Synthesis of Deceptive Strategies in Reachability Games with Action Misperception

Abhishek N. Kulkarni and **Jie Fu**
Worcester Polytechnic Institute, Worcester, USA
{ankulkarni, jfu2}@wpi.edu

Abstract

We consider a class of two-player turn-based zero-sum games on graphs with reachability objectives, known as reachability games, where the objective of Player 1 (P1) is to reach a set of goal states, and that of Player 2 (P2) is to prevent this. In particular, we consider the case where the players have asymmetric information about each other's action capabilities: P2 starts with an incomplete information (misperception) about P1's action set, and updates the misperception when P1 uses an action previously unknown to P2. When P1 is made aware of P2's misperception, the key question is *whether P1 can control P2's perception so as to deceive P2 into selecting actions to P1's advantage?* To answer this question, we introduce a dynamic hypergame model to capture the reachability game with evolving misperception of P2. Then, we present a fixed-point algorithm to compute the deceptive winning region and strategy for P1 under almost-sure winning condition. Finally, we show that the synthesized deceptive winning strategy is at least as powerful as the (non-deceptive) winning strategy in the game in which P1 does not account for P2's misperception. We illustrate our algorithm using a robot motion planning in an adversarial environment.

1 Introduction

Synthesis of winning strategies in reachability games is a central problem in reactive synthesis [Pnueli and Rosner, 1989], control of discrete event systems [Ramadge and Wonham, 1989], and robotics [Fainekos *et al.*, 2009]. In a two-player reachability game, a controllable player, P1 (pronoun "she"), plays against an uncontrollable adversarial player, P2 (pronoun "he"), to reach the goal states. These games have been extensively studied in algorithmic game theory [de Alfaro *et al.*, 2007] and reactive synthesis [Bloem *et al.*, 2012]. Polynomial-time algorithms are known for synthesizing *sure-winning* and *almost-sure winning* strategies, when both players have complete and symmetric information. However, the solution concepts for such games under asymmetric information have not been thoroughly studied.

Information asymmetry arises when a player has some private information that are not shared with others [Rasmusen, 1989]. We consider the case when P1 has complete information about both players' action capabilities, but P2 starts with an incomplete information about P1's action capabilities. As two players interact, their information evolves. Particularly, when P1 uses an action previously unknown to P2, P2 can update his knowledge about the other's capabilities using an *inference mechanism*. In response, P2 would update his counter-strategy. We are interested in the following question: *if P1 is made aware of the initial information known to P2 and his inference mechanism, can P1 find a strategy to control P2's information in such a way that P2's counter-strategy given his evolving information is advantageous to P1?* In the context of qualitative analysis of reachability games, this could be interpreted as: *from a state that is losing for P1 in a game with symmetric information, can P1 reach his goal from the same state when the information is asymmetric?* We note that a strategy of P1 that controls P2's information to P1's advantage is indeed deceptive [Ettinger and Jehiel, 2010]. In this paper, we establish that such a deceptive winning strategy may exist under almost-sure winning condition and propose an algorithm to synthesize it.

We approach the question by modeling the interaction between P1 and P2 as a hypergame [Bennett, 1977]. A hypergame allows players to play different games in their minds and further allows them to model the games that others might be playing. In literature, hypergames and Bayesian games [Harsanyi, 1967] are common models to capture game-theoretic interactions with asymmetric, incomplete information. In Bayesian games, each player uses his incomplete information to define a probability distribution over the possible types of the opponent. The distributions over types are assumed to be common knowledge. In hypergames, no such probabilistic characterizations of incomplete information is used or assumed. For action deception, P2 has incomplete information about P1's capabilities but does not have a prior knowledge about the set of possible types of P1. Thus, we adopt the hypergame model to understand action deception. In the past, hypergame model has been used to study deception [Gutierrez *et al.*, 2015; Kovach, 2016]. These papers mainly focus on extending the notion of Nash equilibrium to level-k normal form hypergames. [Gharesifard and Cortés, 2014] use the notion of

H-digraph to establish necessary and sufficient conditions for deceivability. An H-digraph models a hypergame as a graph with nodes representing different outcomes in a normal-form game. However, our game model is not a normal-form game, but instead a game on graph. A hypergame model based on a game on graph has been defined in [Kulkarni and Fu, 2019] where one player has incomplete information about the opponent's task specification. However, their model assumes the perception of P2 to remain constant, whereas in our case, the information available to both players' may evolve as the game progresses. Given this background, we state the two key contributions of the paper.

Dynamic Hypergame Model for Action Misperception. We introduce a dynamic hypergame on graph model that captures (i) the evolving information of P2, and (ii) the P1's information regarding current perception of P2, in a single graphical model.

Deceptive Almost-sure Winning Synthesis Algorithm. Given a dynamic hypergame model, we propose an algorithm to compute deceptive almost-sure winning region, and thereby a deceptive almost-sure winning strategy for P1 that effectively exploits P2's misperception. We prove that the deceptive winning region is a superset of the (non-deceptive) winning region in the symmetric information game where P1 does not account for P2's misperception. This in turn implies that the deceptive winning strategy is at least as powerful as the (non-deceptive) winning strategy.

2 Preliminaries

Let Σ be a finite alphabet. A sequence of symbols $w = w_0 w_1 \dots w_n$ with $w_i \in \Sigma, i = 0, 1, \dots, n$ is called a *finite word* and Σ^* is the set of finite words that can be generated with alphabet Σ. We denote by Σ^ω, the set of ω-regular words obtained by concatenating the elements in Σ infinitely many times. Given a set X, let $\mathsf{Dist}(X)$ be the set of probability distributions over X. Given a distribution $\delta \in \mathsf{Dist}(X)$, the set $\mathsf{Supp}(\delta) = \{x \in X \mid \delta(x) > 0\}$ is called the support of the distribution.

2.1 Games on Graph

Consider an interaction between two players; P1 with a reachability objective and P2 with an objective of preventing P1 from completing her task.

Definition 1 (Game on Graph)**.** *Let the action sets of P1 and P2 be A_1 and A_2, respectively. Then, a turn-based game on graph is the tuple*

$$\mathcal{G} = \langle S, Act, T, F \rangle,$$

where

- *$S = S_1 \cup S_2$ is the set of states partitioned into P1's states, S_1, and P2's states, S_2. P1 chooses an action when $s \in S_1$ and P2 chooses an action when $s \in S_2$.*
- *$Act = A_1 \cup A_2$ is set of actions for P1 and P2.*
- *$T : S \times Act \to S$ is a deterministic transition function that maps a state and an action to a successor state.*
- *$F \subseteq S$ is a set of final states.*

Algorithm 1 Almost-Sure Winning Region [Mazala, 2002]

```
Inputs: $\mathcal{G}, F$
$Z_0 = F$
repeat
    $\mathsf{Pre}_1(Z_k) = \{s \in S_1 \mid \exists a \in A_1 \text{ s.t. } T(s,a) \in Z_k\}$
    $\mathsf{Pre}_2(Z_k) = \{s \in S_2 \mid \forall b \in A_2 : T(s,b) \in Z_k\}$
    $Z_{k+1} = Z_k \cup \mathsf{Pre}_1(Z_k) \cup \mathsf{Pre}_2(Z_k)$
until $Z_{k+1} = Z_k$
return $\mathsf{Win}_1 = Z_k$
```

A *trace* in the game $\mathcal{G}$ is an infinite, ordered sequence of state-action pairs $\tau = (s_0, a_0), (s_1, a_1), (s_2, a_2), \dots$. We write $\tau[n] = (s_n, a_n)$ to denote n-th state-action pair. A *run* ρ is the projection of trace onto the state space. We denote it as the sequence $\rho = \tau \downharpoonright_S = s_0 s_1 s_2 \dots$. Similarly, the *action-history* is the projection of trace onto the action space, denoted by $\alpha = \tau \downharpoonright_{Act} = a_0 a_1 a_2 \dots$. The k-th element in a run (resp. action-history) is denoted by ρ_k (resp. α_k).

In this paper, we consider *reachability objectives* for P1. The set of states that occur in a run is given by $\mathsf{Occ}(\rho) = \{s \in S \mid \exists k \in \mathbb{N} \cdot s = \rho_k\}$. A run ρ is said to be winning for P1 in the reachability objective if it satisfies $\mathsf{Occ}(\rho) \cap F \neq \emptyset$. If a run is not winning for P1, then it is winning for P2.

The memoryless stochastic or randomized strategies for P1 and P2 are defined as $\pi : S_1 \to \mathsf{Dist}(A_1)$ and $\sigma : S_2 \to \mathsf{Dist}(A_2)$, respectively. As the model in Def. 1 is a deterministic turn-based game model, it is sufficient to consider memoryless strategies [Mazala, 2002]. Let $\Omega_s^{\pi,\sigma}$ be the exhaustive set of runs that result when P1 and P2 play strategies π and σ in a game starting at the state $s \in S$. The randomized strategies of P1 and P2 induce a Markov chain from $\mathcal{G}$–that is, a probability distribution over the set $\Omega_s^{\pi,\sigma}$.

Given a state $s \in S$, a randomized strategy π is *almost-sure winning* (ASW) for P1, if and only if for every possible randomized strategy σ of P2, the probability is one for a run that satisfies $\mathsf{Occ}(\rho) \cap F \neq \emptyset$, given the distribution of runs induced by (π, σ). A state is called an *almost-sure winning (ASW) state* for P1, if there exists an ASW strategy for P1 from that state. The exhaustive set of almost-sure winning states for P1 is called her *ASW region*. The almost-sure winning region can be computed using Alg. 1.

Let us introduce a running example that we shall use to explain the concepts in this paper.

Example 1. *Consider the game graph in Fig. 1. The circle states, $\{s_1, s_3\}$, are P1 states and the square states, $\{s_0, s_2\}$, are P2 states. The objective of P1 is to reach to the final state s_0 from the initial state s_2. P1's action set is $A_1 = \{a_1, a_2\}$, and P2's action set is $A_2 = \{b_1, b_2\}$.*

The ASW region for P1 in the game is $\mathsf{Win}_1 = \{s_0, s_1\}$*. This can intuitively be understood as follows. P1 can win from state s_1 by choosing the action a_1. However, the states s_2 and s_3 are losing for P1 because P2 has a strategy to indefinitely restrict the game within the states s_2, s_3 by choosing the action b_2 at the state s_2.*

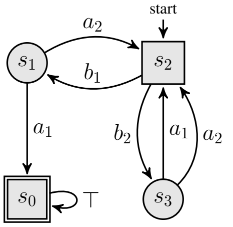

Figure 1: An example game on graph

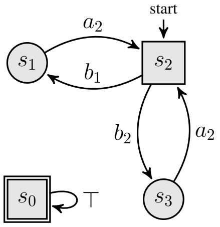

Figure 2: Perceptual game of P2

2.2 Action Misperception and Information Asymmetry

In this paper, we consider the sub-class of games with asymmetric information in which P2 has incomplete information about P1's action capabilities. This is formalized in the following assumption.

Assumption 1. *P1 has complete information about the players' action sets, i.e. P1 knows A_1 and A_2. P2 only knows his own action set A_2, but (mis)perceives P1's action set to be a subset $X \subsetneq A_1$. Both players have complete information about the game state space S and the final states F.*

The result of Assumption 1 is that P1 and P2, in their minds, play different games to synthesize their respective strategies. We refer to these games as the *perceptual games* of the players. P1's perceptual game is identical to the ground-truth game; $\mathcal{G}(A_1) = \langle S, A_1 \cup A_2, T, F\rangle$, while P2's perceptual game is a game under misperception; $\mathcal{G}(X) = \langle S, X \cup A_2, T, F\rangle$. Let us formalize the new notation used to distinguish between the perceptual games of P1 and P2.

Notation 1. Let $X \subseteq A_1$ be a subset of P1's action set. We denote a perceptual game in which P1's action set is X by $\mathcal{G}(X) = \langle S, X \cup A_2, T, F\rangle$. The winning regions for P1 and P2 in the game $\mathcal{G}(X)$ are denoted by $\mathsf{Win}_1(X)$ and $\mathsf{Win}_2(X)$, respectively.

Assuming P1 and P2 to be rational players, they would use the solution approach reviewed in Section 2.1 to compute their winning strategies in their respective perceptual games. That is, P1 will solve $\mathcal{G}(A_1)$ in her mind to obtain π and P2 will solve $\mathcal{G}(X)$ in his mind to compute σ. However, P1 is likely to compute a conservative strategy because she overestimates the information available to P2. Naturally, we want to know *whether P1 can improve her strategy if she is made aware of P2's current misperception X?*

Before we answer the above question, recall from Section 1 that we allow P2's misperception to evolve during the game. For instance, what would happen when P2 observes P1 playing an action $a \in A_1$, which P2 did not believe to be in P1's action set? We might argue that P2 will at least add a new action a to his perceived action set, X, of P1. Thus, the new perception would be $X \cup \{a\}$. Also, P2 might be capable of complex inference. That is, on observing that P1 can perform an action a, P2 might infer that P1 must be capable of actions b and c, thus, updating his perception set to $X \cup \{a, b, c\}$. To capture such inference capabilities, we introduce a generic perception update function for P2 as follows,

Definition 2 (Inference Mechanism)**.** *A deterministic inference mechanism is a function $\eta : 2^{A_1} \times A_1 \to 2^{A_1}$ that maps a subset of actions $X \subseteq A_1$ and an action $a \in A_1$ to an updated subset of actions $Y = \eta(X, a)$ such that $a \in Y$.*

Given the formalism of inference mechanism to capture the evolving misperception of P2 during the game, we now state our problem statement.

2.3 Problem Statement

When P2's misperception evolves during the game, P1 should also strategize to reveal an action that is not currently known to P2. By doing so, P1 may control the evolution of P2's misperception to her advantage. Let us revisit Example 1 to develop an intuition of how P1 might control P2's perception.

Example 2 (Example 1 contd.)**.** *Suppose that, in Example 1, P2 starts with a misperception about P1's action capabilities as $X_0 = \{a_2\}$. In this setup, let us understand the perceptual games of the players. P1's perceptual game; $\mathcal{G}_1 = \mathcal{G}(A_1)$, is the same as the ground-truth game as shown in Fig. 1. P2's perceptual game, initially, is the game $\mathcal{G}_2 = \mathcal{G}(X_0)$ that does not include edges labeled with action a_1 as shown in Fig. 2. Clearly, as the final state s_0 is not reachable in $\mathcal{G}_2$, P2 misperceives both actions b_1 and b_2 to be safe to play at state s_2, when only the action b_2 is safe in the ground-truth game.*

When P1 is aware of P2's misperception, X_0, a deceptive strategy should, intuitively, not use a_1 unless the game state is s_1. Assuming P2 uses a randomized strategy with support A_2, it is easy to compute that the probability of reaching the state s_1 from the initial state s_2 is one. At s_1, P1 can win the game by choosing a_1 in one step. We note that if P1 uses a_1 in state s_3, then P2 will update his perception to $X_1 = A_1$, and mark the action b_1 to be unsafe in state s_2. Thus, P1 will never be able to win the game.

We call such a strategy of P1, where she intentionally controls P2's misperception, as an *action-deceptive strategy* or simply a deceptive strategy (see Def. 6 for a formal definition). We formalize our problem statement.

Problem 1. *Consider a reachability game under information asymmetry in which Assumption 1 holds. If P1 is informed of the initial misperception of P2, X_0, and his inference mechanism η, then determine a deceptive almost-sure winning strategy for P1 to satisfy her reachability objective.*

In particular, we want to investigate whether the use of deception is advantageous for P1 or not. We say P1 gets advantage with deception if at least one game state that is not almost-sure winning for P1 in the game without deception becomes winning for her with use of deception.

3 Dynamic Hypergame for Action Deception

When two players play different games in their minds, their interaction is better modeled as a hypergame [Bennett, 1977].

Definition 3 (First-level Hypergame)**.** *A first-level hypergame is defined as a tuple of the perceptual games being played by the players,*

$$\mathcal{H}^1 = \langle \mathcal{G}_1, \mathcal{G}_2 \rangle,$$

where the P1 (resp. P2) solves the game $\mathcal{G}_1$ (resp. $\mathcal{G}_2$) to compute the winning strategy.

When one of the players is aware of the other player's perception, but the other player is not, we say that a second-level hypergame is being played. In line with Problem 1, we assume that P1 is aware of the P2's misperception, *i.e.* P1 knows the action set $X \subseteq A_1$ as perceived by P2. If P1 knows X, then P1 can construct the perceptual game of P2, $\mathcal{G}(X)$, and therefore P1 knows the first-level hypergame $\mathcal{H}^1$.

However, P2's perception evolves when he observes P1 using actions that are not included in X. This means that the game $\mathcal{G}(X)$ changes when P2's perception changes, and so does the hypergame $\mathcal{H}^1$. We now define a graph to model the hypergame representing the evolving misperception of P2, called as a dynamic hypergame on graph.

Definition 4 (Dynamic Hypergame on Graph)**.** *Let* $\Gamma = 2^{A_1}$. *We define the dynamic hypergame on graph with action misperception as*

$$\mathcal{H} = \langle V, Act, \Delta, \mathcal{F} \rangle,$$

where

- $V = S \times \Gamma$ *is the set of hypergame states,*
- $Act = A_1 \cup A_2$ *is the set of actions of P1 and P2,*
- $\Delta : V \times Act \to V$ *is the transition function such that* $(s, X) \xrightarrow{a} (s', X')$ *if and only if* $s' = T(s, a)$ *and* $X' = \eta(X, a)$,
- $\mathcal{F} = F \times \Gamma$ *is the set of final states.*

For convenience, we shall refer to the dynamic hypergame on graph as simply hypergame in the remainder of the paper. Analogous to game on graph, a trace in a hypergame is an infinite, ordered sequence of state-action pairs given by $\tau = (v_0, a_0)(v_1, a_1)\ldots$ and the action-history is defined as $\alpha = \tau \downharpoonright_{Act} = a_0 a_1 a_2 \ldots$. In contrast with the game on graphs, we distinguish between a *hypergame-run* (h-run) as a projection of trace onto the hypergame state space $\nu = \tau \downharpoonright_V = v_0 v_1 v_2 \ldots$ and a *game-run* as a projection of trace onto game state space $\rho = \tau \downharpoonright_S = s_0 s_1 s_2 \ldots$, where s_k is the game state corresponding to hypergame state $v_k = (s_k, \cdot)$. A reachability objective is said to be satisfied over the hypergame if and only if $\mathsf{Occ}(\nu) \cap \mathcal{F} \neq \emptyset$, *i.e.* the hypergame-run ν visits a winning state in $\mathcal{F}$. By definition, the following statement is always true; $\mathsf{Occ}(\rho) \cap F \neq \emptyset$ if and only if $\mathsf{Occ}(\nu) \cap \mathcal{F} \neq \emptyset$.

Example 3 (Example 1 contd.)**.** *The hypergame modeling the asymmetric information from Example 2 is shown in Fig. 3 (the figure only shows the reachable states). Every state is represented as a tuple of a game state and the current misperception of P2 at that state. Letting* $X_1 = \{a_2\}$ *and* $X_2 = \{a_1, a_2\}$, *the traces* $\tau_1 = ((s_2, X_1), b_1), ((s_1, X_1), a_1), (s_0, X_2)$, *and* $\tau_2 = ((s_2, X_1), b_2), ((s_3, X_1), a_1), ((s_2, X_2), b_1), ((s_1, X_2), a_1), (s_0, X_2)$ *are the examples of winning traces for P1 in the hypergame. However, in the next section, we will see that (a)* τ_2 *will never be observed when both players act rationally, whereas* τ_1 *might be observed, and (b) how to identify all rationally possible winning traces for P1.*

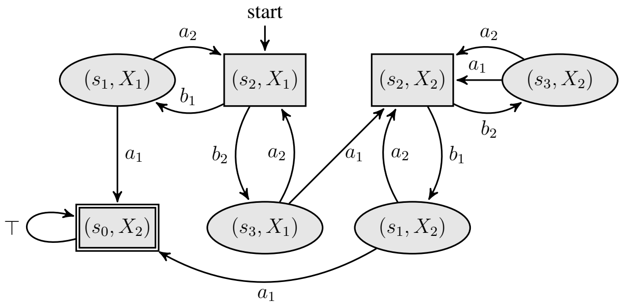

Figure 3: The dynamic hypergame on graph

4 Synthesis of Deceptive Almost-Sure Winning Strategy

In this section, we present an algorithm to synthesize deceptive almost-sure winning (DASW) strategies that are rationally possible in the hypergame. To do so, we must reason about P2's perception and his subjectively rational actions. To understand this, we revisit the concept of permissive strategies in a game on graph.

Recall that an action is permissive for a player at a given state if the player can stay within the winning region by performing that action [Bernet *et al.*, 2002]. In a game under information asymmetry, whether a state is winning or not depends on the player's perception. Hence, we define the notion of perceptually permissive actions, which extends the definition of permissive actions to model evolving perception.

Definition 5 (Perceptually Permissive Actions of P2)**.** *Let* $u = (s, X) \in V_2$ *and* $v = (s', X)$ *be two hypergame states such that* $v = \Delta(u, b)$ *for some* $b \in A_2$. *Then, the set* $M(u) = \{a \in A_2 \mid s' \in \mathsf{Win}_2(X)\}$ *is the set of P2's perceptually permissive actions at* u.

In words, the perceptually permissive actions for a given state $u = (s, X)$ is the set of permissive actions for P2 in the perceptual game with action set X.

Remark 1. By Def. 2, P2's action does not have any effect on P2's perception. Hence, the P2's perception at v is the same as her perception at u, *i.e.* X.

Assumption 2. *At a state* $v \in V_2$, *P2 plays a randomized strategy,* σ, *defined over the perceptually permissive actions* $M(v)$ *such that* $\mathsf{Supp}\,(\sigma(v)) = M(v)$.

Now, we formalize the notion of Deceptive Almost-Sure Winning (DASW) strategy.

Definition 6 (Deceptive Almost-Sure Winning (DASW) Strategy)**.** *Given a hypergame state* $v \in V$, *a strategy* π *is said to be deceptive almost-sure winning (DASW) strategy for*

P1 if and only if for every P2's strategy σ satisfying Assumption 2, the probability of an h-run ν induced from $\mathcal{H}$ by (π, σ) satisfying $\mathsf{Occ}(\nu) \cap \mathcal{F} \neq \emptyset$ is one.

The states at which P1 has a DASW strategy are called as DASW states. The exhaustive set of all DASW states is called DASW region.

Now, we discuss Alg. 2 that computes the DASW region for P1. Our algorithm is inspired by the algorithm presented in [de Alfaro *et al.*, 2007] to compute the almost-sure winning (ASW) region in the concurrent ω-regular games. The idea behind Alg. 2 is to identify the states where P2 perceives some unsafe actions as safe due to misperception. This is achieved by modifying the definitions of SAFE-1 and SAFE-2 in [de Alfaro *et al.*, 2007] using the following definitions:

$$\mathsf{DAPre}_1^1(U) = \{v \in V_1 \mid \exists a \in A \text{ s.t. } \Delta(v, a) \in U\},$$
$$\mathsf{DAPre}_1^2(U) = \{v \in V_2 \mid \forall b \in M(v) \text{ s.t. } \Delta(v, b) \in U\},$$
$$\mathsf{DAPre}_2^1(U) = \{v \in V_1 \mid \forall a \in A \text{ s.t. } \Delta(v, a) \in U\},$$
$$\mathsf{DAPre}_2^2(U) = \{v \in V_2 \mid \forall b \in M(v) \text{ s.t. } \Delta(v, b) \in U\}.$$

The Alg. 2 works as follows. It starts with $Z_0 = \mathsf{Win}_1(A_1) \times \Gamma$ from where P1 has an ASW strategy to win the game no matter of P2's perception. Then it iteratively expands the set by invoking SAFE-2 followed by SAFE-1 until a fixed-point is reached. The SAFE-1 sub-routine computes the largest subset Y of the input set U, such that P1 has a strategy to restrict the game indefinitely within Y. SAFE-2 sub-routine computes the largest subset Y of the input set U, such that given his current (mis)perception, P2 can restrict the game indefinitely within Y. Here, it is important to note that P2 chooses his actions based on his perceptual game $\mathcal{G}(X)$, and not the hypergame. Only P1 knows the hypergame because she is aware of P2's misperception. As a consequence, before reaching the fixed-point, SAFE-2 might include states from which P2 may not have a strategy to indefinitely restrict P1 from reaching Z_0, *i.e.* P1 may have a DASW strategy from these states. However, after reaching the fixed-point, say in the iteration k, we observe that all DASW states are included in Z_k. A DASW strategy can then be computed based on the proof of Thm. 1. Let us now revisit the Example 3 to understand Alg. 2.

Example 4 (Example 3 contd.)**.** *Consider the hypergame graph as shown in Fig. 3. Recall from Example 1 that Almost-Sure Winning (ASW) region is* $\mathsf{Win}_1(A_1) = \{s_0, s_1\}$*, therefore, we have* $Z_0 = \{(s_0, X_2), (s_1, X_2), (s_1, X_1)\}$ *(we omit* (s_0, X_1) *as it is unreachable). The perceptually permissive actions for P2 are* $M((s_2, X_1)) = \{b_1, b_2\}$ *and* $M((s_2, X_2)) = \{b_2\}$*.*

Iteration 1 of DASW. *The first step is to compute* C_0*, i.e. the subset of* $V \setminus Z_0$ *which P2 perceives to be safe for himself. The* SAFE-2 *sub-routine takes 3 iterations to reach a fixed-point, at the end of which* $C_0 = \{(s_2, X_2), (s_3, X_2)\}$*. The next step is to compute* Z_1*, which the largest subset of* $V \setminus C_0$ *in which P1 can stay indefinitely. The* SAFE-1 *sub-routine takes 2 iterations to reach a fixed point. In its first iteration,* DAPre_1^1 *adds a state* (s_3, X_1) *and* DAPre_1^2 *adds a state* (s_2, X_1) *to* Z_1*. The interesting observation here is that* (s_2, X_1) *is added because the actions* b_1 *and* b_2 *are perceptually permissive actions for P2, both of which lead to a state in* $V \setminus C_0$*.*

Algorithm 2 Deceptive Almost-Sure Winning Region for P1

```
function DASW($\mathcal{H}$)
    $Z_0 = \mathsf{Win}_1(A_1) \times \Gamma$
    repeat
        $C_k = \text{SAFE-2}(V \setminus Z_k)$
        $Z_{k+1} = \text{SAFE-1}(V \setminus C_k)$
    until $Z_{k+1} = Z_k$
    return $\mathsf{DASWin}_1 = Z_k$
end function
function SAFE-$i$($U$)
    $Y_0 = U$
    repeat
        $W_1 = \mathsf{DAPre}_i^1(Y_k)$
        $W_2 = \mathsf{DAPre}_i^2(Y_k)$
        $Y_{k+1} = Y_k \cap (W_1 \cup W_2)$
    until $Y_{k+1} = Y_k$
    return $Y_k$
end function
```

Iteration 2 of DASW. *The fixed-point of* DASW *algorithm is reached in this iteration with* $Z_2 = \{(s_0, X_2), (s_1, X_2), (s_1, X_1), (s_2, X_1), (s_3, X_1)\}$*. The states* (s_2, X_1) *and* (s_3, X_1) *are idenitifed as the DASW states for P1.*

Using intuition from Example 4 with the observation that $\mathsf{Win}_1(A_1) \subseteq \mathsf{DASWin}_1 \lfloor_S$ holds for every hypergame $\mathcal{H}$ by definition, we formalize our first key result. It establishes that using action deception could be advantageous to P1.

Proposition 1. *There exists a hypergame* $\mathcal{H}$ *for which* $\mathsf{Win}_1(A_1) \subsetneq \mathsf{DASWin}_1 \lfloor_S$ *.*

Next, we proceed to prove the correctness of Alg. 2 by showing that from every state in DASWin_1, we can construct a DASW strategy for P1 to ensure a visit to final states with probability one. We first prove two lemmas.

Lemma 1. *In the* i*-th iteration of Alg. 2, P1 has a strategy to restrict the game indefinitely within* Z_i*, for all states in* Z_i*.*

Proof. $(v \in V_2)$. For a P2's state in Z_i, every state $v' = \Delta(v, b)$ for a perceptually permissive action $b \in \mu(v)$ of P2 is in Z_i, by definition of DAPre_1^2. Hence, no action of P2 at any state $v \in Z_i$ can lead the game state outside Z_i.

$(v \in V_1)$. For every P1's state in Z_i, there exists an action $a \in A$ such that the successor $v' = \Delta(v, a)$ is in Z_i, by definition of DAPre_1^1. Hence, P1 always has an action, consequently a strategy, to stay within Z_i. □

Lemma 2. *For every state* $v \in Z_{i+1} \setminus Z_i$ *added in the* i*-th iteration of Alg. 2, there exists an action that leads into* Z_i*.*

Proof. Given any state $v \in V$ at the beginning of the i-th iteration, observe that it would belong to either C_{i-1}, Z_i or $V \setminus (C_{i-1} \cup Z_i)$. We will prove the statement by showing that the every new state added to Z_{i+1} has at least one transition into Z_i.

Consider i-th iteration of Alg. 2. The sub-routine SAFE-2 will add a P1 state $v \in V_1 \setminus Z_i$ to C_i if all the actions of P1 stay within $V \setminus Z_i$. Similarly, SAFE-2 will include a P2 state $v \in V_2 \setminus Z_i$ in C_i if all perceptually permissive actions of P2 lead to a state within $V \setminus Z_i$. Therefore, a state that is not included in C_i must have at least one action leading outside $V \setminus Z_i$, *i.e.* entering Z_i. In the next step, the sub-routine SAFE-1 may add new states to Z_{i+1} from the set $V \setminus C_i$. But, all states in $V \setminus C_i$ have an action entering Z_i. Hence, all new states added to Z_{i+1} satisfy the statement. □

Lma. 2 shows that P1 has a strategy to reach Z_i from a state added to Z_{i+1} in one-step. However, this is not true for P2. From a P2 state in Z_{i+1}, there exists a positive probability to reach Z_i because of Assumption 2. In the next theorem, we prove a stronger statement that from every state in Z_{i+1}, P1 can reach not only Z_i but also Z_0 with probability one.

Theorem 1. *From every state $v \in \mathsf{DASWin}_1$, P1 has a DASW strategy to satisfy φ.*

Proof. For any $v \in Z_i$, $i > 1$, P1 has a strategy to stay within Z_i indefinitely, by Lma. 1. Furthermore, by Lma. 2, the probability of reaching to a state $v' \in Z_{i-1}$ from v is strictly positive. Thus, given a run of infinite length, the probability of reaching Z_{i-1} from Z_i is one. By repeatedly applying this argument, the probability of reaching Z_0 from Z_i is one. □

The DASW strategy can be constructed based on the proof of Thm. 1. At a P1 state $v \in V_1$, if $i \geq 1$ is the smallest integer such that $v \in Z_i$, then $\pi(v) = \{a \in A_1 \mid v' = \Delta(v, a)$ and $v' \in Z_{i-1}\}$ is the DASW strategy of P1 at v. Given $\pi(v)$ is a set, P1 can select any action from this set. We also state the following two important corollaries (proofs omitted due to space) that follow from Prop. 1 and Lma. 2.

Corollary 1.1. *For every $i \geq 0$, we have $Z_i \subseteq Z_{i+1}$.*

Corollary 1.2. *The projection of DASW region onto the game states is a superset of the ASW region.*

Following the approach used in [de Alfaro *et al.*, 2007], we note the complexity of Alg. 2 is quadratic in the size of V.

5 An Illustrative Example

We present a robot motion planning example over a 4×4 gridworld, shown in Fig. 4, to illustrate how a robot (P1) may use action deception in presence of an adversary (P2). The objective of the robot is to visit the two cells $(3, 1)$ and $(3, 3)$ containing the flags, while the task of the adversary is to prevent this. The readers familiar with Linear Temporal Logic (LTL) may recognize the above objective as a co-safe LTL specification $\Diamond G_1 \wedge \Diamond G_2$. The action set of the robot is $A_1 = \{$N, E, S, W, NE, NW, SW$\}$ while that of adversary is $A_2 = \{$N, E, S, W$\}$, where N, E, S, W stand for north, east, south and west. At the start of the game, the adversary has incomplete information about the robot's action set as $X_0 = \{$N, E, S, W$\}$. When the adversary observes the robot performing any of the actions from $\{$NE, NW, SW$\}$, he updates his perception to $X_1 = A_1$.

A game on graph representing above scenario can be constructed using the product operation given in [Baier and Katoen, 2008, Def. 4.16]. Every game state is a tuple $(x_1, y_1, x_2, y_2, t, q)$ where x_i, y_i for $i = 1, 2$ denote the cell that P1 and P2 occupy, t represents the player who chooses the next move and q denotes a state of a deterministic finite automaton that keeps track of the progress P1 has made towards completion of her objective. The resulting game has $4^4 \times 2 \times 4 = 2048$ states. We mark the states where P1 or P2 collide with an obstacle or with each other as the losing states for both players and, therefore, any action that leads to such states is disabled. Given the game on graph, a hypergame graph is constructed according to Def. 4. The hypergame graph has $2048 \times 2 = 4096$ states because the adversary has two information states; X_0 and X_1.

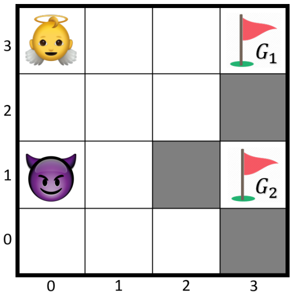

Figure 4: A game between P1 (angel) and P2 (demon).

When Alg. 2 is applied to the above hypergame graph, 2106 out of 4096 states are identified as DASW states. The projection of DASW states onto game state space results in 1172 states, while the ASW region has the size of 934 states. This means that $1172 - 934 = 238$ game states that were not almost-sure winning for P1 became winning for her, when P1 uses the DASW strategy.

6 Discussion and Conclusion

In this paper, we introduce a hypergame model to represent the interactions between two players with asymmetric information about their action capabilities. Given this model, we present an algorithm to synthesize action-deceptive strategies in a two-player turn-based zero-sum reachability game, where P2 starts with incomplete information about the P1's action capabilities. The synthesized strategy has two desirable properties. First, the DASW strategy is guaranteed to satisfy the reachability objective with probability one. Second, it is at least as powerful as the ASW strategy, because the DASW region is a superset of the ASW region. It worth noting that another game model of asymmetric information is Bayesian game, in which players have a Bayesian view of the types of others given their incomplete information. Our choice of hypergame model over Bayesian game model is motivated by the fact that a hypergame does not require *consistency of priors* that is assumed by Bayesian games [Halpern, 2002; Morris, 1995], and it also allows us to consider non-Bayesian players. Future work will focus on investigating action deception within hypergames with temporal logic payoffs and partial observations–that is, players have imperfect information about the history of games.

Acknowledgements

This material is based upon work supported by the Defense Advanced Research Projects Agency (DARPA) under Agreement No. HR00111990015.

References

[Baier and Katoen, 2008] Christel Baier and Joost-Pieter Katoen. *Principles of model checking*. MIT press, 2008.

[Bennett, 1977] PG Bennett. Toward a theory of hypergames. *Omega*, 5(6):749–751, jan 1977.

[Bernet *et al.*, 2002] Julien Bernet, David Janin, and Igor Walukiewicz. Permissive strategies: from parity games to safety games. *RAIRO - Theoretical Informatics and Applications*, 36(3):261–275, 2002.

[Bloem *et al.*, 2012] Roderick Bloem, Barbara Jobstmann, Nir Piterman, Amir Pnueli, and Yaniv Sa'ar. Synthesis of reactive(1) designs. *Journal of Computer and System Sciences*, 78(3):911 – 938, 2012. In Commemoration of Amir Pnueli.

[de Alfaro *et al.*, 2007] Luca de Alfaro, Thomas A Henzinger, and Orna Kupferman. Concurrent reachability games. *Theoretical Computer Science*, 386(3):188–217, 2007.

[Ettinger and Jehiel, 2010] David Ettinger and Philippe Jehiel. A theory of deception. *American Economic Journal: Microeconomics*, 2(1):1–20, February 2010.

[Fainekos *et al.*, 2009] Georgios E. Fainekos, Antoine Girard, Hadas Kress-Gazit, and George J. Pappas. Temporal logic motion planning for dynamic robots. *Automatica*, 45(2):343 – 352, 2009.

[Gharesifard and Cortés, 2014] Bahman Gharesifard and Jorge Cortés. Stealthy deception in hypergames under informational asymmetry. *IEEE Trans. Syst. Man, Cybern. Syst.*, 44(6):785–795, 2014.

[Gutierrez *et al.*, 2015] Christopher N Gutierrez, Saurabh Bagchi, H Mohammed, and Jeff Avery. Modeling Deception In Information Security As A Hypergame–A Primer. In *Proc. 16th Annu. Inf. Secur. Symp.*, page 41. CERIAS-Purdue University, 2015.

[Halpern, 2002] Joseph Y. Halpern. Characterizing the common prior assumption. *Journal of Economic Theory*, 106(2):316 – 355, 2002.

[Harsanyi, 1967] John C. Harsanyi. Games with Incomplete Information Played by "Bayesian" Players, I–III Part I. The Basic Model. *Management Science*, 1967.

[Kovach, 2016] Nicholas S Kovach. A Temporal Framework for Hypergame Analysis of Cyber Physical Systems in Contested Environments. Technical report, Air Force Institute of Technology, 2016.

[Kulkarni and Fu, 2019] Abhishek Ninad Kulkarni and Jie Fu. Opportunistic synthesis in reactive games under information asymmetry. *CoRR*, abs/1906.05847, 2019.

[Mazala, 2002] René Mazala. *Infinite Games*, pages 23–38. Springer Berlin Heidelberg, Berlin, Heidelberg, 2002.

[Morris, 1995] Stephen Morris. The common prior assumption in economic theory. *Economics and Philosophy*, 11(2):227–253, 1995.

[Pnueli and Rosner, 1989] A Pnueli and R Rosner. On the synthesis of a reactive module. pages 179–190, 1989.

[Ramadge and Wonham, 1989] P. J. G. Ramadge and W. M. Wonham. The control of discrete event systems. *Proceedings of the IEEE*, 77(1):81–98, Jan 1989.

[Rasmusen, 1989] Eric Rasmusen. *Games and information: An introduction to game theory*. Number 519.3/R22g. Blackwell Oxford, 1989.