2023 62nd IEEE Conference on Decision and Control (CDC)
December 13-15, 2023. Marina Bay Sands, Singapore

Synthesis of Opacity-Enforcing Winning Strategies Against Colluded Opponent

Chongyang Shi, Abhishek N. Kulkarni, Hazhar Rahmani, and Jie Fu*

Abstract— **This paper studies language-based opacity enforcement in a two-player, zero-sum game on a graph. In this game, player 1 (P1) wins if he can achieve a secret temporal goal described by the language of a finite automaton, no matter what strategy the opponent player 2 (P2) selects. In addition, P1 aims to win while making its goal opaque to a passive observer with imperfect information. However, P2 colludes with the observer to reveal P1's secret whenever P2 cannot prevent P1 from achieving its goal, and therefore, opacity must be enforced against P2. We show that a winning and opacity-enforcing strategy for P1 can be computed by reducing the problem to solving a reachability game augmented with the observer's belief states. Furthermore, if such a strategy does not exist, winning for P1 must entail the price of revealing his secret to the observer. We demonstrate our game-theoretic solution of opacity-enforcement control through a small illustrative example and in a robot motion planning problem.**

I. Introduction

Non-interference or opacity [3] is a security and privacy property that evaluates whether an observer (intruder) can infer a secret of a system by observing its behavior. The secret could be a language generated by the system (language-based opacity) [6], [12], [20]; a system state—such as initial, current, or final state—(state-based opacity) [2], [1], [16], [18], [20]; or some past state (k-step and infinite step opacity) [16], [17], [20]. Recent work by Wintenberg *et al.* [19] unifies those notions of opacity for discrete event systems and provides transformations from state-based notions of opacity to language-based notions.

This paper investigates a game-theoretic approach to enforce language-based opacity against an opponent who colludes with the passive observer. Our problem formulation is motivated by security applications of cyber-physical systems and robotics: Consider a robot (referred to as player 1/P1) that aims to accomplish a highly confidential task in a dynamic environment. Besides task completion, the robot must ensure a passive observer cannot infer if the task is accomplished, based on the observer's partial information. However, there are uncontrollable events or other agents in the operational environment. In the worst-case scenario, the uncontrollable environment (referred to as player 2/P2) may act in a way that forces the robot to reveal its secret. The question is, does the robot have an opacity-enforcing strategy to satisfy the secret surely, against a colluding P2?

This work was sponsored in part by the Army Research Office and was accomplished under Grant Number W911NF-22-1-0034 and W911NF2210166 and in part by NSF under grant No. 2144113.

The authors are with the Department of Electrical and Computer Engineering, University of Florida, Gainesville, FL 32605, USA. {c.shi, a.kulkarni2, h.rahmani, fujie}@ufl.edu .
*: the corresponding author.

Enforcing opacity in supervisory control has been extensively studied (see the survey by Lafortune *et al.* [11], the overview by Jacob *et al.* [10], and chapters in the book by Hadjicostis [8]). This work is to establish a connection of opacity-enforcing supervisory control with reactive synthesis for a subclass of games on graphs. In [21], the authors investigate a problem similar to ours where the supervisor enforces state-based opacity to an observer despite that the controller is public knowledge and there are uncontrollable events. They show that a non-deterministic supervisor is more powerful than a deterministic one for opacity-enforcing. In our formulation, the passive observer has no access to the controller design but the uncontrollable environment plays to reveal the secret if it is satisfied. We thus employ a game-theoretic approach between the controller P1 and its environment P2. Our game formulation is closely related to that by Hélouët *et al.* [9], who develop a game-theoretic controller that enforces state-based opacity against different types of attackers. They employ a turn-based safety game where the goal of the controller is to avoid reaching a state at which the attacker's belief is a subset of the secret states. The safety objective can be satisfied in two ways: Either a path/execution never reaches a secret state or a path that reaches a secret state but is observation-equivalent to a path that does not reach a secret state. Different from [9], in our game, P1 must satisfy the secret and ensure its opacity against all possible uncontrollable events in its environment (P2), given information leaking to the passive observer. In [4], the authors develop opacity-enforcement in a DES through masking events using a game-theoretic approach, where P1 controls which events to be masked/unobservable and P2 controls the next observable events and aims to reveal the secret to an observer. In our game, both P1 and P2 have controllable events/actions but P1 is not allowed to alter the observation function at run time. Thus, P1 must employ control actions to maintain secrecy to the observer.

We show that opacity-enforcing control can be formulated as a reachability game whose states augment the observer's belief. Based on the determinacy of turn-based games [7], [14], we prove that if, for a given state, P1 can satisfy the secret from but cannot enforce its opacity with probability one, then for any secret-satisfying (called winning) strategy of P1, any counter-strategy of P2 will surely reveal the secret of P1 to the observer. This fact holds regardless of whether P1 or P2 uses a deterministic/non-deterministic/randomized strategy. We illustrate the proposed synthesis method using a simple game on graph example and a robot motion planning problem in an adversarial environment.

II. Preliminaries and Problem Formulation

A. Preliminaries

Notations: Let X be a finite set. We use $\mathcal{D}(X)$ to denote the set of all probability distributions over X. For any distribution $d \in \mathcal{D}(X)$, the set of elements in X with non-zero probability under d is denoted $\mathsf{Supp}(d)$, that is, $\mathsf{Supp}(d) = \{x \in X \mid d(x) > 0\}$.

We model the interaction between a controllable agent, player 1 (P1), and an uncontrollable environment, player 2 (P2) using a turn-based game arena defined as follows.

Definition 1: The *transition system* (arena) of a two-player, turn-based, deterministic game on a graph is a tuple

$$G = \langle S := S_1 \cup S_2, A := A_1 \cup A_2, T, s_0, \mathcal{AP}, L \rangle$$

in which

- $S = S_1 \cup S_2$ is a finite set of states partitioned into S_1 and S_2 where at a state in S_1, P1 chooses an action, and at a state in S_2, P2 selects an action;
- $A = A_1 \cup A_2$ is the set of actions, where A_1 (resp., A_2) is the set of actions for P1 (resp., P2);
- $T : (S_1 \times A_1) \cup (S_2 \times A_2) \to S$ is a *deterministic* transition function that maps a state-action pair $(s, a) \in S_i \times A_i$ to a next state $s' \in S$ for all $i \in \{1, 2\}$;
- $s_0 \in S$ is the initial state;
- $\mathcal{AP}$ is the set of atomic propositions; and
- $L : S \to 2^{\mathcal{AP}}$ is the labeling function that maps a state to a set of atomic propositions that evaluate true at that state.

A finite play $\rho = s_0 a_0 s_1 a_1 s_2 \dots s_n$ is a sequence of interleaving states and players' actions such that $T(s_i, a_i) = s_{i+1}$ for all integers $0 \le i \le n-1$. The labeling of the play ρ, denoted $L(\rho)$, is defined as $L(\rho) = L(s_0)L(s_1)\dots L(s_n)$. That is, the labeling function omits the actions from the play and applies to states only. We use $\mathsf{Plays}(G) \subseteq (S \times A)^* S$ to denote the set of finite plays that can be generated from the game G and we use $\mathsf{PrefPlays}(G)$ to denote the set of prefixes[1] of the finite plays within $\mathsf{Plays}(G)$. A randomized (resp. deterministic) strategy of player i is a function $\pi_i : \mathsf{PrefPlays}(G) \to \mathcal{D}(A_i)$ (resp. $\pi_i : \mathsf{PrefPlays}(G) \to A_i$) that maps a prefix/history of a play into a distribution over actions (resp. a single action for player i). In our notation, $\mathsf{Plays}(\rho, \pi_1, \pi_2)$ is the set of possible plays that can be generated (with a non-zero probability) when P1 and P2 follow the strategy profile (π_1, π_2) starting from the play ρ. Formally, $\rho' \in \mathsf{Plays}(\rho, \pi_1, \pi_2)$ if and only if ρ is a prefix of ρ', i.e. , $\rho' = \rho \cdot s_k a_k s_{k+1} \dots s_n$ where if $s_j \in S_i$ then $a_j \in \mathsf{Supp}(\pi_i(\rho \cdot s_k a_k \dots s_j))$ and $T(s_j, a_j) = s_{j+1}$, for all $k \le j < n$ and $i \in \{1, 2\}$. For $i \in \{1, 2\}$, the set of all randomized strategies for player i is denoted Π_i.

P1' temporal objective: In the game arena, P1 (pronouns he/him/his) intends to achieve a temporal objective $\varphi \subseteq (2^{\mathcal{AP}})^*$, while P2 (pronouns she/her) aims to prevent P1 from achieving that objective.

[1]A word u is a prefix of word w if and only if there exists word v that $w = u \cdot v$, where $\cdot$ is the symbol for concatenation.

The set of words satisfying the temporal objective φ is equivalently represented by the language of a Deterministic Finite Automaton (DFA)[2].

Definition 2 (DFA): A DFA is a tuple $\mathcal{A} = (Q, \Sigma, \delta, \iota, F)$ in which (1) Q is the set of states; (2) Σ is the alphabet (set of input symbols); (3) $\delta : Q \times \Sigma \to Q$ is a deterministic transition function and is complete [3]; (4) ι is the initial state; and (5) $F \subseteq Q$ is the set of accepting states.

The transition function δ is extended as $\delta(q, \sigma \cdot w) = \delta(\delta(q, \sigma), w)$. A word $w = w_0 w_1 \dots w_n \in \Sigma^*$ is accepted by $\mathcal{A}$ if and only if $\delta(\iota, w) \in F$. The set of words accepted by $\mathcal{A}$ is called the language of $\mathcal{A}$, denoted by $\mathcal{L}(\mathcal{A})$. Formally, $\mathcal{L}(\mathcal{A}) = \{w \in \Sigma^* \mid \delta(q_0, w) \in F\}$.

For notation simplicity, let $\Sigma := 2^{\mathcal{AP}}$. The DFA that determines P1's winning plays is defined as follows.

Definition 3: Let $\mathcal{A} = (Q, \Sigma, \delta, \iota, F)$ be a DFA that specifies P1's temporal objective, i.e., $\varphi = \mathcal{L}(A)$. A play $\rho \in \mathsf{Plays}(G)$ is called *winning* for P1 if $L(\rho) \in \mathcal{L}(\mathcal{A})$. The set of all winning plays for P1 is denoted WPlays_1, i.e.,

$$\mathsf{WPlays}_1 = \{\rho \in \mathsf{Plays}(G) \mid L(\rho) \in \mathcal{L}(\mathcal{A})\}$$

In this paper, we assume the DFA that specifies the temporal goal has a specific structure.

Assumption 1: All the accepting states of the DFA specifying the temporal goal of P1 are absorbing, that is, for each $q \in F$ and $\sigma \in \Sigma$, $\delta(q, \sigma) = q$.

Definition 4: Given a history $\rho \in \mathsf{PrefPlays}(G)$, a strategy $\pi_1 \in \Pi_1$ is *winning* for P1 starting from ρ if for any $\pi_2 \in \Pi_2$, $\mathsf{Plays}(\rho, \pi_1, \pi_2) \subseteq \mathsf{WPlays}_1$. A strategy $\pi_2 \in \Pi_2$ is *winning* for P2 if for any $\pi_1 \in \Pi_1$, $\mathsf{Plays}(\rho, \pi_1, \pi_2) \cap \mathsf{WPlays}_1 = \varnothing$.

B. Problem formulation

In this two-player game with a temporal objective, we consider one single attacker of the system who is assumed to have full knowledge about the game arena but can only partially observe the plays in the game. The attacker's task is to infer if a play satisfies a secret property using his partial observations.

Definition 5 (Attacker's observation function): Given a finite set of observations Ω, the attacker's observation function is a function $O : S \times A \times S \to \Omega$ that maps each transition (s, a, s') to an observation the attacker receives when the transition system takes the transition (s, a, s'). The observation function naturally extends to plays: For each $\rho = s_0 a_0 s_1 a_1 \dots s_n \in \mathsf{Plays}(G)$, $O(\rho) = O(s_0, a_0, s_1) O(s_1, a_1, s_2) \dots O(s_{n-1}, a_{n-1}, s_n)$. Two plays ρ_1, ρ_2 are observation-equivalent if and only if $O(\rho_1) = O(\rho_2)$. We denote by $[\rho]$ the set of observation-equivalent plays of ρ.

Note this observation function is general enough to capture partial observations for both actions and states and also allows the action observations to be state-dependent. For

[2]For linear temporal logic over finite traces, the formula can be represented as DFAs. We omit the introduction of temporal logic for conciseness.

[3]For any $Q \times \Sigma$, $\delta(q, \sigma)$ is defined. An incomplete transition function can be completed by adding a sink state and redirecting all undefined transitions to that sink state.

example, we can capture the case when the attacker can observe P2's actions but not P1's actions.

The following information structure is considered:

1. P1 and P2 have perfect observations.
2. P1 and P2 know the temporal objective φ. However, the observer does not know φ.
3. The observation function of the attacker is common knowledge to all players (P1, P2, and the attacker).
4. There is no direct communication from P2 to the attacker. That is, P2 cannot inform the attacker if the secret is satisfied or not.

To define the attacker's objective, we adopt the language-based opacity [12].

Definition 6 (Language-based opacity): Given a temporal objective φ, called the secret of P1, the secret φ is *opaque* with respect to a play $\rho \in \mathsf{Plays}(G)$ if and only if 1) $L(\rho) \in \varphi$; and 2) there exists at least one observation-equivalent play $\rho' \in [\rho]$ such that $L(\rho') \notin \varphi$.

Using this notion of opacity, we can identify the set of winning and opaque plays as follows.

Definition 7: P1's *opacity-enforcing winning* plays is a set $\mathsf{OWPlays}_1$ of plays such that

$$\mathsf{OWPlays}_1 = \{\rho \in \mathsf{WPlays}_1 \mid \exists \rho' \in [\rho], L(\rho') \notin \mathcal{L}(\mathcal{A})\}.$$

P1's *secret-revealing winning* plays is a set $\mathsf{RWPlays}_1$ of plays such that

$$\mathsf{RWPlays}_1 = \{\rho \in \mathsf{WPlays}_1 \mid \forall \rho' \in [\rho], L(\rho') \in \mathcal{L}(\mathcal{A})\}$$

Clearly, $\mathsf{RWPlays}_1 \cap \mathsf{OWPlays}_1 = \varnothing$.

We consider that P2 is colluded with the attacker. In addition to P2's objective which is to prevent P1 from satisfying the objective φ, when φ is satisfied, P2 will react in a way trying to reveal this secret to the attacker.

Definition 8: Given a prefix $\rho_0 \in \mathsf{PrefPlays}(G)$, a strategy $\pi_1 \in \Pi_1$ is *opacity-enforcing winning* if for any strategy $\pi_2 \in \Pi_2$, $\mathsf{Plays}(\rho_0, \pi_1, \pi_2) \subseteq \mathsf{OWPlays}_1$. A strategy $\pi_1 \in \Pi_1$ is called *winning and positive opacity-enforcing* if for any strategy $\pi_2 \in \Pi_2$, $\mathsf{Plays}(\rho_0, \pi_1, \pi_2) \subseteq \mathsf{WPlays}_1$ and $\Pr^{\rho_0,\pi_1,\pi_2}(\rho \in \mathsf{OWPlays}_1) > 0$ and $\Pr^{\rho_0,\pi_1,\pi_2}(\rho \in \mathsf{WPlays}_1 \setminus \mathsf{OWPlays}_1) > 0$ where $\Pr^{\rho_0,\pi_1,\pi_2}$ is the probabilistic distribution induced by the strategy profile (π_1, π_2) given a prefix $\rho_0 \in \mathsf{PrefPlays}(G)$.

The notion of *positive opacity-enforcing* is related to *probabilistic opacity* and is relevant when the distribution of plays induced by players' strategies is described by a stochastic process.

Remark 1: The rationale to treat P2 and the attacker as separate entities is that P2 represents uncontrollable factors in the environment with which the system P1 interacts [13]. The goal of P1 is to satisfy the secret task φ while ensuring its opacity to an external observer (the attacker), regardless of all possible uncontrollable events in its environment (P2). Thus, P2 and the attacker cannot be treated as a single entity. Otherwise, the problem becomes trivial as P2 can share its observations and knowledge with the attacker.

Remark 2: Note that though the game arena is deterministic, we do not restrict players' strategies to be deterministic. If at least one player employs a randomized strategy against another using either deterministic/randomized strategy, then the joint strategy profile induces a probability measure over plays in G.

Problem 1: Given a turn-based, deterministic game arena $G = \langle S := S_1 \cup S_2, A := A_1 \cup A_2, T, s_0, \mathcal{AP}, L\rangle$, a DFA $\mathcal{A} = (Q, \Sigma, \delta, \iota, F)$ describing P1's secret temporal goal, and an observation function $O : S \times A \times S \to \Omega$ describing an attacker's partial observations, for a given prefix ρ_0 from which P1 can ensure a winning play in WPlays, compute an opacity-enforcing winning strategy for P1 if exists. Otherwise, determine whether P1 has a winning and positive opacity-enforcing strategy.

III. Main Results

In this section, we describe our algorithm to solve an opacity-enforcing winning strategy for P1 through the construction of a belief-augmented game. Then we prove the correctness and completeness of the algorithm. In the end, we prove that P1 does not have a winning and positive opacity-enforcing strategy in the turn-based deterministic game arena irrespective of whether deterministic/non-deterministic/randomized strategies are used.

Definition 9 (Belief-augmented game arena): Given the two-player, turn-based game arena $G = \langle S := S_1 \cup S_2, A := A_1 \cup A_2, T, s_0, \mathcal{AP}, L\rangle$, and a DFA $\mathcal{A} = (Q, 2^{\mathcal{AP}}, \delta, \iota, F)$ describing P1's secret objective φ, the belief-augmented game arena is a tuple

$$\mathcal{G} = (V := V_1 \cup V_2, A_1 \cup A_2, \Delta, v_0)$$

in which

- $V = S \times Q \times 2^{S\times Q}$ is the state space, partitioned into P1's states $V_1 = S_1 \times Q \times 2^{S\times Q}$ and P2's states $V_2 = S_2 \times Q \times 2^{S\times Q}$, where each state (s, q, b) includes a game state $s \in S$, an automaton state $q \in Q$, and a belief state $b \subseteq S \times Q$ of the attacker;
- $A_1 \cup A_2$ are the players' actions, same as in G;
- $\Delta : V \times (A_1 \cup A_2) \to V$ is the transition function such for a state $(s, q, b) \in V_i$, and an action $a \in A_i$,
$$\Delta((s, q, b), a) = (s', q', b')$$
where $s' = T(s, a)$, $q' = \delta(q, L(s'))$, and $b' = \{(s^o, q^o) \mid \exists(\bar{s}, \bar{q}) \in b, \exists \bar{a} \in A : \ O(\bar{s}, \bar{a}, s^o) = O(s, a, s')$ and $T(\bar{s}, \bar{a}) = s^o$ and $\delta(\bar{q}, L(s^o)) = q^o\}$
- $v_0 = (s_0, q_0, b_0)$ where $q_0 = \delta(\iota, L(s_0))$ and $b_0 = \{(s_0, q_0)\}$, is the initial belief of the observer.

Each state of this belief-augmented game is a tuple (s, q, b) in which s indicates the current state of the original arena, q indicates the state of the specification DFA, to which the DFA reaches by tracing the label sequence of the current play, and b is the belief of the attacker about the current state of the arena and the current state of the specification DFA. The belief update in the transition function Δ is based on subset construction (see [5] for a detailed construction). Intuitively, it computes all possible states that can be reached with some action from the current belief and removed from these states that are inconsistent with the observation. All definitions related to the game arena in Def. 1, including those of a play and a strategy, are applicable to the belief-augmented game arena $\mathcal{G}$.

We can establish a one-to-one mapping $\mathfrak{R} : \mathsf{Plays}(G) \to \mathsf{Plays}(\mathcal{G})$ between plays in the original game G and plays in the belief-augmented game $\mathcal{G}$ as follows: For a play $\rho = s_0 a_0 s_1 a_1 \dots s_n \in \mathsf{Plays}(G)$, there is a unique play $\mathbf{p} = (s_0, q_0, b_0) a_0 (s_1, q_1, b_1) \dots (s_n, q_n, b_n) \in \mathsf{Plays}(\mathcal{G})$ where $q_0 = \delta(\iota, L(s_0))$ and $q_i = \delta(q_{i-1}, L(s_{i-1}))$ for any $1 \le i < n$. The belief state b_i is constructed according to Definition 9 for $0 \le i \le n$. By construction, for a given belief b_i, the observation $O(s_i, a_i, s_{i+1})$, the new belief b_{i+1} is uniquely determined [4].

The following property can be shown.

Lemma 1: Let $\mathbf{p} = (s_0, q_0, b_0) a_0 (s_1, q_1, b_1) \dots (s_n, q_n, b_n)$ be a play over $\mathcal{G}$. For all $i = 0, \dots, n$, it holds $(s_i, q_i) \in b_i$.

Proof: We prove by induction on the length of $\mathbf{p}$. For $i = 0$, $b_0 = (s_0, q_0)$ by construction.

Assume that $(s_k, q_k) \in b_k$ where $1 \le k \le n-1$. Then by Def. 9, for $i = k+1$, $b_{k+1} = \{(s^o, q^o) \mid \exists (\bar{s}, \bar{q}) \in b_k, \bar{a} \in A : O(\bar{s}, \bar{a}, s^o) = O(s_k, a_k, s_{k+1})$ and $T(\bar{s}, \bar{a}) = s^o$ and $\delta(\bar{q}, L(s^o)) = q^o\}$. Since $(s_k, q_k) \in b_k$ and there is $\bar{a} = a_k \in A$ such that $s_{k+1} = T(s_k, \bar{a})$ and $O(s_k, \bar{a}, s_{k+1}) = O(s_k, a_k, s_{k+1})$, and that $\delta(q_k, L(s_k)) = q_{k+1}$, it holds that $(s_{k+1}, q_{k+1}) \in b_{k+1}$. Therefore, $(s_i, q_i) \in b_i$ for $i = 0, \dots, n$. ■

This lemma states that the attacker's belief always contains the true state of the game and the current state of the specification DFA on tracing the label of the play.

Next, we show that two observation-equivalent plays yield, at each instant, equal beliefs about the status of the game.

Lemma 2: Given a play $\rho = s_0 a_0 s_1 a_1 s_2 \dots s_n \in \mathsf{Plays}(G)$, for any of its observation-equivalent play $\rho' \in [\rho]$, assuming

$$\mathfrak{R}(\rho) = (s_0, q_0, b_0) a_0 (s_1, q_1, b_1) \dots (s_n, q_n, b_n)$$

and

$$\mathfrak{R}(\rho') = (s'_0, q'_0, b'_0) a'_0 (s'_1, q'_1, b'_1) \dots (s'_n, q'_n, b'_n),$$

it holds that $b'_i = b_i$ for all $0 \le i \le n$.

Proof: For any $\rho' \in [\rho]$ given $\rho = s_0$, it holds that $\rho' = s \in [s_0]$. By definition, b_0 is a function of the set $[s_0]$ and b'_0 is a function of the set $[s] = [s_0]$. Thus, $b'_0 = b_0$.

Assume that $b'_k = b_k$ where $1 \le k \le n-1$. The belief update gives the new observation $O(s_k, a_k, s'_k)$ based on the current belief b_k. The hypothesis $b_k = b'_k$ and $\rho' \in [\rho]$ implies $O(s'_k, a'_k, s'_{k+1}) = O(s_k, a_k, s_{k+1})$. By the definition, b_{k+1} is a function of b_k, $O(s_k, a_k, s_{k+1})$ and b'_{k+1} is a function of b'_k, $O(s'_k, a'_k, s'_{k+1})$. Since $O(s'_k, a'_k, s'_{k+1}) = O(s_k, a_k, s_{k+1})$, we obtain $b_{k+1} = b'_{k+1}$. Therefore, we have $b'_i = b_i$ for all $1 \le i \le n$ by induction. ■

Lemma 3: For any play $\rho = s_0 s_1 \dots s_n \in \mathsf{Plays}(G)$, let $\mathbf{p} = \mathfrak{R}(\rho) = (s_0, q_0, b_0) a_0 (s_1, q_1, b_1) \dots (s_n, q_n, b_n)$ be the corresponding play in the belief-augmented game $\mathcal{G}$.

- $\rho \in \mathsf{OWPlays}_1$ if and only if $q_n \in F$, $b_n \cap (S \times F) \neq \varnothing$ and $b_n \cap (S \times (Q \setminus F)) \neq \varnothing$.
- $\rho \in \mathsf{RWPlays}_1$ if and only if $q_n \in F$, $b_n \subseteq (S \times F)$.

Proof: $\Rightarrow$: First, for all $i \ge 0$, $(s_i, q_i) \in b_i$. Thus, if $\rho \in \mathsf{OWPlays}_1$, then $q_n \in F$ and thus $b_n \cap S \times F \neq \varnothing$.

Also because $\rho \in \mathsf{OWPlays}$, by Def. 7, there exists $\rho' \in [\rho]$ such that $L(\rho') \not\models \varphi$. Let $\mathfrak{R}(\rho') = (s'_0, q'_0, b'_0) a'_0 \dots (s'_n, q'_n, b'_n)$ be the corresponding play in $\mathcal{G}$ of ρ'. By Lemma 1, it holds that $q'_n \in Q \setminus F$ and thus $b'_n \cap S \times (Q \setminus F) \neq \varnothing$.

Finally, because ρ and ρ' are observation equivalent, then $b_i = b'_i$ for all $0 \le i \le n$ by Lemma 2. Thus, $b'_n \cap S \times (Q \setminus F) \neq \varnothing$ is equivalent to $b_n \cap S \times (Q \setminus F) \neq \varnothing$.

$\Leftarrow$: Conversely, if $q_n \in F$, $b_n \cap S \times F \neq \varnothing$ and $b_n \cap S \times (Q \setminus F) \neq \varnothing$, then, it is noted that, first, $q_n \in F$ means that $\rho \in \mathsf{WPlays}$. Second, let $(s^o_n, q^o_n) \in b_n \cap S \times F$ and $(s'_n, q'_n) \in b_n \cap S \times (Q \setminus F)$. By the construction of the belief, there exists a play $\mathbf{p}' = (s'_0, q'_0, b_0) \dots (s'_n, q'_n, b_n)$ such that $\mathfrak{R}^{-1}(\mathbf{p}') \in [\rho]$. Because $(s'_n, q'_n) \in b_n \cap S \times (Q \setminus F)$, then $L(s'_0 s'_1 \dots s'_n) \not\models \varphi$. And therefore, $\rho \in \mathsf{OWPlays}_1$.

The proof of the second statement follows analogously. ■

Definition 10 (Reachability game and winning region [7]): Given the two-player turn-based game arena $\mathcal{G} = \langle V, A_1 \cup A_2, \Delta, v_0 \rangle$, a reachability objective, denoted $\Diamond X$ for $X \subseteq V$, can be satisfied by a play $\mathbf{p} = v_0 v_1 \dots v_n$ if there exists $0 \le i \le n$, $v_i \in \mathcal{F}_O$. Given a reachability objective $\Diamond X$ for P1, the sure-winning region of P1 is a set $\mathsf{Win}_1(X)$ of states in $\mathcal{G}$ from which P1 has a strategy to enforce a play that satisfies $\Diamond X$, regardless of the counter-strategy for P2.

Theorem 1: Given the original game arena $G = (S, A_1 \cup A_2, T, s_0)$, P1's temporal logic objective φ and its corresponding DFA $\mathcal{A} = (Q, 2^{\mathcal{AP}}, \delta, \iota, F)$, P1 has an opacity-enforcing winning strategy in the original game G with the temporal objective φ if and only if P1 has a sure-winning strategy in the belief-augmented game arena $\mathcal{G}$ with a reachability objective $\Diamond \mathcal{F}_O$ [5] where

$$\mathcal{F}_O = \{(s, q, b) \mid q \in F, b \cap (S \times (Q \setminus F)) \neq \varnothing\}.$$

Proof: If P1 has an opacity-enforcing winning strategy π_1^* in the original game G, then for any strategy π_2 of P2, for any play $\rho \in \mathsf{Plays}(s_0, \pi_1^*, \pi_2)$, $\rho \in \mathsf{OWPlays}_1$. Let $\mathbf{p} = \mathfrak{R}(\rho) = (s_0, q_0, b_0) a_0 (s_1, q_1, b_1) \dots (s_n, q_n, b_n)$. By lemma 3, $q_n \in F$ and $b_n \cap (S \times (Q \setminus F)) \neq \varnothing$. In other words, the strategy π_1^* enforces a run into $\mathcal{F}_O$ regardless of P2's strategy. Thus, this strategy π_1^* is a sure-winning strategy in the game $\mathcal{G}$ for the reachability objective $\Diamond \mathcal{F}_O$.

Conversely, if P1 has a sure-winning strategy π_1^{b*} in the belief-augmented game $\mathcal{G}$. Then for any strategy π_2 of P2, for any play $\mathbf{p} = (s_0, q_0, b_0) a_0 \dots (s_n, q_n, b_n) \in \mathsf{Plays}(\mathcal{G}, \pi_1^{b*}, \pi_2)$, the last state (s_n, q_n, b_n) satisfies $q_n \in F$ and $b_n \cap (S \times (Q \setminus F)) \neq \varnothing$. It is also noted that by Lemma 1, if $q_n \in F$, then $b_n \cap (S \times F)) \neq \varnothing$. According to lemma 3,

[4]This is because the observation function is deterministic. This is not the case when the observation function becomes probabilistic.

[5]A reachability objective is equivalently expressed as $\Diamond p$, reads "eventually p is true" and let atomic proposition $p \in L(q)$ for each $q \in \mathcal{F}_O$.

$\rho = \mathbf{p} \in \mathsf{OWPlays}_1$. Therefore, P1 has an opacity-enforcing winning strategy in the original game G. ■

Since Theorem 1 shows the opacity-enforcing winning strategy in a deterministic game can be computed by solving a zero-sum turn-based deterministic game with a reachability objective for P1 (Note that the safety game of P2 is an infinite game because P2 aims to ensure P1 never eventually reach a goal. See [15] for a tutorial on the solutions of infinite games). The following statements can be derived from the solution concepts of zero-sum reachability games [14], [7]:

- Memoryless, deterministic strategies are sufficient for sure-winning for both P1 and P2.
- The game is determined: For any state $v \in V$, either P1 has a sure-winning strategy or P2 has a sure-winning strategy.
- The sure-winning region for player i is the same as the almost-sure-winning region, which includes all states where player i has a strategy to ensure his/her objective is satisfied with probability one.

Thus, we refer to the sure-winning/almost-sure-winning region as the winning region.

We introduce the winning regions in the belief-augmented game $\mathcal{G}$:

- $\mathsf{Win}_1^{opaque} = \mathsf{Win}_1(\mathcal{F}_O)$ is the opacity-enforcing winning region for P1 (the sure-winning region for P1's objective $\Diamond \mathcal{F}_O$.)
- $\mathsf{Win}_1 = \mathsf{Win}_1(\mathcal{F})$, where $\mathcal{F} = S \times F \times 2^{S\times Q}$, is the winning region for P1 given the objective to satisfy φ without enforcing opacity (the sure-winning region for P1's objective $\Diamond \mathcal{F}$.)

The question is for a play $\rho \in \mathsf{PrefPlays}(G)$, if P1 can enforce satisfying the secret goal φ but cannot enforce opacity with probability one, does P1 have a strategy to satisfy φ but only enforce opacity with some positive probability? The following theorem provides a negative answer to this question.

Theorem 2: Given the belief-augmented game $\mathcal{G}$, let $\widetilde{\Pi}_1$ be the set of winning strategies for P1 with respect to the reachability objective $\Diamond \mathcal{F}$. For any state $v \in \mathsf{Win}_1$, if P1 is restricted to policies in $\widetilde{\Pi}_1$, then one of the two cases occurs:

- If $v \in \mathsf{Win}_1^{opaque}$, then there exists a strategy $\pi_1^o \in \widetilde{\Pi}_1$ that ensures opacity-enforcing winning.
- otherwise, for any strategy $\pi_1 \in \widetilde{\Pi}_1$, any P2's strategy π_2 ensures to reveal the secret when the secret is satisfied.

Proof: The first case is a natural consequence because $\mathcal{F}_O \subseteq \mathcal{F}$. To prove the second case, consider a state $v = (s,q,b) \in \mathsf{Win}_1 \setminus \mathsf{Win}_1^{opaque}$, P2 has a strategy π_2^{safe} to enforce the set Win_1^{opaque} is never reached, due to the duality of the reachability game. Let $\pi_1 \in \widetilde{\Pi}_1$ and $\mathbf{p} = v_0 v_1 v_2 \dots v_n$ be a play starting from $v_0 = v$, induced by the strategy profile $(\pi_1, \pi_2^{\mathsf{safe}})$, the following two conditions shall hold:

- Since π_1 ensures $\mathcal{F}$ must be reached, there exists k, $0 \le k \le n$ such that $v_k = (s_k, q_k, b_k) \in \mathcal{F}$, which means $q_k \in F$ and $b_k \cap S \times F \neq \varnothing$ due to $(s_k, q_k) \in b_k$ (Lemma 1).
- Since π_2^{safe} ensures a state in $\mathcal{F}_O$ is not reached, for any $j \ge 0$, $v_j = (s_j, q_j, b_j) \notin \mathcal{F}_O$, which means either $q_j \notin F$ or $b_j \cap S \times (Q \setminus F) = \varnothing$.

Given that $q_k \in F$ followed from condition 1, then it is the case that $b_k \cap S \times F \neq \varnothing$ and, from condition 2, $b_k \cap S \times (Q \setminus F) = \varnothing$. As a result, it must be the case that $b_k \subseteq S \times F$. Thus, the safety strategy played by P2 against P1 who only selects a winning strategy in $\widetilde{\Pi}_1$ ensures to reveal P1's secret. ■

Based on Theorem 2, from $v \in \mathsf{Win}_1 \setminus \mathsf{Win}_1^{opaque}$ a P2's strategy to reveal P1's secret can be any strategy in the game in which P2 aims to prevent a play reaching Win_1^{opaque}. There can be two outcomes, either P1 chooses to not satisfy the goal by not committing to a strategy in $\widetilde{\Pi}_1$, or the secret will be revealed the moment a state in $\mathcal{F}$ is reached.

Lemma 4: The time complexity of our algorithm to solve our problem with the game arena $G = (S, A_1 \cup A_2, T, s_0)$ and the DFA $\mathcal{A} = (Q, 2^{\mathcal{AP}}, \delta, \iota, F)$ as inputs to the problem is $\mathcal{O}(|A||S|^2|Q|^2 2^{|S||Q|})$.

Proof: The time of our algorithm consists of (1) the time to construct the belief-based game $\mathcal{G}$ and (2) the time to compute a winning strategy for P1 on $\mathcal{G}$. The belief-based game $\mathcal{G}$ has $\mathcal{O}(|V|) = \mathcal{O}(|S||Q|2^{|S||Q|})$ states and $\mathcal{O}(|\Delta|) = \mathcal{O}(|A||V|)$ transitions. Using appropriate data structures, mainly hash-tables, it takes $\mathcal{O}(|A||S|^2|Q|^2 2^{|S||Q|})$ to construct the belief-based game. This is because for each state of this game, which is in the form of a $v = (s,q,b) \in V$, and for each action $a \in A$, we must compute what is $\Delta(v,a)$. Computing that requires access $T(\bar{s},a)$ and $\delta(\bar{q},a)$ for each $(\bar{s},\bar{o}) \in b$ and the size of b can be $|S||Q|$ in the worst case. Using the Zielonka's algorithm [23], it takes $\mathcal{O}(|V| + |\Delta|) = \mathcal{O}(|A||S||Q|2^{|S||Q|})$ to compute a winning strategy for P1 on $\mathcal{G}$. Therefore, the time complexity of our algorihtm is $\mathcal{O}(|A||S|^2|Q|^2 2^{|S||Q|})$. ■

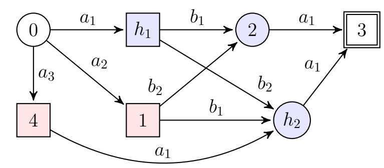

Fig. 1: An illustrative example of opacity-enforcing winning.

IV. EXPERIMENTS [6]

Example 1: In this illustrative example (Fig 1), we demonstrate how to solve an opacity-enforcing winning strategy for P1. The game consists of seven states, with circles representing P1's states and squares representing P2's states. P1 can choose from three actions, $A_1 = \{a_1, a_2, a_3\}$, while P2 has two actions, $A_2 = \{b_1, b_2\}$. State 3 is a blocking state so that when 3 is reached, the game ends.

The partial observations of the attacker are described as follows: The states $h_1, h_2, 2$ are indistinguishable, and the states $1, 4$ are indistinguishable. The actions chosen by both players cannot be observed by the attacker. The goal of P1

[6]The code is available on https://github.com/AronYoung414/Synthesis-of-Opacity-Enforcing-Winning-Strategies-Against-Colluded-Opponent.

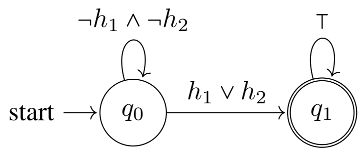

Fig. 2: The DFA for $\Diamond(h_1 \vee h_2)$.

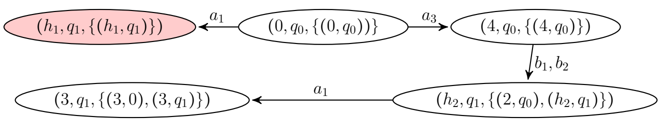

Fig. 3: The belief-augmented game for illustrative example.

is to visit h_1 or h_2, i.e. , to satisfy $\varphi = \Diamond(h_1 \vee h_2)$. The DFA representing the goal is shown in Fig. 2.

A fragment of the belief-augmented game is drawn in Fig. 3. A state, for example, $(h_2, q_1, \{(2, q_0), (h_2, q_1)\})$ means the game state is h_2, the DFA state is q_1 and the attacker's belief is $\{(2, q_0), (h_2, q_1)\}$. The action a_2 is omitted. It leads to a losing state because P2 can select action b_2 at 1 and thus P1 cannot satisfy the objective.

Based on the solution of the reachability game, the following result is obtained: At state 0, P1 has an opacity-enforcing winning strategy. The strategy results in two possible plays of the form $0 \xrightarrow{a_3} 4 \xrightarrow{b} h_2 \xrightarrow{a_1} 3$, where $b \in \{b_1, b_2\}$. The observations made during these plays are $0\{1,4\}\{h_1, h_2, 2\}3$. By applying the inverse of the observation function to this observation, we can identify a play $\rho' = 0 \xrightarrow{a_2} 1 \xrightarrow{b_2} 2 \xrightarrow{a_1} 3$ such that $L(\rho') \notin \mathcal{L}(\mathcal{A})$ and $O(\rho') = 0\{1,4\}\{h_1, h_2, 2\}3 = O(\rho)$. Thus, this play ρ is opaque and winning for P1.

It is noted that in this game, if opacity is not enforced, P1 can satisfy his objective by taking action a_1 or a_3 from state 0. However, action a_1 can lead to a state outside of Win_1^{opaque} in the belief-augmented game because by taking a_1, state h_1 will be reached. And two runs are possible: $0 \xrightarrow{a_1} h_1 \xrightarrow{b_1} 2 \xrightarrow{a_1} 3$ or $0 \xrightarrow{a_1} h_1 \xrightarrow{b_2} h_2 \xrightarrow{a_1} 3$. The observations are $0, \{h_1, h_2, 2\}, \{h_1, h_2, 2\}, 3$ for both runs. Regardless of P2 taking action b_1 or b_2, the attacker can infer at least h_1 is visited and the formula φ is satisfied.

Example 2: Consider the 5×5 grid-world in Fig. 4. In this example, the aerial robot, P1, and the ground robot, P2, have 4 movement actions $\{N, S, W, E\}$ to navigate the environment. P1 can navigate the whole area, while P2 can only navigate the area in gray and it cannot enter the water area, the 9 cells in the middle. P1's temporal goal is described by the temporal logic formula $\varphi = \Diamond(G_0 \wedge \Diamond(G_1 \vee G_2))$, stating first visit G_0 and then G_1 or G_2.

If the robots move into the same cell, P2 can intercept P1, and the game ends. The game is turn-based, and at each step, only one robot makes a move. Besides the two robots, there is an observer to whom the current state of the game is only partially observable. The observer obtains information from two kinds of sensors in the environment,

1. The 5 range sensors where each guards a row of the grid world (when P1 enters a row, the observer knows at which row P1 is located, but it does not know in which column P1 is); and
2. The cross-shaped range sensor is shown in Figure 4b, which is carried by robot P2. Once robot P1 enters the range of the sensor, the observer will receive the exact location of P1.

The goal of P2 is to intercept P1 if it can or reveal P1's secret to the observer if it cannot prevent P1 from winning the game.

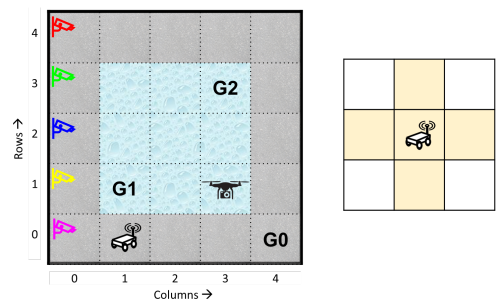

Fig. 4: **(left)** A robot motion planning problem motivating our opacity-enforcing temporal planning. The aerial robot's temporal goal is to first collect a sample at G_0 and then collect a sample at either G_1 or G_2. Five range sensors guard the environment, where each tells the observer the P1 is in that row whenever P1 enters that row. The range sensors do not know the column at which P1 might be located. **(right)** The cross-shaped range sensor is carried by the ground robot.

For this example, we considered 25 different instances for the 25 possible initial locations of the aerial robot, P1. In each instance, we fix the initial location of the ground robot, P2, at the bottom-left cell $(0,0)$ of the environment. We used our algorithm to solve these 25 instances of our problem. Fig. 5 shows the results of our algorithm. The cells that contain red circles show the P1's initial locations from which P2 has a winning strategy to prevent P1 from satisfying φ. Cells with a yellow circle show P1's initial locations from which, if the game starts, P1 has a winning strategy to satisfy φ but cannot enforce opacity. The green circles are shown in the cells from which, if the game starts, P1 has an opacity-enforcing winning strategy. Accordingly, the green and yellow cells are winning initial states for P1. But if P1 wants to enforce opacity, then only green initial states allow it to do so.

We have run 25 experiments to solve the winning regions from 25 different initial states on Intel (R) Core (TM) i7-5820K CPU @ 3.30GHz 3.30 GHz. The average time consumed for solving one game is 7.2 seconds. Fig. 6 shows the opacity-enforcing winning run when P1 starts from $(0,1)$ and P2's initial location is $(0,0)$. We encode the state as $((p_{1r}, p_{1c}, p_{2r}, p_{2c}, turn), q)$ when

- P1's position is (p_{1r}, p_{1c}), P2's position is (p_{2r}, p_{2c}),
- $turn = 1, 2$ indicates the turn of the game (either P1's turn or P2's turn),
- q is the state of DFA. The accepting states $F = \{0\}$.

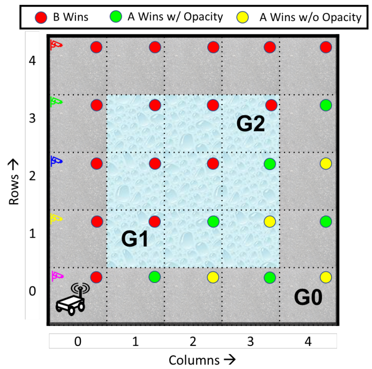

Fig. 5: The results of our experiment for the 25 instances of the grid-world example, where P2's initial state is fixed at $(0,0)$ in all those instances, and P1's initial state varies across the instances.

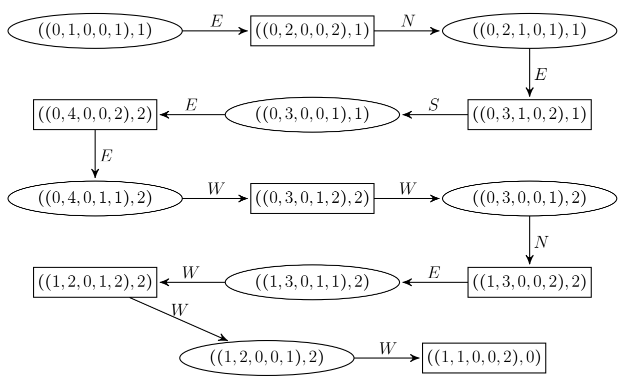

Fig. 6: Opacity-enforcing winning path given P1 starts from $(0,1)$. Circles: P1's states, Rectangles: P2's states

In the last state of the path, the belief state is $\{(1,1,0,0,2),0),((1,3,0,0,2),2)\}$ that contains a state with $q=2$. Since $q=2$ is not in F, it is an opacity-enforcing winning play for P1.

V. Conclusion and Future Work

In this work, we formulate and solve the control synthesis problem for an agent to satisfy its temporal goal while enforcing opacity against its uncontrollable environment player, who colludes with a passive observer. We prove that in a turn-based deterministic game with perfect observation between P1 and P2, there does not exist a winning and positive opacity enforcing strategy for P1. However, it remains to be answered whether a winning and positively opacity-enforcing strategy may exist for concurrent/stochastic games or games with partial observations, or whether existing game objectives can capture infinite-state opacity [22]. For other classes of games and different information structures between players and observers, quantitative notions of opacity such as probabilistic opacity [3], [18] can be further investigated, leveraging results from stochastic games.

References

[1] J. Bryans, M. Koutny, L. Mazaré, and P. Ryan, "Opacity generalised to transition systems." *International Journal of Information Security*, vol. 7, no. 6, 2008.

[2] J. W. Bryans, M. Koutny, and P. Y. Ryan, "Modelling opacity using petri nets," *Electronic Notes in Theoretical Computer Science*, vol. 121, pp. 101–115, 2005, proceedings of the 2nd International Workshop on Security Issues with Petri Nets and other Computational Models (WISP 2004).

[3] B. Bérard, J. Mullins, and M. Sassolas, "Quantifying opacity," *Mathematical Structures in Computer Science*, vol. 25, no. 2, pp. 361–403, Feb. 2015, publisher: Cambridge University Press.

[4] F. Cassez, J. Dubreil, and H. Marchand, "Synthesis of opaque systems with static and dynamic masks," *Formal Methods in System Design*, vol. 40, no. 1, pp. 88–115, Feb. 2012. [Online]. Available: https://doi.org/10.1007/s10703-012-0141-9

[5] K. Chatterjee and L. Doyen, "Partial-Observation Stochastic Games: How to Win When Belief Fails," in *2012 27th Annual IEEE Symposium on Logic in Computer Science*, Jun. 2012, pp. 175–184, iSSN: 1043-6871.

[6] J. Dubreil, P. Darondeau, and H. Marchand, "Opacity enforcing control synthesis," in *2008 9th International Workshop on Discrete Event Systems*, 2008, pp. 28–35.

[7] E. Gradel and W. Thomas, *Automata, Logics, and Infinite Games: A Guide to Current Research*. Springer Science & Business Media, Dec. 2002.

[8] C. N. Hadjicostis, *Estimation and Inference in Discrete Event Systems: A Model-Based Approach with Finite Automata*. Springer, 2020.

[9] L. Hélouët, H. Marchand, and L. Ricker, "Opacity with powerful attackers," *IFAC-PapersOnLine*, vol. 51, no. 7, pp. 464–471, 2018.

[10] R. Jacob, J.-J. Lesage, and J.-M. Faure, "Overview of discrete event systems opacity: Models, validation, and quantification," *Annual reviews in control*, vol. 41, pp. 135–146, 2016.

[11] S. Lafortune, F. Lin, and C. N. Hadjicostis, "On the history of diagnosability and opacity in discrete event systems," *Annual Reviews in Control*, vol. 45, pp. 257–266, 2018-01-01.

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

[13] Z. Manna and A. Pnueli, *The Temporal Logic of Reactive and Concurrent Systems*. New York, NY: Springer New York, 1992. [Online]. Available: http://link.springer.com/10.1007/978-1-4612-0931-7

[14] D. A. Martin, "Borel Determinacy," *Annals of Mathematics*, vol. 102, no. 2, pp. 363–371, 1975.

[15] R. Mazala, "Infinite games," in *Automata logics, and infinite games: A guide to current research*, E. Grädel, W. Thomas, and T. Wilke, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2002, pp. 23–38.

[16] A. Saboori and C. N. Hadjicostis, "Notions of security and opacity in discrete event systems," in *2007 46th IEEE Conference on Decision and Control*. IEEE, 2007, pp. 5056–5061.

[17] ——, "Verification of infinite-step opacity and analysis of its complexity*," *IFAC Proceedings Volumes*, vol. 42, no. 5, pp. 46–51, 2009, 2nd IFAC Workshop on Dependable Control of Discrete Systems.

[18] ——, "Current-State Opacity Formulations in Probabilistic Finite Automata," *IEEE Transactions on Automatic Control*, vol. 59, no. 1, pp. 120–133, Jan. 2014, conference Name: IEEE Transactions on Automatic Control.

[19] A. Wintenberg, M. Blischke, S. Lafortune, and N. Ozay, "A general language-based framework for specifying and verifying notions of opacity," *Discrete Event Dynamic Systems*, vol. 32, no. 2, pp. 253–289, 2022.

[20] Y.-C. Wu and S. Lafortune, "Comparative analysis of related notions of opacity in centralized and coordinated architectures," *Discrete Event Dynamic Systems*, vol. 23, no. 3, pp. 307–339, 2013.

[21] Y. Xie, X. Yin, and S. Li, "Opacity enforcing supervisory control using nondeterministic supervisors," *IEEE Transactions on Automatic Control*, vol. 67, no. 12, pp. 6567–6582, 2021.

[22] 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, 2019-01-01.

[23] W. Zielonka, "Infinite games on finitely coloured graphs with applications to automata on infinite trees," *Theoretical Computer Science*, vol. 200, no. 1-2, pp. 135–183, 1998.