X_2 & \text{and } \ (x_1, x_2) \in R_{12} \end{array} \right\} \tag{1}$$

Proposition 1 essentially says that, if $T_1 \preceq_{AS} T_2$, then any controller T_c designed for T_1 can be *refined* to a controller for T_2. We denote by $T_{ref} = T_c \times_{\mathcal{F}} T_1$ the refined controller with the interconnection relation defined in Equation (1). In particular, the refined controller has the following property

$$T_{ref} \times T_2 \preceq_S T_c \times T_1, \tag{2}$$

where $\preceq_S$ denotes the standard simulation relation [24]. Therefore, we say that T_1 is an *abstraction* of T_2, if $T_1 \preceq_{AS} T_2$. Throughout the paper, we denote the original system by T_2 and the abstract system is denoted by T_1.

However, alternating simulation relation does not necessarily preserve the enforcement of opacity. Therefore, we are interested in finding a new type of *opacity preserving* alternating simulation relation, so that it can be applied to the opacity-enforcing control problem.

IV. Initial-State Opacity Preserving Alternating Simulation Relation

In this section, we propose the *initial-state opacity preserving* (InitSOP) alternating simulation relation. Specifically, we want that the new relation from T_1 to T_2 satisfies the followings requirements:

- it is still an alternating simulation relation, so that control non-determinism can be captured in the abstraction;
- enforcing opacity for T_1 implies the enforcement of opacity for T_2 after controller refinement.

To this end, we propose the following definition.

Definition 4: (InitSOP Alternating Simulation Relation) Let T_1, T_2 be two LTSs, where $T_i = (X_i, X_{i,0}, S_i, U_i, \xrightarrow[i]{}, Y, H_i), i = 1, 2$. A relation $R \subseteq X_1 \times X_2$ is said to be an InitSOP alternating simulation relation from T_1 to T_2 if

1) a) $\forall x_{1,0} \in X_{1,0}, \exists x_{2,0} \in X_{2,0} : (x_{1,0}, x_{2,0}) \in R$;

b) $\forall x_{1,0} \in X_{1,0} \setminus S_1, \exists x_{2,0} \in X_{2,0} \setminus S_2 : (x_{1,0}, x_{2,0}) \in R$;

c) $\forall x_{2,0}\in X_{2,0}\cap S_2, \exists x_{1,0}\in X_{1,0}\cap S_1 : (x_{1,0}, x_{2,0})\in R$;

2) $\forall (x_1, x_2) \in R : H_1(x_1) = H_2(x_2)$;

3) for any $(x_1, x_2) \in R$, we have

a) $\forall u_1 \in U_1(x_1), \exists u_2 \in U_2(x_2), \forall x_2 \xrightarrow{u_2} x_2', \exists x_1 \xrightarrow{u_1} x_1'$ such that $(x_1', x_2') \in R$;

b) $\forall x_1 \xrightarrow{u_1} x_1', \exists x_2 \xrightarrow{u_2} x_2'$ such that $(x_1', x_2') \in R$.

We say that T_1 is InitSOP alternatingly simulated by T_2 (or T_2 InitSOP alternatingly simulates T_1), denoted by $T_1 \preceq_{AS}^{IOP} T_2$, if there exists an InitSOP alternating simulation relation from T_1 to T_2.

Note that an InitSOP alternating simulation relation is still an alternating simulation relation, which makes controller refinement still possible. The main differences between InitSOP alternating simulation relation and the standard alternating simulation relation are:

(i) We further specify explicitly what initial states can be related in terms of secret states; and

(ii) Another simulation type condition for the other direction, i.e., condition 3)-b), is added in addition to the original alternating simulation type condition.

The following result shows that InitSOP alternating simulation relation indeed satisfies the requirements for the purpose of opacity-enforcing control synthesis.

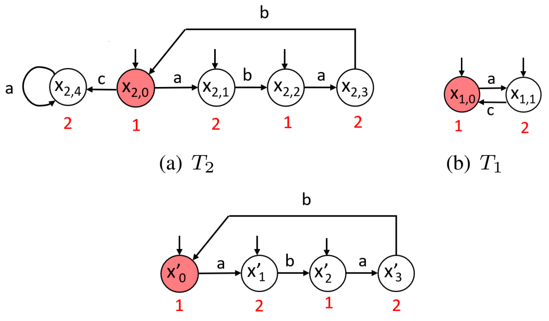
Theorem 1: Let T_1 and T_2 be two LTSs, where $T_i = (X_i, X_{i,0}, S_i, U_i, \xrightarrow[i]{}, Y, H_i), i = 1, 2$, and suppose that $T_1 \preceq_{AS}^{IOP} T_2$. Then for any controller T_c that enforces initial-state opacity for the abstract system T_1, the refined controller $T_{ref} = T_c \times_{\mathcal{F}} T_1$ also enforces initial-state opacity for the original system T_2.

In essence, the role of InitSOP alternating simulation relation is to build a "bridge" between the abstract system and the original one. The fact that it is still an alternating simulation relation ensures that any control input in the abstract system can be refined to the original system by matching an input pair in the relation. Moreover, Theorem 1 shows that the refined controller T_{ref} can still enforce opacity for the original system if T_c enforces opacity for the abstract system.

Example 2: Let us consider LTSs T_1 and T_2 shown in Figures 1(b) and 1(a), respectively, where both T_1 and T_2 are not initial-state opaque. We consider the following InitSOP alternating simulation relation from T_1 to T_2

$$R_{12} = \{(x_{1,0}, x_{2,0}), (x_{1,2}, x_{2,2}), (x_{1,1}, x_{2,1}), (x_{1,3}, x_{2,3})\}.$$

To enforce opacity for T_1, we can design a controller T_c shown in Figure 1(c) that is feedback composable to T_1 with alternating simulation relation $R_{c1} = \{(x_{c,0}, x_{1,0}), (x_{c,1}, x_{1,1}), (x_{c,2}, x_{1,2}), (x_{c,3}, x_{1,3})\}$. Clearly, we see that internal behavior $x_{1,0}x_{1,0}\cdots$ that reveals the secret is excluded under the control of T_c. The composed system $T_{ref} = T_c \times T_1$ is isomorphic (by renaming state names) to T_c. By Theorem 1, we can use T_{ref} as the refined controller to enforce initial-state opacity for the original system T_2. Intuitively, the refined controller will exclude both transitions $x_{2,0} \xrightarrow{c} x_{2,0}$ and $x_{2,0} \xrightarrow{c} x_{2,4}$ in T_2. ■



(a) T_2

(b) T_1

(c) controller T' for T_2

Fig. 2. $T_1 \preceq_{AS}^{IOP} T_2$, where T_1 cannot be enforced to be initial-state opaque but T_2 can be enforced to be initial-state opaque.

InitSOP alternating simulation relation ensures that, if there exists a controller T_c enforcing opacity for T_1, then it can be refined to enforce opacity for T_2. However, we cannot conclude that the original system cannot be enforced to be opaque if the opacity enforcement problem is unsolvable for the abstract system as illustrated by the following example.

Example 3: Let us consider LTSs T_1 and T_2 shown in Figure 2(b) and 2(a), respectively, where $T_1 \preceq_{AS}^{IOP} T_2$ and both T_1 and T_2 are not initial-state opaque. Clearly, for T_1, there exists no controller that can enforce initial-state opacity as the control choice at each state in unique. However, we can still design a controller T' shown in Figure 2(c), which is feedback composable to T_2 under relation $R = \{(x_0', x_{2,0}), (x_1', x_{2,1}), (x_2', x_{2,2}), (x_3', x_{2,3})\}$, to enforce initial-state opacity for T_2 . ■

The above example reveals that T_1 may excessively abstract T_2 so that some information of the original system is lost. Therefore, a stronger and symmetric version of Definition 4 is proposed as follows.

Definition 5: (InitSOP Alternating Bisimulation Relation) Let T_1, T_2 be two LTSs, where $T_i = (X_i, X_{i,0}, S_i, U_i, \xrightarrow[i]{}, Y, H_i), i = 1, 2$. A relation $R \subseteq X_1 \times X_2$ is said to be an InitSOP alternating bisimulation relation between T_1 and T_2 if

1) a) $\forall x_{1,0}\in X_{1,0}\setminus S_1, \exists x_{2,0}\in X_{2,0}\setminus S_2 : (x_{1,0}, x_{2,0})\in R$;

b) $\forall x_{1,0}\in X_{1,0}\cap S_1, \exists x_{2,0}\in X_{2,0}\cap S_2 : (x_{1,0}, x_{2,0})\in R$;

c) $\forall x_{2,0}\in X_{2,0}\setminus S_2, \exists x_{1,0}\in X_{1,0}\setminus S_1 : (x_{1,0}, x_{2,0})\in R$;

d) $\forall x_{2,0}\in X_{2,0}\cap S_2, \exists x_{1,0}\in X_{1,0}\cap S_1 : (x_{1,0}, x_{2,0})\in R$;

2) $\forall (x_1, x_2) \in R : H_1(x_1) = H_2(x_2)$;

3) for any $(x_1, x_2) \in R$, we have

a) $\forall u_1 \in U_1(x_1), \exists u_2 \in U_2(x_2), \forall x_2 \xrightarrow{u_2} x_2', \exists x_1 \xrightarrow{u_1} x_1'$ such that $(x_1', x_2') \in R$;

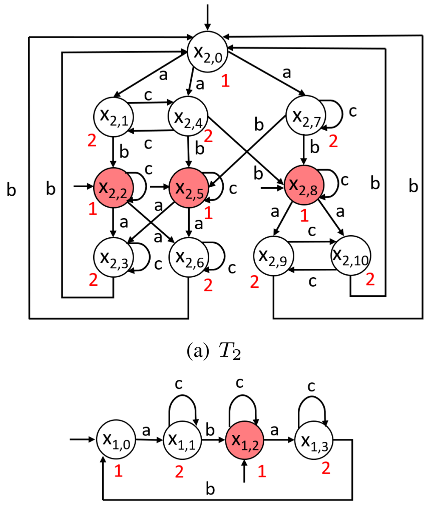
b) $\forall u_2 \in U_2(x_2), \exists u_1 \in U_1(x_1), \forall x_1 \xrightarrow{u_1} x_1', \exists x_2 \xrightarrow{u_2} x_2'$ such that $(x_1', x_2') \in R$;

c) $\forall x_1 \xrightarrow{u_1} x_1', \exists x_2 \xrightarrow{u_2} x_2'$ such that $(x_1', x_2') \in R$;

d) $\forall x_2 \xrightarrow{u_2} x_2', \exists x_1 \xrightarrow{u_1} x_1'$ such that $(x_1', x_2') \in R$.

We say that T_1 is InitSOP alternatingly bisimular to T_2, denoted by $T_1 \cong_{AS}^{IOP} T_2$, if there exists an InitSOP alternating bisimulation relation between T_1 and T_2.

Then we have the following result that follows directly from Theorem 1. It shows that, if $T_1 \cong_{AS}^{IOP} T_2$, then the



(a) T_2

(b) T_1

Fig. 3. $T_1 \cong_{AS}^{IOP} T_2$, where both T_1 and T_2 can be enforced to be initial-state opaque.

opacity enforcing control problem is solvable for T_2 *if and only if* it is solvable for T_1.

Theorem 2: Let T_1 and T_2 be two LTSs, where $T_i = (X_i, X_{i,0}, S_i, U_i, \xrightarrow[i]{}, Y, H_i), i = 1, 2$, and suppose that $T_1 \cong_{AS}^{IOP} T_2$. Then there exists a controller T_c that enforces initial-state opacity for T_1, if and only if, there exists a controller T_c' that enforces initial-state opacity for T_2.

We illustrate the notion of InitSOP alternating bisimulation relation by the following two examples. The first example shows that, although Definition 5 is a strong condition, it may still obtain a fairly succinct abstraction.

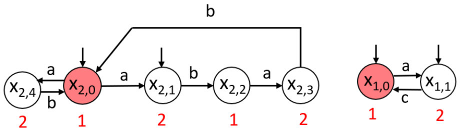
Example 4: Let us consider LTSs T_1 and T_2 shown in Figures 3(b) and 3(a), respectively, where $T_1 \cong_{AS}^{IOP} T_2$ and both T_1 and T_2 are not initial-state opaque. Specifically, one can readily verify that the following relation is an InitSOP alternating bisimulation relation from T_1 to T_2

$$R_{12} = \left\{ \begin{array}{c} (x_{1,0}, x_{2,0}), (x_{1,1}, x_{2,1}), (x_{1,1}, x_{2,4}), (x_{1,1}, x_{2,7}), \\ (x_{1,2}, x_{2,2}), (x_{1,2}, x_{2,5}), (x_{1,2}, x_{2,8}), (x_{1,3}, x_{2,3}), \\ (x_{1,3}, x_{2,6}), (x_{1,3}, x_{2,9}), (x_{1,3}, x_{2,10})) \end{array} \right\}.$$

For T_1, it is easy to design a controller T_c enforcing initial-state opacity, e.g., by eliminating the transition $x_{1,2} \xrightarrow{c} x_{1,2}$. Hence, we can refine this controller to enforce initial-state opacity for the original system T_2. Note that the original system T_2 is more complicated with more states and transitions compared with T_1. This example shows, under the InitSOP alternating bisimulation relation, we may still earn a succinct abstraction. It also shows that the relation is effective in the nondeterministic systems that we are interested in. ■

The next example shows that we can conclude that the opacity enforcing control problem is unsolvable for the original system if it is unsolvable for the abstract system.

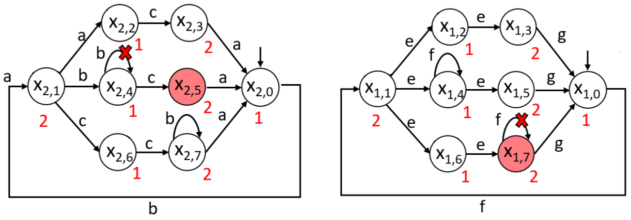
Example 5: Let us consider LTSs T_1 and T_2 shown in Figures 4(b) and 4(a), respectively, where $T_1 \cong_{AS}^{IOP} T_2$ and both T_1 and T_2 are not initial-state opaque. Specifically, one can readily verify that $R_{12} = \{(x_{1,0}, x_{2,0}), (x_{1,1}, x_{2,1}), (x_{1,0}, x_{2,2}), (x_{1,1}, x_{2,3}), (x_{1,1}, x_{2,4})\}$ is an InitSOP alternating bisimulation relation between T_1 and T_2. As we have discussed earlier in Example 3, T_1 cannot be enforced to be initial-state opaque. Therefore, by Theorem 2, we know that the original system T_2 also cannot be enforced to be initial-state opaque. ■



(a) T_2

(b) T_1

Fig. 4. $T_1 \cong_{AS}^{IOP} T_2$, where both T_1 and T_2 cannot be enforced to be initial-state opaque.



(a) T_2

(b) T_1

Fig. 5. $T_1 \cong_{AS}^{IOP} T_2$, where both T_1 and T_2 are not infinite-step opaque, and arbitrarily one of their controllers cannot be refined to enforce each other infinite-step opaque.

V. Infinite Opacity-preserving Relation

In the previous section, we have discussed how to modify the standard alternating (bi)simulation to preserve initial-state opacity. In this section, we further extend our result to the case of infinite-step opacity.

First, we show by the following example that InitSOP alternating (bi)simulation does not preserve infinite-step opacity enforcement after control refinement.

Example 6: Let us consider LTSs T_1 and T_2 shown in Figures 5(b) and 5(a), respectively. We have $T_1 \cong_{AS}^{IOP} T_2$ under InitSOP alternating bisimulation relation $R_{1,2} = \{(x_{1,0}, x_{2,0}), (x_{1,1}, x_{2,1}), (x_{1,2}, x_{2,2}), (x_{1,3}, x_{2,3}), (x_{1,4}, x_{2,4}), (x_{1,5}, x_{2,5}), (x_{1,6}, x_{2,6}), (x_{1,7}, x_{2,7})\}$.

Note that neither of them are infinite-step opaque. For example, if the self-loop transition at state $x_{1,7}$ occurs more than once, then we knows for sure that T_1 is/was at the secret state. To enforce infinite-step opacity, we can design a controller eliminating transition $x_{1,7} \xrightarrow{f} x_{1,7}$ for T_1. The refinement of this controller will then eliminate transition $x_{2,7} \xrightarrow{b} x_{2,7}$ for T_2. However, this does not enforces infinite-step opacity for T_2 as transition $x_{2,4} \xrightarrow{b} x_{2,4}$ violates infinite-step opacity in T_2. Similarly, we can show that a controller that enforces infinite-step opacity for T_2 does not enforce infinite-step opacity for T_1 after refinement. ■

Therefore, in order to handle infinite-step opacity, we further strengthen the definition of InitSOP alternating simulation relation to *infinite-step opacity preserving* (InfSOP) alternating simulation relation as follows.

Definition 6: (InfSOP Alternating Simulation Relation) Let T_1, T_2 be two LTSs, where $T_i = (X_i, X_{i,0}, S_i, U_i, \xrightarrow[i]{}, Y, H_i), i = 1, 2$. A relation $R \subseteq X_1 \times X_2$ is said to be an InfSOP alternating simulation relation from T_1 to T_2 if

1) a) $\forall x_{1,0} \in X_{1,0}, \exists x_{2,0} \in X_{2,0} : (x_{1,0}, x_{2,0}) \in R$;
   b) $\forall x_{1,0} \in X_{1,0} \setminus S_1, \exists x_{2,0} \in X_{2,0} \setminus S_2 : (x_{1,0}, x_{2,0}) \in R$;
   c) $\forall x_{2,0} \in X_{2,0} \cap S_2, \exists x_{1,0} \in X_{1,0} \cap S_2 : (x_{1,0}, x_{2,0}) \in R$;
2) $\forall (x_1, x_2) \in R : H_1(x_1) = H_2(x_2)$;
3) for any $(x_1, x_2) \in R$,
   a) $\forall u_1 \in U_1(x_1), \exists u_2 \in U_2(x_2), \forall x_2 \xrightarrow{u_2} x_2' \in S_2$, $\exists x_1 \xrightarrow{u_1} x_1' \in S_1$ such that $(x_1', x_2') \in R$;
   b) $\forall u_1 \in U_1(x_1), \exists u_2 \in U_2(x_2) : \forall x_2 \xrightarrow{u_2} x_2' \in X_2 \setminus S_2$, $\exists x_1 \xrightarrow{u_1} x_1' \in X_1 \setminus S_1$ such that $(x_1', x_2') \in R$;
   c) $\forall x_1 \xrightarrow{u_1} x_1' \in S_1, \exists x_2 \xrightarrow{u_2} x_2' \in S_2$ such that $(x_1', x_2') \in R$;
   d) $\forall x_1 \xrightarrow{u_1} x_1' \in X_1 \setminus S_1, \exists x_2 \xrightarrow{u_2} x_2' \in X_2 \setminus S_2$ such that $(x_1', x_2') \in R$.

We say that T_1 is InfSOP alternatingly simulated by T_2 (or T_2 InfSOP alternatingly simulates T_1), denoted by $T_1 \preceq_{AS}^{IfOP} T_2$, if there exists an InfSOP alternating simulation relation from T_1 to T_2.

Intuitively, Definition 6 strengthen Definition 4 by further specifying what transitions can be matched in the (alternating) simulation type conditions, i.e., condition 3). Particularly, conditions 3)-a) and 3)-b) in Definition 6 together imply condition 3)-a) in Definition 4. Therefore, an InfSOP alternating simulation relation is still an alternating simulation relation. Moreover, it further requires that for any transition that goes to a secret state (resp. non-secret state) in T_1, its matching transition should also go to a secret state (resp. non-secret state) in T_1. The same requirement for conditions 3)-c) and 3)-d) in Definition 6, which together imply condition 3)-b) in Definition 4.

The following theorem shows that InfSOP alternating simulation relation indeed preserves infinite-step opacity.

Theorem 3: Let T_1, T_2 be two LTSs, where $T_i = (X_i, X_{i,0}, S_i, U_i, \xrightarrow[i]{}, Y, H_i), i = 1, 2$, and suppose that $T_1 \preceq_{AS}^{IfOP} T_2$. Then for any controller T_c that enforces infinite-step opacity for the abstract system T_1, the refined controller $T_{ref} = T_c \times_{\mathcal{F}} T_1$ also enforces infinite-step opacity for the original system T_2.

Similar to the case of InitSOP alternating bisimulation relation, we can also define InfSOP alternating bisimulation relation by symmetrizing Definition 6 to preserve infinite-step opacity for both directions.

VI. Conclusion

In this paper, we investigated abstraction-based synthesis of opacity-enforcing controllers using alternating simulation relations for labeled transition systems. We proposed a new concept called opacity-preserving alternating (bi)simulation relation that preserves opacity during the abstraction. We investigated this concept for both initial-state opacity and infinite-step opacity. Using the proposed approach, one can first synthesize an opacity-enforcing controller based on the abstract system and then refine it back to controller enforcing opacity of the original system.

References

[1] L. Mazaré, "Using unification for opacity properties," in *Workshop on Issues in the Theory of Security*, vol. 4, pp. 165–176, 2004.

[2] J. Bryans, M. Koutny, L. Mazaré, and P. Ryan, "Opacity generalised to transition systems," *Internationa Journal of Information Security*, vol. 7, no. 6, pp. 421–435, 2008.

[3] F. Lin, "Opacity of discrete event systems and its applications," *Automatica*, vol. 47, no. 3, pp. 496–503, 2011.

[4] A. Saboori and C. Hadjicostis, "Verification of infinite-step opacity and complexity considerations," *IEEE Trans. Automatic Control*, vol. 57, no. 5, pp. 1265–1269, 2012.

[5] Y. Ji, X. Yin, and S. Lafortune, "Opacity enforcement using nondeterministic publicly-known edit functions," *IEEE Transactions on Automatic Control*, 2019.

[6] X. Yin and S. Lafortune, "A new approach for the verification of infinite-step and K-step opacity using two-way observers," *Automatica*, vol. 80, pp. 162–171, 2017.

[7] S. Chédor, C. Morvan, S. Pinchinat, and H. Marchand, "Diagnosis and opacity problems for infinite state systems modeled by recursive tile systems," *Discrete Event Dynamic Systems: Theory & Appllications*, vol. 25, no. 1-2, pp. 271–294, 2015.

[8] R. Julio Barcelos and J. Basilio, "Enforcing current-state opacity through shuffle in event observations," in *14th Int. Workshop on Discrete Event Systems*, pp. 106–111, 2018.

[9] X. Yin and S. Li, "Synthesis of dynamic masks for infinite-step opacity," *IEEE Trans. Automatic Control*, 2019.

[10] C. Keroglou and C. Hadjicostis, "Probabilistic system opacity in discrete event systems," *Discrete Event Dynamic Systems: Theory & Appllications*, pp. 1–26, 2017.

[11] X. Yin, Z. Li, W. Wang, and S. Li, "Infinite-step opacity and K-step opacity of stochastic discrete-event systems," *Automatica*, vol. 99, pp. 266–274, 2019.

[12] J. Dubreil, P. Darondeau, and H. Marchand, "Supervisory control for opacity," *IEEE Trans. Automatic Control*, vol. 55, no. 5, pp. 1089–1100, 2010.

[13] S. Takai and Y. Oka, "A formula for the supremal controllable and opaque sublanguage arising in supervisory control," *SICE J. Control, Measu. & Syst. Integration*, vol. 1, no. 4, pp. 307–311, 2008.

[14] X. Yin and S. Lafortune, "A uniform approach for synthesizing property-enforcing supervisors for partially-observed discrete-event systems," *IEEE Trans. Automatic Control*, vol. 61, no. 8, pp. 2140–2154, 2016.

[15] Y. Tong, Z. Li, C. Seatzu, and A. Giua, "Current-state opacity enforcement in discrete event systems under incomparable observations," *Discrete Event Dynamic Systems: Theory & Appllications*, vol. 28, no. 2, pp. 161–182, 2018.

[16] S. Mohajerani, Y. Ji, and S. Lafortune, "Efficient synthesis of edit functions for opacity enforcement using bisimulation-based abstractions," in *IEEE CDC*, pp. 4849–4854, 2018.

[17] M. Noori-Hosseini, B. Lennartson, and C. Hadjicostis, "Incremental observer reduction applied to opacity verification and synthesis." arXiv:1812.08083, 2018.

[18] K. Zhang, X. Yin, and M. Zamani, "Opacity of nondeterministic transition systems: A (bi) simulation relation approach," *IEEE Transactions on Automatic Control*, 2019.

[19] X. Yin and M. Zamani, "On approximate opacity of cyber-physical systems," *arXiv preprint:1902.09411*, 2019.

[20] M. Noori-Hosseini, B. Lennartson, and C. Hadjicostis, "Compositional visible bisimulation abstraction applied to opacity verification," in *14th Int. Workshop on Discrete Event Systems*, pp. 434–441, 2018.

[21] B. Wu and H. Lin, "Privacy verification and enforcement via belief abstraction," *IEEE Cont. Sys. Letters*, vol. 2, no. 4, pp. 815–820, 2018.

[22] R. Alur, T. Henzinger, O. Kupferman, and M. Vardi, "Alternating refinement relations," in *International Conference on Concurrency Theory*, pp. 163–178, Springer, 1998.

[23] M. Zamani, G. Pola, M. Mazo, and P. Tabuada, "Symbolic models for nonlinear control systems without stability assumptions," *IEEE Trans. Automatic Control*, vol. 57, no. 7, pp. 1804–1809, 2012.

[24] P. Tabuada, *Verification and Control of Hybrid Systems: A Symbolic Approach.* Springer Science & Business Media, 2009.