Obstruction Alternating-time Temporal Logic: a Strategic Logic to Reason about Dynamic Models

Davide Catta
Università di Napoli, Federico II
Naples, Italy
davidecatta2@gmail.com

Jean Leneutre
Télécom Paris
Paris, France
jean.leneutre@telecom-paris.fr

Vadim Malvone
Télécom Paris
Paris, France
vadim.malvone@telecom-paris.fr

Aniello Murano
Università di Napoli, Federico II
Naples, Italy
aniello.murano@unina.it

ABSTRACT

Multi-Agent Systems (MAS) operating within dynamic models have been extensively studied in various domains, including cybersecurity and planning. In this paper, we introduce a dedicated logic for analyzing a specific category of MAS that involve strategic objectives within dynamic models. Within these MAS, there exists an agent known as the "Demon", which possesses the capability to modify the MAS model itself, while other agents operate as traditional MAS entities. We demonstrate that the model-checking problem for our logic is solvable in polynomial time. Furthermore, we showcase how this logic can be effectively employed to articulate significant properties within the realm of cybersecurity.

KEYWORDS

Formal Verification, Dynamic Models, Concurrent Games

ACM Reference Format:
Davide Catta, Jean Leneutre, Vadim Malvone, and Aniello Murano. 2024. Obstruction Alternating-time Temporal Logic: a Strategic Logic to Reason about Dynamic Models. In *Proc. of the 23rd International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2024), Auckland, New Zealand, May 6 – 10, 2024*, IFAAMAS, 10 pages.

1 INTRODUCTION

System Verification. Over the course of the past half-century, researchers have been driven to create a multitude of verification techniques for software and hardware systems, all with the common aim of ensuring these systems align with their intended specifications [34, 40, 45]. Within the realm of formal verification, *model checking* [15] has emerged as a verification technique due to its versatility and conceptual simplicity. In order to verify whether a system exhibits specific characteristics, a system's representation is provided by a mathematical model $\mathfrak{M}$ (typically a labeled directed graph). The desired property is expressed using a formula φ written in some logical system characterized by precise mathematical semantics, such as the temporal logics LTL [39] and CTL [16]. Assuming that $\mathfrak{M}$ faithfully represents the target system and φ aptly encapsulates the desired property, the sole remaining task revolves around confirming that $\mathfrak{M}$ complies with the semantic clauses dictating the truthfulness of φ. The initial applications of model checking primarily focused on closed systems, which are defined by the fact that their behavior is entirely determined by their internal states. Unfortunately, model checking techniques developed to address closed systems proved to be largely impractical, since real-world systems are often open and involve ongoing interactions with other systems. To address this challenge, model checking has been extended to Multi-Agent Systems (MAS). Multi-agent systems represent the behavior of two or more rational agents engaging in interactions, be it cooperative or adversarial, all with the goal of achieving a specific objective [26]. Typically, this behavior is captured through a combination of modal logic and game theory, where agents act as players in games played on directed graphs and their objectives are defined using logics for strategic reasoning, such as Alternating-time Temporal Logic (ATL) [3] and Strategy Logic [35].

System Security. In recent years, digital systems have become increasingly complex and dynamic. As a result, it is increasingly difficult to ensure the security of these systems when facing attackers able to exploit this complexity. This is confirmed by the emergence in recent years of sophisticated attacks such as Advanced persistent threats (APTs) [29], which are complex attacks with a long life cycle and a high degree of stealth. They can exploit supply chain vulnerabilities (as in the case of the Stuxnet worm infecting Iranian Uranium enrichment centrifuges in 2010 [46]), or even "cascade effects" due to system interconnections (such as the BlackEnergy Trojan causing a power cut in Ukraine in December 2015 [27]). Other new coordinated attacks, such as Wormhole Attacks [23], requires the simultaneous action of several attackers located at different points in a wireless network. In this context, supervision is an essential security service, enabling protection mechanisms to be strengthened and any shortcomings to be remedied. Initially, system supervision mainly concerned with detection, management, analysis, and correlation of security events. Today, it encompasses risk analysis and security orchestration more generally, taking into account upstream information on current attacks and downstream

Proc. of the 23rd International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2024), N. Alechina, V. Dignum, M. Dastani, J.S. Sichman (eds.), May 6 – 10, 2024, Auckland, New Zealand. © 2024 International Foundation for Autonomous Agents and Multiagent Systems (www.ifaamas.org).

automation of the response to detected attacks. Active cybersecurity mechanisms enabling these tasks are currently available such as active Intrusion Detection System (also called Intrusion Prevention System) or Moving Target Defense mechanisms (MTDs) [17]. However, their deployment and configuration still remains critical. The aim is obviously to reduce the risks associated with new threats, while minimizing the impact on the system and legitimate users (in terms of performance or usability). In addition, there may be other constraints, such as the financial cost of the deployed mechanisms. In this context, it is essential to have methods and tools to define appropriate and optimal response strategies.

Our contribution. In this paper, we devise a logic, called Obstruction Alternating-time Temporal Logic (OATL), allowing one to reason about the existence of **reaction strategies** that can counter **all** possible identified threat scenarios over a given cyberphysical system. We model attack scenarios as instances of multi-agent systems. In all the previously mentioned logics designed to analyze the strategic abilities of agents, the game model where the players operate is typically regarded as a static entity. In this view, the players' actions determine their positions within the arena (the graph representing the system) but do not modify the arena's structure. Conversely, dynamic game models, where games evolve or change over time, have received significant attention across various domains, including cybersecurity, planning, and normative systems [1, 2, 11, 12, 36, 43].

Let's suppose that the game environment can change at the will of an intelligent entity, which we will refer to as the Demon. Specifically, let's assume that the Demon's power is to prevent all players from coordinating to perform a particular action. Furthermore, let's also assume that the Demon's powers are not limitless: he can prevent a specific action from taking place, but only at a certain cost. Furthermore, he cannot prevent all possible coordinated actions from occurring at a given moment. We want to answer to the following question. Assume that $\mathfrak{S}$ ranges over strategies of the Demon and Σ over strategies of some coalition of agents A.

$\forall_{\mathfrak{S}}.\exists_{\Sigma}$) Can the coalition A devise a collective strategy to achieve their goals, whatever the evolution of the game-enviroment dictated by a strategy of the Demon may be?

$\exists_{\mathfrak{S}}.\forall_{\Sigma}$) Can the Demon act in such a way as to prevent the agents from having a strategy that guarantees the achievement of their objective?

2 SYNTAX AND SEMANTICS

In this section, we define the syntax and semantics of Obstruction Alternating-time Temporal Logic (OATL). First, let us fix some notation and terminology that will be used along the paper.

Preliminary notions. If V is a set and $U \subseteq V$, we denote by $\overline{U}$ the complementary $V \setminus U$ of U in V. If v is a (finite or infinite) sequence over U, we denote by $|v|$ its length (which is ω if v is infinite), by v_i its i-th element, by $v_{\leq_i}$ the finite prefix $v_1, \ldots, v_i$ of v and by $v_{\geq i}$ the (possibly infinite) suffix of v starting at v_i. If v is a finite sequence, $last(v)$ denotes the last element $v_{|v|}$ of v. Given two sequences v and u, we write $v \sqsubset u$ when v is a strict prefix of u.

2.1 OATL Syntax

We now introduce the syntax of our logic.

Definition 1. *Given a countable set* Ap *of atomic propositions, and a finite (non-empty) set of agents* Ag*, state φ and path θ formulas are defined by mutual induction using the following grammar:*

$$\varphi ::= \top \mid p \mid \neg\varphi \mid \varphi \wedge \varphi \mid [\![A]\!]^{\dagger}_{n}\theta$$
$$\theta ::= \mathsf{X}\,\varphi \mid \varphi\,\mathsf{U}\,\varphi \mid \varphi\,\mathsf{R}\,\varphi$$

where $p \in$ Ap is any atomic proposition, A is any subset of Ag *(subset of* Ag *will sometimes be referred to as* coalitions*), and n (the grade) is any natural number bigger or equal than 0.*

In what follows, we use the letters p, q, r, (possibly indexed) to denote arbitrary atomic propositions, and the Greek φ, ψ, θ (possibly indexed) to denote arbitrary formulas. Formulae whose outmost operator is $[\![A]\!]^{\dagger}_{n}$ for some A and n, will be referred to as *strategic* formulae. The height $|\varphi|$ of a formula φ is the height of its construction tree. Formulae of OATL are all and only the state formulas.

2.2 OATL Semantics

We specify the meaning of OATL formulae by means of Concurrent Game Structures (CGSs for short). Intuitively, a CGS is a labeled directed graph that represents the possible evolution of a given Multi-Agent System with respect to simultaneous choices of actions of a group of (autonomous) agents. Both states and edges are labeled by members of two disjoints alphabets. States are labeled by atomic propositions. These atomic propositions represent the properties that are true at a given state. Each edge is labeled by a tuple, and each member of a given tuple represents an action that is available for a given agent at the source state of the edge. The formal definition follows.

Definition 2. *Given a set of atomic proposition* Ap *and a set of agents* Ag $= \{1, \ldots, k\}$*, a concurrent game structure over* Ap *and* Ag *is a tuple $\mathfrak{C} = \langle S, s_I, Act, P, T, \mathcal{V} \rangle$ where:*

- *S is a non-empty set of states and s_I is a distinguished state dubbed* initial state.
- *Act is a finite, non-empty set of actions. We denote by $\mathcal{J}$ the set of maps from the set of agents to the set of actions. Elements of $\mathcal{J}$ will be called* joint actions *and denoted by bold lowercase letters, i.e.,* $\mathbf{a}, \mathbf{b}, \mathbf{c}$*, etc.*
- *$P : S \times$ Ag $\longrightarrow 2^{Act} \setminus \emptyset$ is the protocol function which assigns a non empty-subset of actions $P(s, i)$ of Act to any agent i and state s. The set $P(s, i)$ represents the set of actions that are available at the state s to the agent i.*
- *$T : S \times \mathcal{J} \longrightarrow S$ is the (partial) transition function such that T is defined for a state s and a joint action* $\mathbf{a}$ *only if for every $i \in$ Ag, $\mathbf{a}(i) \in P(s, i)$.*
- *Finally, $\mathcal{V} : S \longrightarrow 2^{\mathsf{Ap}}$ is the valuation function, which assign to any state s a (possibly empty) subset of* Ap.

An obstruction model $\mathfrak{M}$ is a pair $\langle \mathfrak{C}, \$ \rangle$ where $\mathfrak{C}$ is a CGS and $\$: S \times \mathcal{J} \longrightarrow \mathbb{N}^+$ is the (partial) cost function. Such function associates to any state s and joint action $\mathbf{a}$ *such that $T(s, \mathbf{a})$ is defined, a positive natural number $\$(s, \mathbf{a})$.*

A model is a CGS provided with a function that assigns a cost (a positive natural number) to any pair composed of a state and a joint action that defines a transition from the given state. Thus, a model is a labeled, directed weighed graph.

Given an obstruction model $\mathfrak{M} = \langle \mathfrak{C}, \$ \rangle$, a **path** is an infinite sequence of states of the CGS, $\pi = \pi_1, \pi_2, \ldots$ such that: for any $i \in \mathbb{N}^+$ there is a joint action $\mathbf{a} \in \mathcal{J}$ such that $\pi_{i+1} = T(\pi_i, \mathbf{a})$. We will denote paths by the letters π, ρ, τ, and λ. A finite sequence of states h is a **history** iff there is a path π such that $h = \pi_{\leq i}$ for some positive natural number i. We will denote by H the set of all histories over a given model $\mathfrak{M}$. Given a state s of the model $\mathfrak{M}$, we denote by $\mathcal{J}(s)$ the set of joint actions that defines a transition from s, that is $\mathcal{J}(s) = \{\mathbf{a} \in \mathcal{J} \mid T(s, \mathbf{a}) = s' \text{ for some } s' \in S\}$. If A is a coalition, a **A-action** available at s is a function $f : A \longrightarrow Act$ such that $f(i) \in P(s, i)$ for each $i \in A$. If f and g are actions available at s for the coalitions A and B we say that g **extends** f, if $A \subseteq B$ and $f(i) = g(i)$ for each $i \in A$. We write $f \preceq g$ if g extends f. $\mathcal{F}(A, s)$ denotes the set of A-actions available at s and $\mathcal{F}(A, \mathfrak{M})$ is $\bigcup_{s \in S} \mathcal{F}(A, s)$. If $f \in \mathcal{F}(A, \mathfrak{M})$ then $f_s^{\preceq} = \{\mathbf{a} \in \mathcal{J}(s) \mid f \preceq a\}$. Note that, as mentioned before, we will consider a special agent, which is outside any coalition of agents, who has the power of modifying the model itself. This special agent, that we call the Demon, acts rationally, i.e., he can came-up with strategies to modify the model. As already said, we can see a model of OATL as a directed weighted graph in which arcs are labeled by joint actions. Given a history h, a demonic strategy selects a subset of arcs that are adjacent to $last(h)$ and whose sum of weights does not surpass a given threshold. The arcs selected by the demonic strategies are temporarily erased from the set of arcs that coalitions can select, in this sense the structure of the graph is modified by the actions of the Demon. We formally define the notion of demonic strategy as follows.

Definition 3 (Demonic Strategy). *If $\mathfrak{M}$ is an obstruction model and $n \in \mathbb{N}$ is a natural number, a* ***demonic n-strategy*** *is a function $\mathfrak{S} : H \longrightarrow 2^{\mathcal{J}}$ that given an history h, returns a subset $A_{\mathfrak{S}}$ of $\mathcal{J}$ such that:*

(1) $A_{\mathfrak{S}} \subset \mathcal{J}(last(h))$,
(2) $(\sum_{\mathbf{a} \in A_{\mathfrak{S}}} \$(last(h), \mathbf{a})) \leq n$.

A path π is **compatible** with a demonic n-strategy $\mathfrak{S}$ ($\mathfrak{S}$-compatible for short) if for all $i \geq 1$ we have that $\pi_{i+1} = T(\pi_i, \mathbf{a})$ implies $\mathbf{a} \notin \mathfrak{S}((\pi_{\leq i}))$. Given a state s and a demonic n-strategy $\mathfrak{S}$, $\mathrm{Out}^{\dagger}(s, \mathfrak{S})$ denotes the set of paths whose first state is s and that are compatible with $\mathfrak{S}$.

Definition 4 (Strategy). *Given a model $\mathfrak{M}$ and a coalition A, an A-strategy for A (or simply A-strategy) is a function $\Sigma : H \longrightarrow \mathcal{F}(A, \mathfrak{M})$ that maps each history h to an A-action f such that $f \in \mathcal{F}(A, last(h))$.*

A path π is compatible with an A-strategy Σ for the coalition A (Σ-compatible for short) iff for every $i \geq 1$, we have that $\pi_{i+1} = T(\pi_i, \mathbf{a})$ implies $\Sigma(\pi_{\leq i}) \preceq \mathbf{a}$. We denote with $\mathrm{Out}(s, \Sigma)$ the set of all Σ-compatible paths whose first state is s.

Definition 5 (Satisfaction). *The satisfaction relation $\mathfrak{M}, s \models \varphi$ between a model $\mathfrak{M}$, a state s of $\mathfrak{M}$, and a state formula φ is defined as follows:*

- *$\mathfrak{M}, s \models \top$ always;*
- *$\mathfrak{M}, s \models p$ iff $p \in \mathcal{V}(s)$;*
- *$\mathfrak{M}, s \models \neg\varphi_1$ iff it is not the case that $\mathfrak{M}, s \models \varphi_1$ (denoted $\mathfrak{M}, s \not\models \varphi_1$);*
- *$\mathfrak{M}, s \models \varphi_1 \wedge \varphi_2$ iff $\mathfrak{M}, s \models \varphi_1$ and $\mathfrak{M}, s \models \varphi_2$;*
- *$\mathfrak{M}, s \models [\![A]\!]^{\dagger}_{n}\theta$ iff there is a demonic n-strategy $\mathfrak{S}$ such that for all A-strategy Σ if $\mathrm{Out}^{\dagger}(s, \mathfrak{S}) \cap \mathrm{Out}(s, \Sigma) \neq \emptyset$, then there is a $\pi \in \mathrm{Out}^{\dagger}(s, \mathfrak{S}) \cap \mathrm{Out}(s, \Sigma)$ such that $\mathfrak{M}, \pi \models \theta$.*

The satisfaction relation $\mathfrak{M}, \pi \models \varphi$ between a model $\mathfrak{M}$, a path π of $\mathfrak{M}$, and path formula θ is defined as follows:

- *$\mathfrak{M}, \pi \models \mathsf{X}\,\varphi$ iff $\mathfrak{M}, \pi_2 \models \varphi$*
- *$\mathfrak{M}, \pi \models \varphi_1 \,\mathsf{U}\, \varphi_2$ iff there is an $i \geq 1$ such that $\mathfrak{M}, \pi_i \models \varphi_2$ and $\mathfrak{M}, \pi_j \models \varphi_1$ for all $1 \leq j < i$;*
- *$\mathfrak{M}, \pi \models \varphi_1 \,\mathsf{R}\, \varphi_2$ iff either $\mathfrak{M}, \pi_i \models \varphi_2$ for all $i \geq 1$ or there is an $i \geq 1$ such that $\mathfrak{M}, \pi_i \models \varphi_1$ and $\mathfrak{M}, \pi_j \models \varphi_2$ for all $1 \leq j \leq i$.*

A OATL formula φ is true in a model $\mathfrak{M}$ iff $\mathfrak{M}, s_I \models \varphi$. Two formulas φ and ψ are equivalent (denoted by $\varphi \equiv \psi$) if for all models $\mathfrak{M}$ and state s of $\mathfrak{M}$, $\mathfrak{M}, s \models \varphi$ iff $\mathfrak{M}, s \models \psi$.

Remark that since any demonic n-strategy can select only a strict subset of the set of joint actions available at $last(h)$ for a given history h, we can never have that $\mathrm{Out}^{\dagger}(\mathfrak{S}, s) \cap \mathrm{Out}(\Sigma, s) = \emptyset$ for every A-strategy Σ.

3 CASE STUDY

In terms of cybersecurity risk management, the questions specified below are natural questions that arise in the risk identification or risk treatment phases. To illustrate this, in the following of this section, we will take into consideration the following generic cybersecurity scenario, where we consider:

- A system with a given set of assets under protection.This system could be for instance a network with a set of servers storing resources or providing the legitimate users with services. The physical assets of the system may feature known vulnerabilities that could not have been fixed for some reason. For instance, a given server on which runs a critical service that could not be stopped, uses an old Linux distribution version with a privilege escalation vulnerability that could be exploited by an attacker to gain full access privilege on the server.
- A set of legitimate users that use the system under protection. Their action on the system allow them for instance to request access to the servers.
- A set of attackers that can exploit vulnerabilities to launch an attack. This attack will be successful under some preconditions. These preconditions maybe related to the credentials that a subset of attackers already obtained on the system (knowledge of passwords, access to some servers, ...), but also to actions previously performed by legitimate user (access request to some resources,...). Combining such previous attack in a sequence, the attackers will be able to build an attack path that allows to reach a given objective: for instance obtaining the root privilege on a given critical server.
- A unique defender (a centralized defense system) whose objective is to counter the attack, that is prevent the coalition of attackers from reaching its goal. This defender has the power to react to an atomic attack performed by a coalition

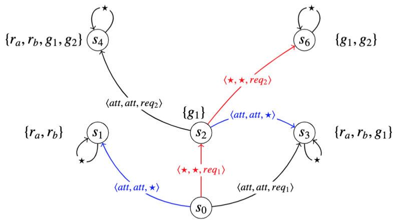
$\{r_a, r_b, g_1, g_2\}$
s_4
$\star$
s_6
$\{g_1, g_2\}$
$\star$
$\langle \star, \star, req_2 \rangle$
$\langle att, att, req_2 \rangle$
$\{g_1\}$
$\langle att, att, \star \rangle$
$\{r_a, r_b\}$
s_1
s_2
s_3
$\{r_a, r_b, g_1\}$
$\star$
$\star$
$\langle \star, \star, req_1 \rangle$
$\langle att, att, \star \rangle$
$\langle att, att, req_1 \rangle$
s_0

Figure 1: A model depicting the considered attack scenario. Joint actions labeling red edges have a cost of 3, joint actions labeling black edges have a cost of 2 while those labeling blue edges have a cost of 1. The symbol $\star$ stands for $\langle \star, \star, \star \rangle$

> of attackers, by dynamically deploying a predefined set of security countermeasures, in order to minimize the future risk. These newly deployed security countermeasures will remove one or several preconditions of atomic attacks (for instance by reconfiguring the firewall filtering rules). The deployment of these countermeasures will come with a cost (that could be related for instance to the resulting impact on the performance or usability of the system). The goal of the defender being also to minimize the impact a constraint on the cumulative costs of the countermeasures being deployed must also be satisfied.

As a toy example[1], consider the following scenario. Three users of a wireless network (call them Alice, Bob, and David) can modify their status inside the network (position and granted privileges) by either making request to the network or by attacking it. Two of the three users (Alice and Bob) are malevolent and aims at compromising the integrity of the network itself. To achieve their goal, they must be in a situation where Alice has been granted root access on a specific network server, Bob has been granted root on another server, and David has asked and obtained a specific resource from the network. Suppose that, in each instant, Alice and Bob can either do nothing or successfully make an attack on the network and obtaining root privilege on their wanted server. Suppose, moreover, that David can either make a specific request on the network or do nothing. Let r_a be the atomic proposition expressing that Alice is root of the needed server, r_b be the atomic proposition expressing that Bob is root of the needed server, and that g_1 and g_2 be the atomic proposition expressing the fact that David has been granted access to resources 1 and 2. Given these premises, a possible interaction between Alice, Bob, and David is depicted in Figure 1. Now, suppose that there is an intelligent defense mechanism on the network whose power is to temporally block users' collective actions. Suppose moreover that blocking these actions has a specific costs that can vary depending on the nature of each user's action. For instance, suppose that, in Figure 1 joint actions labeling blue edges have a cost of 1, joint actions labeling black edges have a cost of 2, while joint actions labeling red edges have a cost of 3. Suppose that the initial state of the model is s_0. Is there a strategy for the defender such that, for any strategy adopted by Alice and Bob, there exists at least one situation in which Alice and Bob are never in position to launch the fatal attack? By using OATL, we can express this property as follows: $\mathfrak{M}, s_0 \models \llbracket Alice, Bob \rrbracket^{\dagger}_{n}(\bot \,\mathsf{R}\, \neg(r_a \wedge r_b \wedge g_1))$ for some $n \in \mathbb{N}$. On this example, the minimum n for which $\llbracket Alice, Bob \rrbracket^{\dagger}_{n}(\bot \,\mathsf{R}\, \neg(r_a \wedge r_b \wedge g_1))$ holds is 3. The 3-demonic strategy $\mathfrak{S}$ selecting the joint action $\mathbf{a}$ such that $\mathbf{a}(Alice) = \star$, $\mathbf{a}(Bob) = \star$ and $\mathbf{a}(David) = req_1$ given any history h such that $last(h) = s_0$. The set of paths in $\mathrm{Out}^{\dagger}(s_0, \mathfrak{S})$ contains $s_0 \cdot s_1^{\omega}$ and $s_0 \cdot s_3^{\omega}$, $\{Alice, Bob\}$-strategies Σ such that $B = \mathrm{Out}^{\dagger}(s_0, \mathfrak{S}) \cap \mathrm{Out}(s_0, \Sigma) \neq \emptyset$ are those in which Alice and Bob both choose att on s_0. For all these strategies, there is a path (i.e., $s_0 \cdot s_1^{\omega}$) that satisfies $\bot \,\mathsf{R}\, \neg(r_a \wedge r_b \wedge g_1)$.

4 OATL PROPERTIES

In this section, we study the formal properties of OATL. In particular we show that, as in ATL, the set of states that satisfies a formula of the form $\llbracket A \rrbracket^{\dagger}_{n}(\varphi_1 \,\mathsf{U}\, \varphi_2)$ or $\llbracket A \rrbracket^{\dagger}_{n}(\varphi_1 \,\mathsf{R}\, \varphi_2)$ can be expressed as the fix-point of particular monotone functions.

Given a model $\mathfrak{M}$ and a formula φ, $\{\{\varphi\}\}^{\mathfrak{M}}$ denotes the set of states of $\mathfrak{M}$ that satisfies φ, that is $\{\{\varphi\}\}^{\mathfrak{M}} = \{s \in S \mid \mathfrak{M}, s \models \varphi\}$. Notice that we can omit $\mathfrak{M}$ when it is clear from the context. If $X \subset \mathcal{J}(s)$ for a given state s, we write $Cost(X)$ for $\sum_{\mathbf{a} \in X}(\$(\mathbf{a}, s))$. Given a state s, a coalition A, and an A-action f available at s, we let $next(f, s)$ denotes the set of states: $next(f, s) = \{s' \in S \mid s' = T(s, \mathbf{a})$ for some $\mathbf{a}$ s.t. $f \preceq \mathbf{a}\}$. If X is a set of states, $\mathcal{D}(s, A, X)$ denote the subset of $\mathcal{J}(s)$: $\mathcal{D}(s, A, X) = \{\mathbf{a} \in \mathcal{J}(s) \mid \exists f \in \mathcal{F}(A, \mathfrak{M})$ s.t.$f \preceq \mathbf{a}$ and $next(f, s) \subseteq \overline{X}\}$ And $Pre(X, A)$ denotes the set of states: $Pre(X, A) = \{s \in S \mid \exists f \in \mathcal{J}(s, A)$ s.t. $f \preceq \mathbf{a}$ and $T(s, \mathbf{a}) = s' \in X\}$. Finally: $\dagger(n, A, X) = \{s \in S \mid s \in Pre(A, X)$ and $Cost(\mathcal{D}(s, A, X)) \leq n\}$. Intuitively, a state s belongs to $\dagger(n, A, X)$ when the Demon can ensure that all action available to A that inevitably lead to a state that is not in X can be erased.

Proposition 1. *For every formula φ and model $\mathfrak{M}$ we have:*

$$s \in \{\{\llbracket A \rrbracket^{\dagger}_{n} \mathsf{X}\, \varphi\}\} \textbf{ iff } s \in \dagger(n, A, \{\{\varphi\}\})$$

Proof. We only prove the $(\rightarrow)$ direction, as the converse is trivial. We proceed by contraposition. Suppose that $s \notin \dagger(n, A, \{\{\varphi\}\})$. If $s \notin Pre(A, \{\{\varphi\}\})$ the result is clear. Otherwise suppose that $s \in Pre(A, \{\{\varphi\}\})$ and consider an arbitrary demonic n-strategy $\mathfrak{S}$. By hypothesis, we must have that $Cost(\mathcal{D}(s, A, \{\{\varphi\}\})) > n$ and $Cost(\mathfrak{S}(s)) \leq n$ we deduce that there is a subset Y of $\mathcal{J}(s)$ and a $f \in \mathcal{F}(A, s)$ such that $f \preceq \mathbf{a}$ for every $\mathbf{a} \in Y$, $T(s, \mathbf{a}) \notin \{\{\varphi\}\}$, and $Y \not\subseteq \mathfrak{S}(s)$. From the fact that $Y \not\subseteq \mathfrak{S}(s)$, it follows that the coalition can always choose an appropriate $f \in \mathcal{F}(A, s)$ and A-strategy Σ, such that $\Sigma(s) = f$ for which we will have that $\pi \in \mathrm{Out}(s, \Sigma)$ implies $\pi_2 \notin \{\{\varphi\}\}$. But this means, by definition of satisfaction, that $\mathfrak{M}, s \not\models \llbracket A \rrbracket^{\dagger}_{n} \mathsf{X}\, \varphi$ as we wanted. □

Theorem 1. *For every formula φ and ψ the following equivalences hold:*

(1) $\llbracket A \rrbracket^{\dagger}_{n}(\varphi \,\mathsf{U}\, \psi) \equiv \psi \vee (\varphi \wedge \llbracket A \rrbracket^{\dagger}_{n} \mathsf{X} \llbracket A \rrbracket^{\dagger}_{n}(\varphi \,\mathsf{U}\, \psi))$
(2) $\llbracket A \rrbracket^{\dagger}_{n}(\varphi \,\mathsf{R}\, \psi) \equiv \psi \wedge (\varphi \vee \llbracket A \rrbracket^{\dagger}_{n} \mathsf{X} \llbracket A \rrbracket^{\dagger}_{n}(\varphi \,\mathsf{R}\, \psi))$

Proof. We give a detailed proof of (1). For the $(\rightarrow)$-direction, let $\mathfrak{M}$ be any model and s any of its states, and suppose that $s \in \{\{\llbracket A \rrbracket^{\dagger}_{n}(\varphi \,\mathsf{U}\, \psi)\}\}$. By the definition of satisfaction, this means that there is a demonic n-strategy $\mathfrak{S}$ such that for every A-strategy if $\mathrm{Out}^{\dagger}(s, \mathfrak{S}) \cap \mathrm{Out}(s, \Sigma) \neq \emptyset$ then there is a $\pi \in \mathrm{Out}^{\dagger}(s, \mathfrak{S}) \cap \mathrm{Out}(s, \Sigma)$

[1] Examples of such real case studies, at least for the attack part, can be found in [38]

such that $\pi_j \in \{\{\psi\}\}$ and $\pi_i \in \{\{\varphi\}\}$ for each $1 \leq i < j$. If $j = 1$, then we can conclude, otherwise $s \in \{\{\varphi\}\}$ and we must show that $s \in \{\{\llbracket A \rrbracket^{\dagger}_n \mathsf{X}\, \llbracket A \rrbracket^{\dagger}_n (\varphi \,\mathsf{U}\, \psi)\}\}$. Consider an arbitrary A-strategy Δ and let π^{Δ} be a path that satisfies $(\varphi \,\mathsf{U}\, \psi)$ given the demonic n-strategy $\mathfrak{S}$. It is clear that $\pi^{\Delta}_{\geq 2} \in \mathrm{Out}^{\dagger}(\pi^{\Delta}_2, \mathfrak{S}) \cap \mathrm{Out}(\pi^{\Delta}_2, \Delta)$ and, since $\pi^{\Delta}_1 \notin \{\{\psi\}\}$, that $\mathfrak{M}, \pi^{\Delta}_{\geq_2} \models \varphi \,\mathsf{U}\, \psi$. Moreover $\pi^{\Delta} \in \mathrm{Out}^{\dagger}(s, \mathfrak{S}) \cap \mathrm{Out}(s, \Delta)$. Since Δ was arbitrary, we can conclude that $\mathfrak{M}, s \models \llbracket A \rrbracket^{\dagger}_n \mathsf{X}\, \llbracket A \rrbracket^{\dagger}_n (\varphi \,\mathsf{U}\, \psi)$ as we wanted. For the converse direction. Suppose that $s \in \{\{\psi \vee (\varphi \wedge \llbracket A \rrbracket^{\dagger}_n \mathsf{X}\, \llbracket A \rrbracket^{\dagger}_n (\varphi \,\mathsf{U}\, \psi))\}\}$. If $s \in \{\{\psi\}\}$ then we are done. Otherwise $s \in \{\{\varphi\}\}$. From the fact that s satisfies $\llbracket A \rrbracket^{\dagger}_n \mathsf{X}\, \llbracket A \rrbracket^{\dagger}_n (\varphi \,\mathsf{U}\, \psi)$, we obtain that there is a strategy $\mathfrak{S}_1$ such that given any A-strategy Δ, we have that $\mathfrak{M}, \pi_2 \models \llbracket A \rrbracket^{\dagger}_n (\varphi \,\mathsf{U}\, \psi)$ for some $\pi \in \mathrm{Out}^{\dagger}(s, \mathfrak{S}) \cap \mathrm{Out}(s, \Sigma)$. By applying again the definition of satisfaction, we obtain that there is a demonic n-strategy $\mathfrak{S}^{\pi}_{\Delta}$ such that given any A-strategy Σ, we have that $\mathfrak{M}, \rho \models (\varphi \,\mathsf{U}\, \psi)$ for some $\rho \in \mathrm{Out}^{\dagger}(\pi_2, \mathfrak{S}^{\pi}_{\Delta}) \cap \mathrm{Out}(\pi_2, \Sigma)$. Consider the n-demonic $\mathfrak{S}^{\star}$ defined by:

$$\mathfrak{S}^{\star}(h) = \begin{cases} \mathfrak{S}_1(h) & \text{if } h = s \\ \mathfrak{S}^{\pi}_{\Delta}(h') & \text{if } h = s \cdot h' \text{ and } h' \sqsubset \tau \text{ for } \tau \in \mathrm{Out}^{\dagger}(\pi_2, \mathfrak{S}^{\pi}_{\Delta}) \\ & \text{and } \pi \in \mathrm{Out}^{\dagger}(s, \mathfrak{S}_1) \cap \mathrm{Out}(s, \Delta) \\ \emptyset & \text{otherwise} \end{cases}$$

That is, $\mathfrak{S}^{\star}$ is obtained by composing $\mathfrak{S}_1$ with the appropriate $\mathfrak{S}^{\pi}_{\Delta}$ for any A-strategy Δ. By construction, we obtain that given any A-strategy Δ, if $\mathrm{Out}(s, \Delta) \cap \mathrm{Out}^{\dagger}(s, \mathfrak{S}^{\star}) \neq \emptyset$ then there is a path $\pi \in \mathrm{Out}^{\dagger}(s, \mathfrak{S}^{\star}) \cap \mathrm{Out}(s, \Delta)$ such that $\mathfrak{M}, \pi \models \varphi \,\mathsf{U}\, \psi$ and we can thus conclude. □

Let $\mathfrak{M}$ be a model and φ, ψ be two formulae. Consider the two functions $\mathsf{U}^{\mathrm{n}}{}_{A,\varphi,\psi}$ and $\mathsf{R}^{\mathrm{n}}{}_{A,\varphi,\psi}$ from 2^S to itself defined by:

$$\mathsf{U}^n_{A,\varphi,\psi}(X) = \{\{\psi\}\}^{\mathfrak{M}} \cup (\{\{\varphi\}\}^{\mathfrak{M}} \cap \dagger(n, A, X)) \qquad (1)$$

$$\mathsf{R}^n_{A,\varphi,\psi}(X) = \{\{\psi\}\}^{\mathfrak{M}} \cap (\{\{\varphi\}\}^{\mathfrak{M}} \cup \dagger(n, A, X)) \qquad (2)$$

we can prove the following.

Theorem 2. *For every model $\mathfrak{M}$ and pair of formulae φ and ψ:*

(1) $\{\{\llbracket A \rrbracket^{\dagger}_n (\varphi \,\mathsf{U}\, \psi)\}\}^{\mathfrak{M}}$ *is the least fix-point of* $\mathsf{U}^n_{A,\varphi,\psi}$;

(2) $\{\{\llbracket A \rrbracket^{\dagger}_n (\varphi \,\mathsf{R}\, \psi)\}\}^{\mathfrak{M}}$ *is the greatest fix-point of* $\mathsf{R}^n_{A,\varphi,\psi}$.

Proof. We only prove (2). In virtue of the Theorem 1, it is clear that $\{\{\llbracket A \rrbracket^{\dagger}_n (\varphi \,\mathsf{R}\, \psi)\}\}$ is a fix-point of the function in Equation 2. To prove that $X = \{\{\llbracket A \rrbracket^{\dagger}_n (\varphi \,\mathsf{R}\, \psi)\}\}$ is the greatest fix-point of the function, we consider another fix-point Y and show that $Y \subseteq X$.

If $Y = \emptyset$ there is nothing to do. Otherwise, let $y \in Y$: we have that $y \in \{\{\psi\}\}$ and either $y \in \{\{\varphi\}\}$ or $y \in \dagger(n, A, Y)$. If this last case holds, we have that $Cost(\mathcal{D}(y, A, Y)) \leq n$. We define a strategy $\mathfrak{S}$:

$$\mathfrak{S}(h) = \begin{cases} \mathcal{D}(last(h), A, Y) & \text{if } last(h) \in \{\{\psi\}\} \cap \dagger(n, A, Y) \\ \emptyset & \text{otherwise} \end{cases}$$

Remark for any history h, the value of $\mathfrak{S}(h)$ only depends on $last(h)$, and we can thus consider $\mathfrak{S}$ as a strategy associating to any state s a set of joint actions. Let Σ be any A-strategy, suppose that $B = \mathrm{Out}^{\dagger}(y, \mathfrak{S}) \cap \mathrm{Out}(y, \Sigma)$ is non-empty. The fact that there is a $\pi \in B$ such that π satisfies $\varphi \,\mathsf{R}\, \psi$ follows by observing that $y \in Y$, and that for a given state $v \in Y$, $v \in \{\{\psi\}\}$ and either $v \in \{\{\varphi\}\}$ or $v \notin \{\{\varphi\}\}$. In this last case, since $v \in Y$ the strategy $\mathfrak{S}$ will select all the joint actions a that leads to a state that is not in Y. Thus given any history h such that $last(h) = v$ we will have that there is a v' in $next(\Sigma(h), v)$ such that $v' \in Y$ (under the hypothesis that B is non-empty) and we can thus construct a path having the wanted property. Since Σ was arbitrary, the previous reasoning holds for any A-strategy and we obtain the wanted result. □

4.1 Model Checking

Here, we show that the model-checking problem for OATL is PTIME-complete. To obtain this result, we first show how each ATL formula φ can be translated to an OATL formula $(\varphi)^{\bullet}$ for which the following holds: given any model $\mathfrak{M} = \langle \mathfrak{C}, \$ \rangle$, we have that $\mathfrak{C}, s \models^{ATL} \varphi$ iff $\mathfrak{M}, s \models (\varphi)^{\bullet}$. We thus obtain a reduction for the model checking problem of ATL to the one of OATL. Since the former is PTIME-hard this gives us a lower-bound. To show that the model-checking problem for OATL is PTIME-easy, we provide an Algorithm (Algorithm 1) that given a model $\mathfrak{M}$ and a formula φ returns the set of states of $\mathfrak{M}$ satisfying φ. This labeling algorithm is nothing but an extension of the one for ATL.

We start with the reduction: we define the 0-fragment of OATL to be the set of OATL formulae in which the grade of every strategic formula is 0. Let $(-)^{\bullet}$ be the function from ATL formulae to OATL formulae that is the identity on atomic propositions and $\top$, commutes with the boolean connective and such that:

$$(\llbracket A \rrbracket \mathsf{X}\, \varphi)^{\bullet} = \llbracket A \rrbracket^{\dagger}_0 \mathsf{X}\, (\varphi)^{\bullet}$$
$$(\llbracket A \rrbracket (\varphi \,\mathsf{U}\, \psi))^{\bullet} = \llbracket A \rrbracket^{\dagger}_0 ((\varphi)^{\bullet} \,\mathsf{U}\, (\psi)^{\bullet})$$
$$(\llbracket A \rrbracket (\varphi \,\mathsf{R}\, \psi))^{\bullet} = \llbracket A \rrbracket^{\dagger}_0 ((\varphi)^{\bullet} \,\mathsf{R}\, (\psi)^{\bullet})$$

we can easily show the following by remarking that $\mathrm{Out}^{\dagger}(\mathfrak{S}, s)$ contains all paths starting at s when $\mathfrak{S}$ is a 0-strategy.

Lemma 1. *For every ATL formula φ, every CGS $\mathfrak{C}$, every cost function $\$$, and every state s of $\mathfrak{C}$ we have that: $\mathfrak{C}, s \models^{ATL} \varphi$ iff $\langle \mathfrak{C}, \$ \rangle, s \models (\varphi)^{\bullet}$*

We can now state the model checking problem.

Definition 6. *Given a finite model $\mathfrak{M}$, a state s of $\mathfrak{M}$, and an OATL formula φ, the model checking problem consists in determining whether $\mathfrak{M}, s \models \varphi$.*

Given a model $\mathfrak{M}$, its size $|\mathfrak{M}|$ is the cardinality of the transition function T of $\mathfrak{M}$. Given a formula φ, we denote by $||\varphi||$ the cardinality of the set of subformulae of φ.

Theorem 3. *The model-checking problem for OATL is PTIME-complete and can be solved in time* $\mathrm{O}(|\mathfrak{M}| \times ||\varphi||)$.

Proof. The lowerbound follows from Lemma 1. For the upperbound, Algorithm 1 shows a procedure for model checking OATL, which manipulates set of states. The procedure is inspired by the model checking for CTL [16] and ATL [3]. However, it uses the additional procedures: the function *Sub* returns an ordered sequence, w.r.t. their complexities, of sub-formulae of a given formula φ.

The function $\dagger(n, A, X)$ which takes a natural number n, coalition A, and a subset of states X. The function returns the subset X' of $Pre(A, X)$, such that $Cost(\mathcal{D}(s', A, \overline{X})) \leq n$ for all $s' \in X'$. The

Algorithm 1 Labeling Algorithm ($\mathfrak{M}, \varphi$)

```
for all $\varphi \in Sub(\varphi)$ do
    switch $\varphi$ do
        case $\varphi = \top$
            $\{\{\varphi\}\} \leftarrow S$
        case $\varphi = p$
            $\{\{\varphi\}\} \leftarrow \{s \in S : p \in \mathcal{V}(s)\}$
        case $\varphi = \neg\varphi_1$
            $\{\{\varphi\}\} \leftarrow S \setminus \{\{\varphi_1\}\}$
        case $\varphi = \varphi_1 \wedge \varphi_2$
            $\{\{\varphi\}\} \leftarrow \{\{\varphi_1\}\} \cap \{\{\varphi_2\}\}$
        case $\varphi = [\![A]\!]^{\dagger}_{n} \mathsf{X}\, \varphi_1$
            $\{\{\varphi\}\} \leftarrow \dagger(n, A, \{\{\varphi_1\}\})$
        case $\varphi = [\![A]\!]^{\dagger}_{n} (\varphi_1 \,\mathsf{U}\, \varphi_2)$
            $X \leftarrow \emptyset$; $Y \leftarrow \{\{\varphi_2\}\}$
            while $Y \neq X$ do
                $X \leftarrow Y$
                $Y \leftarrow \{\{\varphi_2\}\} \cup (\{\{\varphi_1\}\} \cap \dagger(n, A, X))$
            $\{\{\varphi\}\} \leftarrow Y$
        case $\varphi = [\![A]\!]^{\dagger}_{n} (\varphi_1 \,\mathsf{R}\, \varphi_2)$
            $X \leftarrow \{\{\top\}\}$; $Y \leftarrow \{\{\varphi_2\}\}$
            while $X \neq Y$ do
                $X \leftarrow Y$
                $Y \leftarrow \{\{\varphi_2\}\} \cap (\{\{\varphi_1\}\} \cup \dagger(n, A, X))$
            $\{\{\varphi\}\} \leftarrow Y$
```

Algorithm 2 Test (s, A, X, n))

```
1: $Sum \leftarrow 0$
2: for $f \in \mathcal{F}(A, s)$ do
3:     if $next(f, s) \subseteq \overline{X}$ then
4:         $Sum \leftarrow Cost(f_s^{\preceq})$
5: if $Sum \leq n$ then
6:     Return True
7: else
8:     Return False
```

worst possible case is when $Pre(A, X) = S$, and one needs to call $|S|$-times the function $Cost(\mathcal{D}(s', A, X))$. So, we are quadratic in S, since $(|S| \cdot |S|) \leq |T|$ we are also polynomial in $|T|$.

Algorithm 2 calculates for a given state s, if $Cost(\mathcal{D}(s, n, X)) \leq n$. Such an Algorithm runs in polynomial-time in the cardinality of the transition function T. In fact, it has just one **for** loop who ranges over the actions that are available to the coalition A in the given state s and uses as subroutine the next function from ATL (line 3). Since the number of these actions never exceeds the cardinality of T, and since $next(f, s)$ can be computed linearly in the cardinality of T, then Algorithm 2 runs in polynomial-time in the cardinality of T.

Algorithm 1 works bottom-up on the structure of the formula; the cases of interest are for strategic formulas. Termination of such procedure is guaranteed, as the state space S is finite. Soundness and completeness of the algorithm directly follows from Proposition 1 and Theorem 2. □

5 IMPERFECT INFORMATION

Here, we are delving into a semantic variation of our logic, wherein the Demon, the agents, or even both may possess imperfect information regarding the potential developments within a game. As is customary, the agents' lack of complete information about the game model will be represented by dividing the states of the model into distinct equivalence classes (one for each agent, plus one for the Demon). States belonging to the same equivalence class for agent i will be regarded as indistinguishable from agent i's perspective. In the same way, states belonging to the same equivalence class for the Demon, will be regarded as indistinguishable from his perspective.

Definition 7. *Given a set of atoms* Ap *and a set of agents* Ag*, an Imperfect-information CGS (iCGS for short) over* Ap *and* Ag *is a tuple* $\mathfrak{C} = \langle S, s_i, P, T, \mathcal{V}, \{\sim_i\}_{i \in \mathrm{Ag}} \rangle$ *where:*

- $\langle S, s_i, P, T, \mathcal{V} \rangle$ *is a CGS over* Ap *and* Ag*;*
- *for each* $i \in \mathrm{Ag}$, $\sim_i \subseteq S \times S$ *is an equivalence relation over* S*.*

An imperfect information model (iModel for short) is a triple $\langle \mathfrak{C}, \sim_{\dagger}, \$ \rangle$ *where* $\sim_{\dagger}$ *is the Demon's equivalence relation over* S *and* \$ *is a cost function.*

Note that CGS can be seen as a specific instance of iCGS, where $\sim_i$ is simply the identity function on S for each $i \in \mathrm{Ag}$. Likewise, a model can be considered a special case of an iModel, where the iCGS corresponds to a CGS, and $\sim_{\dagger}$ is also the identity function on S. Since we are dealing with imperfect information, it becomes essential to introduce the concept of uniform strategies. To begin, when considering two histories, h and h', and specific agent $i \in \mathrm{Ag}$, we will say that $h \equiv_i h'$ if and only h and h' have the same length n, and $h_j \sim_i h'_j$ for every $j \leq n$. Similarly, we will use the notation $h \equiv_{\dagger} h'$ if and only if h and h' have the same length n, and $h_j \sim_{\dagger} h'_j$ for every $j \leq n$. With these definitions in place, we can now proceed to define uniform strategies for both the agents and the Demon.

Definition 8 (Uniform Strategies). *Given an iModel* $\mathfrak{M}$ *and a coalition, a* ***uniform A-strategy*** *is an* A*-strategy* Σ *such that for every* $i \in A$*, for every pair of histories* h *and* h'*, if* $h \equiv_i h'$ *then* $(\Sigma(h))(i) = (\Sigma(h'))(i)$*. A* ***uniform demonic n-strategy*** *is a demonic n-strategy* $\mathfrak{S}$ *such that, for every pair of histories* h *and* h'*, if* $h \equiv_{\dagger} h'$ *then* $\mathfrak{S}(h) = \mathfrak{S}(h')$*.*

A strategy $\mathcal{S}$ (Demonic or not) is said to be memoryless whenever $last(h) = last(h')$ implies $\mathcal{S}(h) = \mathcal{S}(h')$ for each pair of histories h and h'. Let us now delineate some variations of the satisfaction relation introduced in Definition 5.

Definition 9. *Let* $\mathfrak{M}$ *be an iModel,* s *be any state of* $\mathfrak{M}$*, and* φ *be any formula, we write:*

- $\mathfrak{M}, s \models^{iR} \varphi$ *for the satisfaction relation obtained by replacing, in Definition 5, every occurrence of "demonic n-strategy" with "uniform demonic n-strategy";*
- $\mathfrak{M}, s \models^{ir} \varphi$ *for the satisfaction relation obtained by replacing, in Definition 5, every occurrence of "demonic n-strategy" with "uniform memoryless demonic n-strategy";*
- $\mathfrak{M}, s \models_{iR} \varphi$ *for the satisfaction relation obtained by replacing, in Definition 5, every occurrence of "A-strategy" with "uniform A-strategy";*
- $\mathfrak{M}, s \models^{ir}_{ir}$ *for the satisfaction relation obtained by replacing, in Definition 5, every occurrence of "demonic n-strategy" with "uniform memoryless demonic n-strategy" and every occurrence of "A-strategy" with "uniform memoryless A-strategy".*

First, we provide a result for the worste case.

Proposition 2. *The model-checking problem for OATL under the satisfaction relation* $\models_{iR}$ *is* undecidable.

Proof. Consider a formula belonging to the 0-fragment of OATL, and let $\mathfrak{M} = \langle C, \sim_{\dagger}, \$ \rangle$ be any iModel. Since the set of paths starting at s compatible with a 0-strategy is the set of paths starting at s, it is easy to see that $\mathfrak{M}, s \models_{iR} \varphi$ iff $\mathfrak{C}, s \models^{ATL}_{iR} (\varphi)^{\bullet}$ where $(-)^{\bullet}$ is the translation from ATL to the 0-fragment of OATL given in Subsection 4.1. Since the model checking problem is undecidable for ATL under the $\models^{ATL}_{iR}$ satisfaction relation [18], we can conclude. □

We recall that the bottom-up approach is a methodology that is peculiar to strategic (and temporal) logics. This methodology allows reducing the satisfiability of a formula with multiple strategic operators, to the satisfiability of a formula containing just one strategic operator. The procedure can be described as follows: given a model $\mathfrak{M}$ and formula φ with multiple strategic operators, let $\varphi_1, \dots, \varphi_n$ be the strategic subformulae of φ containing exactly one strategic operator. For any ϕ_i we choose a fresh atom p_i. We add p_i to $\mathcal{V}(s)$ whenever s satisfies ϕ_i in $\mathfrak{M}$, obtaining a new model $\mathfrak{M}'$. We then consider the formula ϕ' obtained from ϕ by substituting each occurrence of ϕ_i with p_i and evaluate ϕ' on $\mathfrak{M}'$ reusing the same procedure, and we then iterate this process.

In a cybersecurity scenario, as described in the Section 3, the case where the Demon (the defender) has imperfect information and no memory, and the players possess perfect memory and information represents the worst-case scenario to consider when determining the existence of a defense strategy on a given system. Hopefully, the complexity of the latter scenario is not prohibitive. The proof of this fact is a copycat of the one given by Schobbens for ATL in his classic paper [42], and we thus omit it.

Theorem 4. *The model checking problem for OATL under the satisfaction relation $\models^{ir}$ is in P^{NP}.*

From the above theorem, and from the classic result of Schobbens mentioned above, one obtains the following.

Corollary 1. *The model checking problem for OATL under the satisfaction relation $\models^{ir}_{ir}$ is in P^{NP}.*

In what follows, we prove that deciding if $\mathfrak{M}, s \models^{iR} \varphi$ is EXPTIME-complete. Let us introduce some notation that will be used in the following paragraphs. If $\mathfrak{M}$ is a iModel and s is one of its states, we denote by $\mathcal{P}(\mathfrak{M}, s)$ the set of paths of $\mathfrak{M}$ whose first state is s. If $\mathfrak{M}$ is a model, we use $S^{\mathfrak{M}}$ to denote the set of states of $\mathfrak{M}$, $T^{\mathfrak{M}}$ will denote its transition function and so on.

If $\mathfrak{M}$ and $\mathfrak{M}'$ are two iModels, then $\mathfrak{M}$ is a submodel of $\mathfrak{M}'$ if $S^{\mathfrak{M}} \subseteq S^{\mathfrak{M}'}$, $Act^{\mathfrak{M}} \subseteq Act^{\mathfrak{M}'}$. For any state s and s' of $\mathfrak{M}$ and joint action $\mathbf{a}$ of $\mathfrak{M}$ we have that $T^{\mathfrak{M}}(s, \mathbf{a}) = s'$ iff $T^{\mathfrak{M}'}(s, \mathbf{a}) = s'$. Furthermore, $\mathcal{L}^{\mathfrak{M}}(s) = \mathcal{L}^{\mathfrak{M}'}(s)$ for any state s of $\mathfrak{M}$. Finally, $\{\sim_i\}^{\mathfrak{M}}_{i \in \mathrm{Ag} \cup Demon}$ is the restriction of $\{\sim_i\}^{\mathfrak{M}'}_{i \in \mathrm{Ag} \cup Demon}$ over $S^{\mathfrak{M}}$.

If $\mathfrak{C} = \langle S, s_I, P, T, \mathcal{V}, \{\sim_i\}_{i \in \mathrm{Ag}} \rangle$ is an iCGS, s is a state of $\mathfrak{C}$ and $X \subseteq \mathcal{P}(\mathfrak{C}, s)$, then the unwinding of $\mathfrak{C}$ over X is the iCGS $\mathfrak{C}' = \langle S', s'_I, P', T', \mathcal{V}', \{\sim'_i\}_{i \in \mathrm{Ag}} \rangle$, where:

- the set of states S' is the set of non-empty finite prefixes of paths from X, the initial state s'_I is the sequence whose the only element is s itself;
- for every $h \in S'$, we have $P'(h) = P(last(h))$;
- for every pair of histories h and h' in S', for every joint action $\mathbf{a}$, we have that $h' = T'(h, \mathbf{a})$ iff $h' = h \cdot s$ and $s = T(lasth(h), \mathbf{a})$;
- for every $h \in S'$, $\mathcal{V}'(h) = \mathcal{V}(last(h))$;
- for every $i \in \mathrm{Ag}$, $\sim'_i = \{\langle h, h' \rangle \in S' \times S' \mid h \equiv_i h'\}$.

The unwinding $\mathfrak{C}^{\mathcal{U}}$ of an iCGS $\mathfrak{C}$ is simply the unwinding over the set of paths starting at the initial state of the of $\mathfrak{C}$. Given an iModel $\mathfrak{M}$, the unwinding $\mathfrak{M}^{\mathcal{U}}$ of $\mathfrak{M}$ is the iModel $\langle \mathfrak{C}^{\mathcal{U}}, \sim^{\mathcal{U}}_{\dagger}, \$ \rangle$ where $\mathfrak{C}^{\mathcal{U}}$ is the unwinding of $\mathfrak{C}$, $\sim^{\mathcal{U}}_{\dagger}$ is $\{\langle h, h' \rangle \in S' \times S' \mid h \equiv_{\dagger} h'\}$ and $\$^{\mathcal{U}}(h, \mathbf{a}) = n$ iff $\$(last(h), \mathbf{a}) = n$, for any $n \in \mathbb{N}^{+}$.

Given a path λ of a iCGS:

- its magnitude $M(\lambda)$ is defined as the cost of the joint action who defines the transition from λ_1 to λ_2, i.e., $\mathcal{M}(\lambda) = \$(\lambda_1, \mathbf{a})$ for $\mathbf{a} \in \mathcal{J}(\lambda_1)$ such that $\lambda_2 = T(\lambda_1, \mathbf{a})$;
- its label $\mathcal{L}(\lambda)$ is the joint action $\mathbf{a}$ that defines the transition from λ_1 to λ_2 i.e., $\mathcal{L}(\lambda) = \mathbf{a}$ such that $\lambda_2 = T(\lambda_1, \mathbf{a})$;
- if X is a set of paths, then we denote by $\mathcal{L}(X) = \{\mathbf{a} \in \mathcal{J}(s) \mid \mathbf{a} = \mathcal{L}(\lambda) \text{ for some } \lambda \in X\}$.

Definition 10. *Let $\mathfrak{M}$ be an iModel and $\mathfrak{M}^{\mathcal{U}}$ be its unwinding. A demonic n-strategy tree $\mathfrak{T}$ is any submodel of $\mathfrak{M}^{\mathcal{U}}$ such that:*

1. *$\mathcal{P}(\mathfrak{T}, h) \neq \emptyset$ for any h that belongs to the set of states of $\mathfrak{T}$;*
2. *for any state h of $\mathfrak{T}$, for any subset X of $\mathcal{P}(\mathfrak{M}^{\mathcal{U}}, h)$, if $X \nsubseteq \mathcal{P}(\mathfrak{T}, h)$ then $(\sum_{\lambda \in X} \mathcal{M}(\lambda)) \leq n$;*
3. *for any pair of states h and h' of $\mathfrak{T}$, if $\mathcal{P}(\mathfrak{M}^{\mathcal{U}}, h) \cap \overline{\mathcal{P}(\mathfrak{T}, h)} = X$, $\mathcal{P}(\mathfrak{M}^{\mathcal{U}}, h') \cap \overline{\mathcal{P}(\mathfrak{T}, h')} = Y$, and $h \sim_{\dagger} h'$ then $\mathcal{L}(X) = \mathcal{L}(Y)$.*

Now, we have all the ingredients to prove the following result.

Lemma 2. *Let $\mathfrak{M} = \langle \mathfrak{C}, \sim_{\dagger}, \$ \rangle$ be an iModel and $\varphi = [\![A]\!]^{\dagger}_{n}\psi$ be a formula such that ψ does not contain any occurrence of a strategic operator. We can prove that $\mathfrak{M} \models^{iR} [\![A]\!]^{\dagger}_{n}\psi$ iff there is a demonic n-tree $\mathfrak{T}$ over the unwinding $\mathfrak{M}^{\mathcal{U}}$ of $\mathfrak{M}$ such that $\mathfrak{T} \models^{ATL} [\![A]\!]\psi$.*

Proof. For the $(\rightarrow)$-direction, suppose that $\mathfrak{M} \models [\![A]\!]^{\dagger}_{n}\psi$. Thus there is a demonic uniform n-strategy $\mathfrak{S}$ such that for every A-strategy Σ, if $\mathrm{Out}^{\dagger}(s_I, \mathfrak{S}) \cap \mathrm{Out}(s_I, \Sigma) \neq \emptyset$ then there is a $\pi \in \mathrm{Out}^{\dagger}(s_I, \mathfrak{S}) \cap \mathrm{Out}(s_I, \Sigma)$ such that $\pi \models \psi$. Consider the unwinding $\mathfrak{C}'$ of $\mathfrak{C}$ over $\mathrm{Out}^{\dagger}(s_I, \mathfrak{S})$. It is easy to see that $\mathfrak{C}'$ is a demonic n-strategy tree and that $\langle \mathfrak{C}' \sim'_{\dagger}, \$' \rangle \models [\![A]\!]^{\dagger}_{0}\psi$, where $\sim'_{\dagger}$ and $\$'$ are, respectively, the restriction of $\sim_{\dagger}$ and \$ to $\mathfrak{C}'$. Since a 0-uniform strategy is a 0-strategy, and since $\mathfrak{C}'$ is a CGS, by Lemma 1 we obtain that $\mathfrak{C} \models^{ATL} [\![A]\!](\psi)^{\bullet}$, from the fact that ψ does not contain any strategic operator, we conclude that $(\psi)^{\bullet} = \psi$.

For the converse direction, suppose that $\mathfrak{T} \models^{ATL} [\![A]\!]\psi$ for some demonic n-strategy tree $\mathfrak{T}$ over $\mathfrak{M}^{\mathcal{U}}$. Given any state h of $\mathfrak{T}$, let $h^{\mathfrak{T}}_{\mathcal{J}}$ be $\{\mathbf{a} \in \mathcal{J}(last(h)) \mid \exists h' \in S^{\mathfrak{M}^{\mathcal{U}}} \wedge h' \notin S^{\mathfrak{T}} \wedge T^{\mathfrak{M}^{\mathcal{U}}}(h, \mathbf{a}) = h'\}$. We define a n-demonic uniform strategy $\mathfrak{S}$ by $\mathfrak{S}(h) = h^{\mathfrak{T}}_{\mathcal{J}}$ if $h \in S^{\mathfrak{T}}$ and $\mathfrak{S}(h) = \emptyset$, otherwise. Thus, the result follows by observing that $h \in \mathbf{S}^{\mathfrak{T}}$ iff $h \sqsubset \pi$ for some $\pi \in \mathrm{Out}^{\dagger}(s_I, \mathfrak{S})$. □

Given the above lemma, we can characterize the complexity of deciding whether $\mathfrak{M}, s \models^{iR} \varphi$.

Theorem 5. *The model checking problem for OATL under the $\models^{iR}$ satisfiability relation is EXPTIME-complete.*

Sketch. For the upper-bound, consider a formula $\varphi = [\![A]\!]^{\dagger}_{n}\psi$ where ψ does not contain any strategic operator. By Lemma 2, $\mathfrak{M} \models^{iR} \varphi$ iff there is a n-demonic strategy tree $\mathfrak{T}$, such that $\mathfrak{T} \models^{ATL} \varphi'$, where $\varphi' = [\![A]\!]\psi$. Given a strategy tree $\mathfrak{T}$ to check whether it satisfies a formula such as φ', one can check whether such a tree satisfies ψ within the $(|S^{\mathfrak{M}}| + 1)$-th depth of $\mathfrak{T}$ (see [31–33] to further details on this aspect). One can enumerate all the finite trees of depth $|S^{\mathfrak{M}}|+1$ that are subtrees of n-demonic strategy trees, and check whether one of them satisfies φ' using the ATL model checking algorithm. In the worst case, the number of such subtrees is exponential in the cardinality $T^{\mathfrak{M}}$. Since we call for each of them a polynomial-time algorithm (i.e., ATL model checking), the result follows. To generalize such a result to arbitrary formulae, recall that we can use the classic bottom-up approach for ATL formulas. For the lower-bound, we recall that to solve two-player turn-based games with imperfect information is EXPTIME-hard [41]. □

6 RELATED WORK

In the past years, many works focused on the strategic abilities of agents playing in a dynamic game model. We compare our approach with some of these works, highlighting differences.

Obstruction Logic (OL) [13] is a recently introduced logic allowing reasoning about two-player games played on a labeled and weighted directed graph. In OL, one of the two players, known as the Demon, has the power to temporarily disable edges in the graph whose sum of weights does not exceed a given natural number. For simplicity, let's consider an OL formula $\langle\langle \dagger_n \rangle\rangle\psi$ where ψ does not contain strategic operators. The reader can easily check that $\mathfrak{M}, s \models_{OL} \langle\langle \dagger_n \rangle\rangle\psi$ iff $\mathfrak{M}, s \models [\![\mathrm{Ag}]\!]^{\dagger}_{n}\psi$. This relationship between OL and OATL semantics can be extended in the obvious way for boolean and temporal operators. Thus, OATL is an extension of OL.

Sabotage Modal Logic and its extensions [4, 30, 44] is another line of research related to our work. Sabotage games have been introduced by van Benthem with the aim of studying the computational complexity of a special class of graph-reachability problems. In these games, one player moves over a directed graph by traversing adjacent nodes and tries to reach a certain subset of nodes, while the other player has the power of *erasing* an edge *anywhere* in the graph at any turn. To reason about sabotage games, van Benthem introduced Sabotage Modal Logic (SML). SML is obtained by adding to the $\Diamond$-modality of classical modal logic another modality $\blacklozenge$. Let G be a directed graph and s one of its vertex; the intended meaning of a formula $\blacklozenge\varphi$ is " $\blacklozenge\varphi$ is true at a state s of G iff φ is true at s in the graph obtained by G by *erasing* an edge e". The model checking problem for Sabotage Modal Logic is PSPACE-complete [30]. Our games are incomparable with those considered in SML: they are concurrent and multiplayer games, we consider temporal objectives and the Demon can select subset of edges based on their weight.

The authors of [43] introduce **Dynamic Escape Games** (DEG, for short). Such games have a close resemblance to ours. These are games with weighted transitions in which Player1 (P1) tries to reach a target state while Player2 (P2) tries to prevent it. Along a play, P1 plays as usual, i.e., from his current position, he chooses one of the available successors and moves to it. Conversely, P2, sitting on a set of states S, chooses some of her successors that become *irrevocably* unavailable to P1 and adds them to S. Contrary to us, the authors have proposed an optimized heuristic that provides partial results to check whether P1 has a strategy to reach one of his goal states.

Logics on **Normative Systems** (NS) [1, 2] are also related to OATL. An NS is a transition system in which some transitions are considered illegal and deactivated according to some parameter. Formally, in logic for NS one evaluates CTL or ATL formulae with respect to a transition system in which a set of arcs has been deleted according to a given function. The assignment function on NS in non-local e non-quantitative: any subset of arcs can be deleted by the assignment and there is no notion of deletion cost.

Module checking (MC) for strategic and temporal logics [8, 9, 28] is a line of work which is related to ours: more precisely to the fragment of OATL in which the grade of strategic operators is 1 and the cost of every transition is one. In MC, states of a model are partitioned into those controlled by the environment and those controlled by the system. Given a formula φ and a model $\mathfrak{M}$, the MC problem is solved by determining whether **every** tree obtained by deleting one out-coming edge of an environment state in the unwinding of $\mathfrak{M}$ satisfies φ, and this for every state of the environment. Although there is a similarity between MC and OATL model checking, the two approaches are orthogonal. In OATL each state of the model can be seen as a state that is controlled by the environment (the Demon). Furthermore, we can only ask whether there **exists** a subtree of the unwinding of the model that satisfies a universal ATL formula. This difference is found in the fact that the model checking problem for OATL is polynomial, while the MC problem (even for "simple" logics like CTL) is at least exponential.

From the **cybersecurity** side, several existing works have proposed different game-theoretic solution for finding an optimal defense policy. Most of these approaches try to solve games using analytic and optimization techniques, e.g., [19, 20, 37, 47]. The work in [10] shares some ideas with our approach on the cybersecurity side. However, the authors do not use dynamic models and study a timed-logic framework and timed games to express and evaluate network security properties, which result in an EXPTIME-complete procedure. The already mentioned work in [11] propose a logical approach to play on Attack Graphs. They show that security games can be coded by means of their logic, and that the model-checking problem for such a logic is decidable. The work in [14] shows a technique to check whether an attacker has a strategy to achieve a reachability objective by using an automata-theoretic approach.

7 CONCLUSION

We introduced OATL, a logic allowing to reason about concurrent games in which one of the players has the power to modify, locally and temporarily, the game structure. We showed how to express cybersecurity properties via OATL. We studied the formal properties of OATL and its model-checking problem under different semantics.

As future work, we aim to deeply investigate the context of imperfect information. Unfortunately, as demonstrated, this problem is generally undecidable. To address this challenge, we could consider employing an approximation to perfect information [5], memory [7], or some hybrid technique [21, 22]. Furthermore, we aim to investigate semantic variants of imperfect information OATL where the Demon, the players, or both, employ bounded memory strategies [6] or natural strategies [24, 25]. Finally, we plan to give a sound and complete axiomatization for OATL.

ACKNOWLEDGMENTS

The first author is supported by the PRIN project RIPER (No. 20203FFYLK).

REFERENCES

[1] Thomas Ågotnes, Wiebe van der Hoek, Juan A. Rodríguez-Aguilar, Carles Sierra, and Michael J. Wooldridge. 2007. On the Logic of Normative Systems. In *IJCAI 2007, Proceedings of the 20th International Joint Conference on Artificial Intelligence 2007*, Manuela M. Veloso (Ed.). 1175–1180. http://ijcai.org/Proceedings/07/Papers/190.pdf

[2] Natasha Alechina, Brian S. Logan, and Mehdi M. Dastani. 2018. Modeling Norm Specification and Verification in Multiagent Systems. *FLAP* 5 (2018), 457–490.

[3] Rajeev Alur, Thomas A. Henzinger, and Orna Kupferman. 2002. Alternating-time temporal logic. *J. ACM* 49, 5 (2002), 672–713.

[4] Guillaume Aucher, Johan van Benthem, and Davide Grossi. 2018. Modal logics of sabotage revisited. *Journal of Logic and Computation* 28, 2 (March 2018), 269 – 303. https://doi.org/10.1093/logcom/exx034

[5] Francesco Belardinelli, Angelo Ferrando, and Vadim Malvone. 2023. An abstraction-refinement framework for verifying strategic properties in multi-agent systems with imperfect information. *Artif. Intell.* 316 (2023), 103847. https://doi.org/10.1016/j.artint.2022.103847

[6] Francesco Belardinelli, Alessio Lomuscio, and Vadim Malvone. 2018. Approximating Perfect Recall When Model Checking Strategic Abilities. In *KR*. 435–444. https://aaai.org/ocs/index.php/KR/KR18/paper/view/18010

[7] Francesco Belardinelli, Alessio Lomuscio, Vadim Malvone, and Emily Yu. 2022. Approximating Perfect Recall when Model Checking Strategic Abilities: Theory and Applications. *J. Artif. Intell. Res.* 73 (2022), 897–932. https://doi.org/10.1613/jair.1.12539

[8] Laura Bozzelli and Aniello Murano. 2017. On the Complexity of ATL and ATL* Module Checking. In *Proceedings Eighth International Symposium on Games, Automata, Logics and Formal Verification, GandALF 2017 (EPTCS, Vol. 256)*, Patricia Bouyer, Andrea Orlandini, and Pierluigi San Pietro (Eds.). 268–282. https://doi.org/10.4204/EPTCS.256.19

[9] Laura Bozzelli, Aniello Murano, and Adriano Peron. 2020. Module Checking of Pushdown Multi-agent Systems. In *Proceedings of the 17th International Conference on Principles of Knowledge Representation and Reasoning, KR 2020*, Diego Calvanese, Esra Erdem, and Michael Thielscher (Eds.). 162–171. https://doi.org/10.24963/kr.2020/17

[10] Elie Bursztein and Jean Goubault-Larrecq. 2007. A Logical Framework for Evaluating Network Resilience Against Faults and Attacks. In *Advances in Computer Science - ASIAN 2007. Computer and Network Security, 12th Asian Computing Science Conference (LNCS, Vol. 4846)*, Iliano Cervesato (Ed.). Springer, 212–227. https://doi.org/10.1007/978-3-540-76929-3_20

[11] Davide Catta, Jean Leneutre, and Vadim Malvone. 2022. Subset sabotage games & attack graphs. In *Proceedings of the 23rd Workshop "From Objects to Agents"*, Vol. 3261. CEUR-WS.org, 209–218. http://ceur-ws.org/Vol-3261/paper16.pdf

[12] Davide Catta, Jean Leneutre, and Vadim Malvone. 2023. Attack Graphs & Subset Sabotage Games. *Intelligenza Artificiale* 17, 1 (2023), 77–88. https://doi.org/10.3233/IA-221080

[13] Davide Catta, Jean Leneutre, and Vadim Malvone. 2023. Obstruction Logic: A Strategic Temporal Logic to Reason About Dynamic Game Models. In *ECAI 2023 - 26th European Conference on Artificial Intelligence, 30 September- 4 October 2023, Krakow, Poland (Frontiers in Artificial Intelligence and Applications, Vol. 372)*, Grzegorz J. Nalepa Roy Fairstein Roxana Rădulescu Kobi Gal, Ann Nowé (Ed.). IOS Press, 365–372.

[14] Davide Catta, Antonio Di Stasio, Jean Leneutre, Vadim Malvone, and Aniello Murano. 2023. A Game Theoretic Approach to Attack Graphs. In *Proceedings of the 15th International Conference on Agents and Artificial Intelligence, ICAART 2023, Volume 1, Lisbon, Portugal, February 22-24, 2023*, Ana Paula Rocha, Luc Steels, and H. Jaap van den Herik (Eds.). SCITEPRESS, 347–354. https://doi.org/10.5220/0011776900003393

[15] Edmund M Clarke. 1997. Model checking. In *International Conference on Foundations of Software Technology and Theoretical Computer Science*. Springer, 54–56.

[16] Edmund M. Clarke and E. Allen Emerson. 1981. Design and Synthesis of Synchronization Skeletons Using Branching-Time Temporal Logic.. In *81 (LNCS 131)*. Springer, 52–71.

[17] Robert S. Dewar. 2017-06. *Active Cyber Defense.* Report. Zurich.

[18] Catalin Dima and Ferucio Laurentiu Tiplea. 2011. Model-checking ATL under Imperfect Information and Perfect Recall Semantics is Undecidable. *CoRR* abs/1102.4225 (2011). arXiv:1102.4225 http://arxiv.org/abs/1102.4225

[19] Karel Durkota, Viliam Lisý, Branislav Bosanský, and Christopher Kiekintveld. 2015. Approximate Solutions for Attack Graph Games with Imperfect Information. In *Decision and Game Theory for Security - 6th International Conference, GameSec 2015 (LNCS, Vol. 9406)*, M. H. R. Khouzani, Emmanouil A. Panaousis, and George Theodorakopoulos (Eds.). Springer, 228–249. https://doi.org/10.1007/978-3-319-25594-1_13

[20] Karel Durkota, Viliam Lisý, Branislav Bosanský, and Christopher Kiekintveld. 2015. Optimal Network Security Hardening Using Attack Graph Games. In *Proceedings of the Twenty-Fourth International Joint Conference on Artificial Intelligence, IJCAI 2015*, Qiang Yang and Michael J. Wooldridge (Eds.). AAAI Press, 526–532. http://ijcai.org/Abstract/15/080

[21] Angelo Ferrando and Vadim Malvone. 2022. Towards the Combination of Model Checking and Runtime Verification on Multi-agent Systems. In *20th International Conference, PAAMS 2022 (LNCS, Vol. 13616)*, Frank Dignum, Philippe Mathieu, Juan Manuel Corchado, and Fernando de la Prieta (Eds.). Springer, 140–152. https://doi.org/10.1007/978-3-031-18192-4_12

[22] Angelo Ferrando and Vadim Malvone. 2023. Towards the Verification of Strategic Properties in Multi-Agent Systems with Imperfect Information. In *Proceedings of the 2023 International Conference on Autonomous Agents and Multiagent Systems, AAMAS 2023*, Noa Agmon, Bo An, Alessandro Ricci, and William Yeoh (Eds.). ACM, 793–801. https://doi.org/10.5555/3545946.3598713

[23] Yih Chun Hu and Adrian Perrig. 2006. Wormhole attacks in wireless networks. *IEEE Journal on Selected Areas in Communications* 24, 2 (Feb. 2006), 370–379. https://doi.org/10.1109/JSAC.2005.861394 Funding Information: Manuscript received October 11, 2004; revised August 15, 2005. This work was supported in part by the National Science Foundation (NSF) under Grant CCR-0209204, in part by NASA under Grant NAG3-2534, and in part by Schlumberger and Bosch. The views and conclusions contained here are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either express or implied, of NSF, NASA, Schlumberger, Bosch, The University of Illinois, Carnegie Mellon University, Rice University, or the U.S. Government or any of its agencies. This paper was presented in part at the 22nd Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM 2003), April 3, 2003, San Francisco, CA Y.-C. Hu is with the Department of Electrical and Computer Engineering, University of Illinois at Urbana–Champaign, Urbana, IL 61801 USA (e-mail: yihchun@crhc.uiuc.edu).

[24] Wojciech Jamroga, Vadim Malvone, and Aniello Murano. 2019. Natural strategic ability. *Artif. Intell.* 277 (2019). https://doi.org/10.1016/j.artint.2019.103170

[25] Wojciech Jamroga, Vadim Malvone, and Aniello Murano. 2019. Natural Strategic Ability under Imperfect Information. In *Proceedings of the 18th International Conference on Autonomous Agents and MultiAgent Systems, AAMAS '19, Montreal, QC, Canada, May 13-17, 2019*, Edith Elkind, Manuela Veloso, Noa Agmon, and Matthew E. Taylor (Eds.). International Foundation for Autonomous Agents and Multiagent Systems, 962–970. http://dl.acm.org/citation.cfm?id=3331791

[26] Nicholas R. Jennings and Michael J. Wooldridge. 1998. Application of Intelligent Agents. In *Agent Technology: Foundations, Applications, and Markets*. Springer-Verlag.

[27] Rafiullah Khan, Peter Maynard, Kieran McLaughlin, David Laverty, and Sakir Sezer. 2016. Threat Analysis of BlackEnergy Malware for Synchrophasor Based Real-Time Control and Monitoring in Smart Grid *(ICS-CSR '16)*. BCS Learning & Development Ltd., Swindon, GBR. https://doi.org/10.14236/ewic/ICS2016.7

[28] Orna Kupferman and Moshe Y. Vardi. 1996. Module Checking. In *Computer Aided Verification, 8th International Conference, CAV '96, New Brunswick, NJ, USA, July 31 - August 3, 1996, Proceedings (LNCS, Vol. 1102)*, Rajeev Alur and Thomas A. Henzinger (Eds.). Springer, 75–86. https://doi.org/10.1007/3-540-61474-5_59

[29] Antoine Lemay, Joan Calvet, Franois Menet, and Jos M. Fernandez. 2018. Survey of publicly available reports on advanced persistent threat actors. *Comput. Secur.* 72, C (jan 2018), 26–59. https://doi.org/10.1016/j.cose.2017.08.005

[30] Christof Löding and Philipp Rohde. 2003. Model Checking and Satisfiability for Sabotage Modal Logic. In *FST TCS 2003: Foundations of Software Technology and Theoretical Computer Science (LNCS, Vol. 2914)*, Paritosh K. Pandya and Jaikumar Radhakrishnan (Eds.). Springer, 302–313. https://doi.org/10.1007/978-3-540-24597-1_26

[31] Vadim Malvone and Aniello Murano. 2017. Reasoning About Additional Winning Strategies in Two-Player Games. In *Multi-Agent Systems and Agreement Technologies - 15th European Conference, EUMAS 2017, and 5th International Conference, AT 2017, Évry, France, December 14-15, 2017, Revised Selected Papers (Lecture Notes in Computer Science, Vol. 10767)*, Francesco Belardinelli and Estefania Argente (Eds.). Springer, 163–171. https://doi.org/10.1007/978-3-030-01713-2_12

[32] Vadim Malvone, Aniello Murano, and Loredana Sorrentino. 2017. Hiding Actions in Multi-Player Games. In *Proceedings of the 16th Conference on Autonomous Agents and MultiAgent Systems, AAMAS 2017, São Paulo, Brazil, May 8-12, 2017*, Kate Larson, Michael Winikoff, Sanmay Das, and Edmund H. Durfee (Eds.). ACM, 1205–1213. http://dl.acm.org/citation.cfm?id=3091293

[33] Vadim Malvone, Aniello Murano, and Loredana Sorrentino. 2018. Additional Winning Strategies in Reachability Games. *Fundam. Informaticae* 159, 1-2 (2018), 175–195. https://doi.org/10.3233/FI-2018-1662

[34] Zohar Manna and Amir Pnueli. 2012. *Temporal verification of reactive systems: safety.* Springer Science & Business Media.

[35] Fabio Mogavero, Aniello Murano, Giuseppe Perelli, and Moshe Y. Vardi. 2014. Reasoning About Strategies: On the Model-Checking Problem. *ACM Transactions in Computational Logic* 15, 4 (2014), 34:1–34:47. https://doi.org/10.1145/2631917

[36] Aniello Murano, Giuseppe Perelli, and Sasha Rubin. 2015. Multi-agent Path Planning in Known Dynamic Environments. In *PRIMA 2015: Principles and Practice of Multi-Agent Systems - 18th International Conference (LNCS, Vol. 9387)*, Qingliang Chen, Paolo Torroni, Serena Villata, Jane Yung-jen Hsu, and Andrea Omicini (Eds.). Springer, 218–231. https://doi.org/10.1007/978-3-319-25524-8_14

[37] Thanh H. Nguyen, Mason Wright, Michael P. Wellman, and Satinder Baveja. 2017. Multi-Stage Attack Graph Security Games: Heuristic Strategies, with Empirical Game-Theoretic Analysis *(MTD '17)*. Association for Computing Machinery, New York, NY, USA, 87–97.

[38] Xinming Ou and Anoop Singhal. 2011. *Quantitative Security Risk Assessment of Enterprise Networks.* Springer. https://doi.org/10.1007/978-1-4614-1860-3

[39] Amir Pnueli. 1977. The Temporal Logic of Programs. In *18th Annual Symposium on Foundations of Computer Science, Providence, Rhode Island, USA, 31 October - 1 November 1977.* IEEE Computer Society, 46–57. https://doi.org/10.1109/SFCS.1977.32

[40] Amir Pnueli and Roni Rosner. 1989. On the synthesis of a reactive module. In *Proceedings of the 16th ACM SIGPLAN-SIGACT symposium on Principles of programming languages.* 179–190.

[41] John H. Reif. 1984. The Complexity of Two-Player Games of Incomplete Information. *J. Comput. Syst. Sci.* 29, 2 (1984), 274–301. https://doi.org/10.1016/0022-0000(84)90034-5

[42] Pierre-Yves Schobbens. 2004. Alternating-Time Logic with Imperfect Recall. *ENTCS* 85, 2 (2004), 82–93.

[43] Antonio Di Stasio, Paolo Domenico Lambiase, Vadim Malvone, and Aniello Murano. 2018. Dynamic Escape Game. In *Proceedings of the 17th International Conference on Autonomous Agents and MultiAgent Systems, AAMAS 2018*, Elisabeth André, Sven Koenig, Mehdi Dastani, and Gita Sukthankar (Eds.). ACM, 1806–1808. http://dl.acm.org/citation.cfm?id=3237984

[44] Johan van Benthem. 2005. *An Essay on Sabotage and Obstruction.* Springer Berlin Heidelberg, Berlin, Heidelberg, 268–276. https://doi.org/10.1007/978-3-540-32254-2_16

[45] Moshe Y Vardi. 2005. An automata-theoretic approach to linear temporal logic. *Logics for concurrency: structure versus automata* (2005), 238–266.

[46] Kim Zetter. 2014. *Countdown to Zero Day: Stuxnet and the Launch of the World's First Digital Weapon.* Crown Publishing Group, USA.

[47] Yunxiao Zhang and Pasquale Malacaria. 2021. Bayesian Stackelberg games for cyber-security decision support. *Decis. Support Syst.* 148 (2021), 113599. https://doi.org/10.1016/j.dss.2021.113599