Obstruction Logic: a Strategic Temporal Logic to Reason about Dynamic Game Models

Davide Catta[a;*], **Jean Leneutre**[a] **and Vadim Malvone**[a]

[a]Télécom Paris, Palaiseau, France
ORCiD ID:

Abstract. Games that are played in a dynamic model have been studied in several contexts, such as cybersecurity and planning. In this paper, we introduce a logic for reasoning about a particular class of games with temporal goals played in a dynamic model. In such games, the actions of a player can modify the game model itself. We show that the model-checking problem for our logic is decidable in polynomial-time. Then, using this logic, we show how to express interesting properties of cybersecurity games defined on attack graphs.

1 Introduction

Establishing the soundness of both software and hardware systems is not a mundane task, especially when they are distributed. In the past fifty years, researchers have come up with different solution to this problem, and a story of success is the use of *formal methods* techniques [16]. These techniques make it possible to verify the correctness of a system by formally inspecting whether a mathematical model of it satisfies a formal depiction of its desired behavior. In particular, classic formal approaches such as *model checking* and automata-theoretic techniques, originally developed for monolithic systems [17, 27], have been meaningfully extended to handle *open* and *multi-agent systems* [28, 4, 33, 34, 23]. Multi-agent systems model the behavior of two or more rational agents interacting with each other (cooperatively or adversarially) and whose aim is achieving a certain goal [24]. Usually, such a behavior is modeled by means of a mixture of (temporal or modal) logic and game theory: agents are seen as players of games played over directed graphs (called arenas) and their objectives are specified by means of logical formulae. For instance, the syntax of logics such as ATL and Strategy Logic [3, 34], allows one to express the fact that a coalition of players has a strategy for reaching a certain goal by acting cooperatively. In all the above logics, the game model, where the players are playing, is considered as a fixed object: players' actions determine their location inside the arena but do not modify the structure of the arena itself. In contrast, games that are played in a dynamic (i.e., changing) game model have been studied in several contexts, such as cybersecurity and planning [35, 37, 11, 13].

In this paper, we introduce a logic for reasoning about a particular class of games with temporal goals played in a dynamic model. These games, are played over a directed graph by two players: the Demon and the Traveler. Each edge e of the graph has an associated deactivation-cost $C(e)$. Each round of the game is composed by a move of the Demon followed by a move of the Traveler: given a node v of the graph and a natural number n, the Demon deactivates a proper subset E of the set of edges that are incident to v such that the sum of the deactivation cost of the edges contained in E is less then n. Then the Traveler chooses a node v' such that v is adjacent to v' and $\langle v, v'\rangle$ does not belong to the set of edges that was chosen by the Demon in the previous round. A new round starts from the last node chosen by the Traveler, and the edges that were disabled in the previous round are restored. The Demon wins the game if the infinite sequence of nodes subsequently chosen by the Traveler satisfy a certain property φ expressed by a temporal formula. To reason about the existence of "demonic" strategies for this kind of games, we propose Obstruction Logic (or simply OL). We define its syntax and semantics, and then study the complexity of its model checking problem. In OL, one can quantify over the existence of demonic strategies that allows the Demon to temporally obstruct some reachable states.

Such a logic has direct applications to the cybersecurity field: it can be used to design active security response strategies during an ongoing attack. In fact, these games allow to capture the interactions between an attacker whose possible actions are modeled using an *Attack Graph* [25], and a defender able to dynamically deploy *Moving Target Defense (MTD)* mechanisms [15] based on this attack graph.

Structure of the work. The contribution is structured as follows. In Section 2, we present the syntax and the semantics of our new logic, called Obstruction Logic (OL). In Section 3, we present our case study in the context of a security scenario. Then, in Section 4 we provide some important properties of our logic, such as that memoryless and memoryfull strategies are equivalent. In Section 5 we compare OL with other logic for strategic and temporal reasoning. Subsequently, in Section 6, we show our verification procedure by a fix-point algorithm and prove that the model checking problem is decidable in polyonimal-time. We conclude by providing a related work section (see Section 7), by recapping our results, and pointing to future works (see Section 8).

2 Syntax and Semantics

In this section, we introduce the syntax and semantics of our logic. First, let us fix some notations that will be used later in the paper.

Notation 1. *If $\pi = x_1, \ldots, x_n$ is a finite sequence, $last(\pi)$ denotes the last element x_n of π; we use $\sqsubseteq$ (resp. $\sqsubset$) to denote the prefix relation (resp. strict prefix relation). If X is a set, $|X|$ denotes its cardinality and $\overline{X}$ its complement. By convention, we consider that the set of natural numbers $\mathbb{N}$ does not contain 0. We refer to the set of natural numbers containing 0 as $\mathbb{N}^0$.*

* Corresponding Author. Email: davide.catta@telecom-paris.fr

Now, we present the syntax of our logic.

Definition 1. *Let* Ap *be an at most countable set of atomic formulae (or atoms). Formulae of Obstruction Logic (OL, for short) are defined by the following grammar:*

$$\varphi ::= \top \mid p \mid \neg\varphi \mid \varphi \wedge \varphi \mid \langle \dagger_n \rangle \mathsf{X}\,\varphi \mid \langle \dagger_n \rangle (\varphi\,\mathsf{U}\,\varphi) \mid \langle \dagger_n \rangle (\varphi\,\mathsf{R}\,\varphi)$$

where p is an atomic formula and n is any number in $\mathbb{N}^0$.

The number n is called *the grade* of the strategic operator. In what follows we use small case letters from the end of the roman alphabet p, q, r, etc., to denote atoms and greek letters φ and ψ (eventually indexed by natural numbers), to denote arbitrary formulas. The boolean connectives $\bot$, $\vee$ and $\rightarrow$ can be defined as usual, we define $\langle \dagger_n \rangle \mathsf{F}\varphi := \langle \dagger_n \rangle (\top\,\mathsf{U}\,\varphi)$, $\langle \dagger_n \rangle \mathsf{G}\,\varphi := \langle \dagger_n \rangle (\bot\,\mathsf{R}\,\varphi)$ and $\langle \dagger_n \rangle (\varphi\,\mathsf{W}\,\psi) := \langle \dagger_n \rangle (\psi\,\mathsf{R}\,(\varphi \vee \psi))$. The size $|\varphi|$ of a formula φ is the number of its connectives.

The intuitive meaning of a formula $\langle \dagger \rangle \varphi$ with φ temporal formula is "there is a demonic strategy such that all paths of the graphs that are compatible with the strategy satisfy φ" where "demonic strategy" means "a strategy for disabling arcs". Formulae of OL will be interpreted over obstruction models. The definition follows.

Definition 2. *A Kripke Structure is a tuple $\langle S, R, \mathcal{L} \rangle$ where S is a finite, non-empty set of states, $R \subseteq S \times S$ is a binary serial relation over S (i.e., for any $s \in S$ there is a $s' \in S$ such that $\langle s, s' \rangle \in R$) and $\mathcal{L} : S \rightarrow 2^{\mathcal{A}}$ is a labeling function assigning a set of atomic propositions to any state $s \in S$. An obstruction model $\mathfrak{M}$ (model for short) is given by $\langle S, R, \mathcal{L}, C \rangle$ where $\langle S, R, \mathcal{L} \rangle$ is a Kripke structure and $\mathsf{C} : R \rightarrow \mathbb{N}$ is a function assigning to any $\langle s, s' \rangle \in R$ an $n \in \mathbb{N}$.*

If $\mathfrak{M}$ is a model and s one of its states, we define $pre(s) = \{s' \in S : \langle s', s \rangle \in R\}$, $post(s) = \{s' \in S : \langle s, s' \rangle \in R\}$ and $R(s) = \{e \in R : e = \langle s, s' \rangle$ for some $s' \in S\}$. According to Definition 2, a model is a finite, directed graph, in which nodes are labeled by a set of atomic formulae and edges are labeled by positive integers. Thus, we will use the term nodes and states as synonymous, we will refer to members of R as edges or arcs, and we will call elements of $post(s)$ (resp. $pre(s)$) successors of s (resp. predecessors).

A path π over a model $\mathfrak{M}$ is an infinite sequence of states $s_1, s_2, \ldots$ such that $\langle s_i, s_{i+1} \rangle \in R$ for all $i \in \mathbb{N}$. If π is a path, we write π_i to denote the i-th element s_i of π, $\pi_{\leq i}$ to denote the prefix $s_1, \ldots, s_i$ of π and $\pi_{\geq i}$ to denote the suffix $s_i, s_{i+1} \ldots$ of π. Along the paper, we use π, τ, σ, and ρ to denote paths. A *history* is any finite prefix of some path. We use H to denote the set of histories.

As said in the introduction, our logic aims to capture strategies for a particular type of games played over a directed graph; in such games, one of the two players (the Demon) has the power to temporally deactivate arcs of the graph. Thus, we can define an arc-removing strategy as follows.

Definition 3. *If $\mathfrak{M}$ is a model, and n a natural number, a n-strategy is a function $\mathfrak{S} : H \rightarrow 2^R$ that given an history h, returns a subset E of R such that: (i) $E \subset R(last(h))$, (ii) $(\sum_{e \in E} \mathsf{C}(e)) \leq n$. A memoryless n-strategy is a n-strategy $\mathfrak{S}$ such that for all histories h and h' if $last(h) = last(h')$ then $\mathfrak{S}(h) = \mathfrak{S}(h')$.*

As it is usual, one can see a memoryless n-strategy as a function whose domain is the set S of states of a model $\mathfrak{M}$.

As it happens for the logic ATL, the notion of path that is compatible with a strategy, is the central pivot of the semantic of OL formulae. We define this notion by saying that: a path π is compatible with a n-strategy $\mathfrak{S}$ if for all $i \geq 1$ we have that $\langle \pi_i, \pi_{i+1} \rangle \notin \mathfrak{S}(\pi_{\leq i})$. Given a state s and a n-strategy $\mathfrak{S}$, $Out(s, \mathfrak{S})$ denotes the set of paths whose first state is s and that are compatible with $\mathfrak{S}$. We can now precisely define the semantics of OL formulae.

Definition 4. *The satisfaction relation between a model $\mathfrak{M}$, a state s of $\mathfrak{M}$, and a formula φ is defined by induction on the structure of φ:*

- $\mathfrak{M}, s \models \top$ *for all state s;*
- $\mathfrak{M}, s \models p$ *iff $p \in \mathcal{L}(s)$;*
- $\mathfrak{M}, s \models \neg\varphi$ *iff not $\mathfrak{M}, s \models \varphi$ (notation $\mathfrak{M}, s \not\models \varphi$);*
- $\mathfrak{M}, s \models \langle \dagger_n \rangle \mathsf{X}\,\varphi$ *iff there is a n-strategy $\mathfrak{S}$ such that for all $\pi \in Out(s, \mathfrak{S})$ we have that $\mathfrak{M}, \pi_2 \models \varphi$;*
- $\mathfrak{M}, s \models \langle \dagger_n \rangle (\varphi\,\mathsf{U}\,\psi)$ *iff there is a n-strategy $\mathfrak{S}$ such that for all $\pi \in Out(s, \mathfrak{S})$ there is a $j \in \mathbb{N}$ such that $\mathfrak{M}, \pi_j \models \psi$ and for all $1 \leq k < j$, $\mathfrak{M}, \pi_k \models \varphi$;*
- $\mathfrak{M}, s \models \langle \dagger_n \rangle (\varphi\,\mathsf{R}\,\psi)$ *iff there is a n-strategy $\mathfrak{S}$ such that for all $\pi \in Out(s, \mathfrak{S})$ we have that either $\mathfrak{M}, \pi_i \models \psi$ for all $i \in \mathbb{N}$ or there is a $k \in \mathbb{N}$ such that $\mathfrak{M}, \pi_k \models \varphi$ and $\mathfrak{M}, \pi_i \models \psi$ for all $1 \leq i \leq k$.*

The memoryless satisfaction relation $\mathfrak{M}, s \models_r \varphi$ is defined by writing memoryless n-strategies instead of n-strategies in the above definition. Two formulas φ and ψ are semantically equivalent (denoted by $\varphi \equiv \psi$) iff for any model $\mathfrak{M}$ and state s of $\mathfrak{M}$, $\mathfrak{M}, s \models \varphi$ iff $\mathfrak{M}, s \models \psi$.

Definition 5. *Given $\mathfrak{M}$, s, and φ, the* local *model-checking concerns determining whether $\mathfrak{M}, s \models \varphi$. Given $\mathfrak{M}$ and φ, the* global *model-checking problem concerns determining the set $\{s \in S : \mathfrak{M}, s \models \varphi\}$.*

3 Case study

A number of research studies attempt to propose models of security threats corresponding to multi-step attack scenarios. Amongst those, one of the most used formalism is *Attack Graphs* (for a recent survey, see for instance [25]). An attack graph is generated given a description of the system architecture (topology, configurations of components, etc.) together with the list of existing vulnerabilities, the attacker's profile (his capability, known passwords, privileges, etc.), and attack templates (attacker's atomic action, including preconditions and postconditions). A path in the graph corresponds to a sequence of atomic attacks.

Once generated, attack graphs may then be used to perform a security analysis. It could be a static analysis for the computation of security metrics or selection of an optimal security hardening policy. It could also be a dynamic analysis to define an optimal security attack/response policy during an ongoing attack scenario. In this paper we consider the latter case, and assume that the defender can dynamically prevent an attack using active defense mechanisms such as *Moving Target Defense (MTD)* mechanisms (see for instance [15] for a survey). Based on some security objectives defined as properties on the attack graph, we would like to be able to check whether there exists for the defender a response strategy based on MTD mechanisms that prevents the attacker from violating the security objectives.

Background on attack graphs In this paper, we consider that an attack graph is a labeled oriented graph, where:

- each node represents both the state of the system (including existing vulnerabilities) and the state of the attacker including constants (attacker skills, financial resources, etc.) and variables (obtained privilege level, obtained credentials, etc.);
- each edge represents an action of the attacker (a scan of the network, the execution of an exploit based on a given vulnerability, access to a device, etc.) that changes the state of the network or

the states of the attacker; an edge is labeled with the name of the action (several edges of the attack graph may have the same label).

Figure 1 gives an example of an attack graph. States of the attack graph are denoted as s_i, with $0 \leq i \leq 5$, and atomic attacks as edge labels a_j, with $1 \leq i \leq 7$.

Background on MTDs MTD mechanisms uses reconfiguration techniques to dynamically shifts the attack surface in order to decrease the success probability of an attack. We consider that for each attack step a in the attack graph, there exists a corresponding MTD mechanism d_a able to counter it. During an ongoing attack, when the attacker tries to perform a, if the defender decides to activate d_a, then a will fail (i.e. the attacker will not reach the corresponding terminal state). However, the effect of d_a will be temporary: if the attacker tries to launch again a later and the defender does not activate again d_a, a will succeed. Regarding the attack graph, it means that the defender is able to temporarily remove an (or a subset of) edge(s).

We assign a cost to each MTD, corresponding to the impact on the system due to the reconfiguration phase. We assume that, at a given moment, when deploying a set of MTD countermeasures, the corresponding sum of costs is under a given threshold. This threshold corresponds to the maximal budget in terms of MTD countermeasures that the defender can deploy for each reaction.

Notice that, from the defender's point of view the cost represents the impact due to the deployment of one (or several) MTD mechanism(s) at a given moment. Such mechanisms necessitate a reconfiguration of the system and may have an impact on the availability of the services offered by the system. So, we need to ensure at each iteration that the non-availability of the system due to the reconfiguration is less than a given time period, thus that the cost is less than a given threshold.

Security objectives and assumptions In this context, the defender defines *Security Objectives* based on the attack graph. Let suppose that when reaching state s_1, s_3, or s_5 the attacker has root privilege on a given critical server s Let suppose that, if the attacker completes attack steps a_6 or a_7 (that is, it reaches state s_5), then the defender will obtain information on the identity of the attacker In this example two security objective could be analyzed:

O_1 *the attacker is never able to obtain root privilege on server s unless the defender is able to obtain information on its identity*;

O_2 *while the defender has not obtained information about the attacker identity, the attacker has not root privilege on server s.*

Based on a given attack graph and security objectives, we would like to check whether there are MTD response strategies such that the security objectives are satisfied. To achieve this, we assume that:

1. The defender always knows the attack graph state reached by the attacker (called *attacker current state*).
2. At every moment, there is a unique attacker current state in the attack graph.
3. When detecting the attacker current state, the defender can activate a (or a subset of) MTD(s) temporarily removing an (a subset of) outgoing edge(s). The defender cannot remove edges that are not outgoing from the attacker current state.
4. The sum of the costs associated to the subset of MTDs activated is less than a given threshold.
5. When the attacker launches an attack from its current state, if the corresponding edge has not been removed by the defender, then the attack always succeeds (i.e. the attacker reaches the next state).
6. When the attacker launches an attack from its current state, if the corresponding edge has been removed by the defender, then the attack always fails (i.e. the attacker stays in its current state).

Specification of security objectives Let a be an atomic proposition that express the fact that the identity of the attacker is known. Let r_s be an atomic proposition expressing the fact that the attacker has root privilege on server s. The two security objectives O_1 and O_2 presented above can be expressed by OL formulae. Objective O_1 says that either we want the attacker to never reach a state satisfying r_s *or* if the attacker reach such a state then the defender wants to be able to identify it (a). By using t_1 as variable for a given threshold, the following OL formula captures O_1: $\varphi_1 := \langle \dagger_{t_1} \rangle \mathsf{G}\,(\neg r_s \vee (r_s \rightarrow \langle \dagger_{t_1} \rangle (\mathsf{F}\, a)))$.

Objective O_2 says that we want s_r to be false *until* we have identified the attacker (a) if such an identification ever happens. Thus, by using t_2 as a variable for a given threshold, we can write O_2 using the weak-until connective: $\varphi_2 := \langle \dagger_{t_2} \rangle (\neg r_s \,\mathsf{W}\, a)$.

Remark that it is fairly easy to see the Attack Graph of Figure 1 as a Model in the sense of Definition 2. We can simply forget the edge labels, add a loop-edge on s_5 to grant seriality, add a cost for each edge (representing the defender's cost to apply the corresponding MTD countermeasure), and specify the labeling function as showed in Figure 1. Suppose that t_1 and t_2 are respectively 3 and 5. In the obtained model $\mathfrak{M}$ we have that $\mathfrak{M}, s_0 \models \varphi_1 \wedge \varphi_2$. To satisfy φ_1 consider the 3-memoryless strategy $\mathfrak{S}_1$ that associates $\{\langle s_1, s_2 \rangle\}$ to s_1, $\{\langle s_3, s_4 \rangle\}$ to s_3, and $\emptyset$ to any other state of $\mathfrak{M}$. Remark that for any path $\pi \in Out(\mathfrak{S}_1, s_0)$ and any $i \in \mathbb{N}$ we have that $\mathfrak{M}, \pi_i \models r_s$ iff $\pi_i \in \{s_1, s_3, s_5\}$. Thus, we must establish that $\mathfrak{M}$ satisfies $\langle \dagger_n \rangle \mathsf{F} a$ on s_1 (resp. s_3 and s_5). To do so, we remark that $Out(\mathfrak{S}_1, s_1)$ (resp. $Out(\mathfrak{S}_1, s_3)$ and $Out(\mathfrak{S}_1, s_5)$) only contains the path s_1, s_3, s_5^ω (resp. s_3, s_5^ω and s_5^ω) and that $\mathfrak{M}, s_5 \models a$. Thus, we have obtained that there is a strategy (i.e. $\mathfrak{S}_1$) such that for all $\pi \in Out(\mathfrak{S}_1, s_0)$ and all $i \in \mathbb{N}$ either $\mathfrak{M}, \pi_i \models \neg r_s$ or if $\mathfrak{M}, \pi_i \models r_s$ then there is a strategy ($\mathfrak{S}_1$ itself) such that $\mathfrak{M}, \rho_j \models a$ for some $j \geq 1$ and for all $\rho \in Out(\mathfrak{S}_1, \pi_i)$, as we wanted. Remark that if $t_1 < 3$ then it is not possible to satisfy φ_1 in $\mathfrak{M}$ at s_0. For the specification $\varphi_2 = \langle \dagger_1 \rangle (\neg r_s \,\mathsf{W}\, a)$, consider the 4-memoryless strategy $\mathfrak{S}_2$ that associates $\{\langle s_0, s_1 \rangle\}$ to s_0, $\{\langle s_2, s_1 \rangle, \langle s_2, s_3 \rangle\}$ to s_2, $\{\langle s_4, s_3 \rangle\}$ to s_4 and $\emptyset$ to s_5. The only path in $Out(s_0, \mathfrak{S}^\star)$ is $s_0, s_2, s_4, s_5^\omega$ and since s_5 satisfies a and all the other s_i do not satisfy r_s we obtain the wanted result.

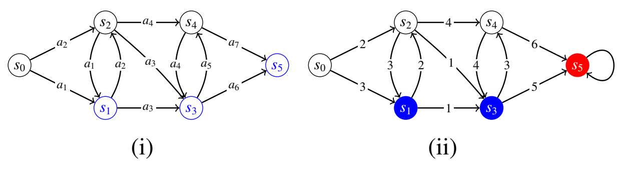

Figure 1: (i) An Attack Graph where states s_1, s_3 and s_5 represents the goals of the attacker. (ii) The model $\mathfrak{M}$ obtained from (i) where the blue nodes satisfies r_s, the red node satisfies both a and r_s, and the white ones satisfy neither r_s nor a.

4 Main Properties

In this section, we first prove some remarkable semantic equivalences between OL formulae. We then show that the satisfaction relation and the memoryless satisfaction relation coincides.

Let $\mathfrak{M} = \langle S, R, \mathcal{L}, \mathsf{C} \rangle$ be a model. We recall that if s is a state $post(s)$ and $pre(s)$ denote, respectively, the set of successors of s and the set of predecessors of s. Given a set of states A, we define $Pre(A) = \bigcup_{s \in A} pre(s)$.

If s is a state, n a natural number, and A a set of states, we write $\blacktriangleright(s, n, A)$ whenever $(\sum_{s' \in A} \mathsf{C}(\langle s, s' \rangle)) \leq n$. If A is a set of states, $\blacktriangledown(n, A)$ denotes the subset of S defined by $\{s \in Pre(A) : \blacktriangleright(s, n, \overline{A})\}$. Let φ be any formula: $\llbracket \varphi \rrbracket^{\mathfrak{M}}$ denotes the set of states of $\mathfrak{M}$ verifying, φ, i.e., $\llbracket \varphi \rrbracket^{\mathfrak{M}} = \{s \in S : \mathfrak{M}, s \models \varphi\}$. We omit the superscript $\mathfrak{M}$ when the model is clear from the context. We are now ready to characterize the set of states satisfying a formula $\langle \nmid_n \rangle \mathsf{X}\, \varphi$.

Proposition 1. *For every model $\mathfrak{M}$, state s, and formula φ we have that $\mathfrak{M}, s \models \langle \nmid_n \rangle \mathsf{X}\, \varphi$ if and only if $s \in \blacktriangledown(n, \llbracket \varphi \rrbracket)$).*

Proof. For the ($\Rightarrow$)-direction: suppose that $s \in \llbracket \langle \nmid_n \rangle \mathsf{X}\, \varphi \rrbracket$, thus there is a n-strategy $\mathfrak{S}$ such that for all $\pi \in Out(s, \mathfrak{S})$ we have that $\mathfrak{M}, \pi_2 \models \varphi$. Suppose, to reach a contradiction, that $s \notin \blacktriangledown(n, \llbracket \varphi \rrbracket)$. This could either mean that $s \notin Pre(\varphi)$ or that $s \in Pre(\varphi)$ and that $\sum_{s' \in \overline{\llbracket \varphi \rrbracket}} \mathsf{C}(s, s') > n$. In the first case, we immediately obtain a contradiction. In the second case, we remark that any n-strategy can deactivate a subset of directed edges whose cost is at most n. Thus given any n-strategy $\mathfrak{S}'$ there is a path $\pi \in Out(s, \mathfrak{S}')$ such that $\pi_2 \in \overline{\llbracket \varphi \rrbracket}$, which contradicts that $\mathfrak{M}, s \models \langle \nmid_n \rangle \mathsf{X}\, \varphi$.

For the ($\Leftarrow$)-direction: suppose that $s \in \blacktriangledown(n, \llbracket \varphi \rrbracket)$. By definition, this means that $s \in Pre(\llbracket \varphi \rrbracket)$ and that if $e_1, \ldots, e_n$ are all the edges from s to $\overline{\llbracket \varphi \rrbracket}$, then $(\mathsf{C}(e_1) + \cdots + \mathsf{C}(e_n)) \leq n$. Thus, a n-strategy verifying $\langle \nmid_n \rangle \mathsf{X}\, \varphi$ is readily obtained by selecting the set $\{e_1, \ldots, e_n\}$ given the history $h = s$, and the empty-set for any other history h'. □

In the next proposition, we provide the fix-point characterization for the OL operators.

Proposition 2. *For any two formulae φ and ψ we have that:*

1. $\langle \nmid_n \rangle (\varphi \mathbin{\mathsf{U}} \psi) \equiv \psi \vee (\varphi \wedge \langle \nmid_n \rangle \mathsf{X}\, (\langle \nmid_n \rangle (\varphi \mathbin{\mathsf{U}} \psi)))$;
2. $\langle \nmid_n \rangle (\varphi \mathbin{\mathsf{R}} \psi) \equiv \psi \wedge (\varphi \vee \langle \nmid_n \rangle \mathsf{X}\, (\langle \nmid_n \rangle (\varphi \mathbin{\mathsf{R}} \psi)))$.

Proof. We only prove (1), the proof of (2) being entirely similar.

($\Rightarrow$) Suppose that $\mathfrak{M}, s \models \langle \nmid_n \rangle (\varphi \mathbin{\mathsf{U}} \psi)$, thus there is a n-strategy $\mathfrak{S}$, such that for every $\tau \in Out(s, \mathfrak{S})$ there is a $j \in \mathbb{N}$ such that $\mathfrak{M}, \tau_j \models \psi$ and $\mathfrak{M}, \tau_k \models \varphi$ for every $1 \leq k < j$. If $\mathfrak{M}, s \models \psi$ then we can conclude, otherwise since $s = \tau_1$ for any $\tau \in Out(s, \mathfrak{S})$, we must have that $\mathfrak{M}, s \models \varphi$ by hypothesis. Consider the n-strategy $\mathfrak{S}'$ defined by:

$$\mathfrak{S}'(h) = \begin{cases} \mathfrak{S}(h') & \text{if } h' = sh \text{ and } h' \sqsubset \tau \text{ for } \tau \in Out(s, \mathfrak{S}) \\ \emptyset & \text{otherwise.} \end{cases}$$

Let $C = \{s' : \exists \tau \in Out(s, \mathfrak{S}) \text{ s.t. } \tau_2 = s'\}$; given any $s' \in C$ we clearly have that $\rho \models \varphi \mathbin{\mathsf{U}} \psi$ for any $\rho \in Out(s', \mathfrak{S}')$. We deduce that $\mathfrak{M}, s' \models \langle \nmid_n \rangle \varphi \mathbin{\mathsf{U}} \psi$ (the Demon can use the above defined n-strategy $\mathfrak{S}'$) and since the Demon can get to any of the s' by disabling the successors of s disabled by $\mathfrak{S}$ we conclude that $\mathfrak{M}, s \models \langle \nmid_n \rangle \mathsf{X}\, (\langle \nmid_n \rangle (\varphi \mathbin{\mathsf{U}} \psi))$.

($\Leftarrow$) Suppose that $\mathfrak{M}, s \models \psi \vee (\varphi \wedge \langle \nmid \rangle \mathsf{X}\, (\langle \nmid \rangle (\varphi \mathbin{\mathsf{U}} \psi)))$. If $\mathfrak{M}, s \models \psi$ then we can conclude. Otherwise $\mathfrak{M}, s \models \varphi$ and $\mathfrak{M}, s \models \langle \nmid_n \rangle \mathsf{X}\, (\langle \nmid_n \rangle (\varphi \mathbin{\mathsf{U}} \psi))$. By the definition of satisfaction, this means that there is a n-strategy $\mathfrak{S}$ such that $\mathfrak{M}, \rho_2 \models \langle \nmid \rangle\ \varphi \mathbin{\mathsf{U}} \psi$ for any $\rho \in Out(s, \mathfrak{S})$. By applying again the definition of satisfaction, we get that there is a n-strategy $\mathfrak{S}_{\rho_2}$ such that for every $\tau \in Out(\rho_2, \mathfrak{S}_{\rho_2})$ we have that $\mathfrak{M}, \tau_k \models \psi$ for some $k \geq 1$ and $\mathfrak{M}, \tau_j \models \varphi$ for any $1 \leq j < k$. We define another n-strategy $\mathfrak{S}''$ by

$$\mathfrak{S}''(h) = \begin{cases} \mathfrak{S}(h) & \text{if } h = s \\ \mathfrak{S}_{\pi_2}(h') & \text{if } h' = sh \text{ and } h \sqsubset \tau \text{ for} \\ & \tau \in Out(\pi_2, \mathfrak{S}_{\pi_2}) \text{ and } \pi \in Out(s, \mathfrak{S}) \\ \emptyset & \text{otherwise} \end{cases}$$

By the definition of $\mathfrak{S}''$, we obtain that $s\rho \in Out(s, \mathfrak{S}'')$ implies $\rho \in Out(s, \mathfrak{S}_{\pi_2})$ for some $\pi \in Out(s, \mathfrak{S})$. Now: suppose that there is a $\tau \in (s, \mathfrak{S}'')$ such that (1) $\mathfrak{M}, \tau_j \not\models \psi$ for all $j \in \mathbb{N}$ or such that (2) $\mathfrak{M}, \tau_k \models \psi$ for some $k \in \mathbb{N}$, and there is a $1 \leq j < k$ such that $\mathfrak{M}, \tau_j \not\models \varphi$. In both cases we get a contradiction because such a path also belongs to $Out(\pi_2, \mathfrak{S}_{\pi_2})$ for some $\pi \in Out(s, \mathfrak{S})$. We can thus finally conclude that $\mathfrak{M}, s \models \langle \nmid_n \rangle (\varphi \mathbin{\mathsf{U}} \psi)$ as we wanted. □

Let $\mathfrak{M} = \langle S, R, \mathcal{L}, C \rangle$ be a model and φ, ψ be two formulae. Consider the two monotone functions $\mathsf{R}^{\mathsf{n}}{}_{\varphi,\psi}$ and $\mathsf{U}^{\mathsf{n}}{}_{\varphi,\psi}$ from 2^S to itself, for any subset X of S, defined by:

$$\mathsf{U}^n_{\varphi,\psi}(X) = \llbracket \psi \rrbracket^{\mathfrak{M}} \cup (\llbracket \varphi \rrbracket^{\mathfrak{M}} \cap \blacktriangledown(n, X)) \quad (1)$$
$$\mathsf{R}^n_{\varphi,\psi}(X) = \llbracket \psi \rrbracket^{\mathfrak{M}} \cap (\llbracket \varphi \rrbracket^{\mathfrak{M}} \cup \blacktriangledown(n, X)) \quad (2)$$

we can prove the following.

Theorem 1. *For every model $\mathfrak{M}$ and two formulae φ and ψ:*

1. *$\llbracket \langle \nmid_n \rangle (\varphi \mathbin{\mathsf{U}} \psi) \rrbracket^{\mathfrak{M}}$ is the least fix-point of $\mathsf{U}^n_{\varphi,\psi}$;*
2. *$\llbracket \langle \nmid_n \rangle (\varphi \mathbin{\mathsf{R}} \psi) \rrbracket^{\mathfrak{M}}$ is the greatest fix-point of $\mathsf{R}^n_{\varphi,\psi}$.*

Proof. We only give a proof of (2). In virtue of Propositions 1 and 2 it is clear that $X = \llbracket \langle \nmid_n \rangle (\varphi \mathbin{\mathsf{R}} \psi) \rrbracket^{\mathfrak{M}}$ is a fix-point of $\mathsf{R}^n_{\varphi,\psi}$. Let Y be any other fix-point of the function. If $Y = \emptyset$ the result trivially holds. Otherwise, we show that given $v \in Y$ we have that $v \in X$. Since Y is a fix-point of the function, we have that $v \in \llbracket \psi \rrbracket$ and either $v \in \llbracket \varphi \rrbracket$ or $v \in \blacktriangledown(n, Y)$. If this last case, for the set E_v of edges whose source is v and whose target is not in Y we have that $(\sum_{e \in E_v} C(e)) \leq n$. We define an n-memoryless strategy $\mathfrak{S}$ by

$$\mathfrak{S}(v) = \begin{cases} E_v & \text{if } v \in \llbracket \psi \rrbracket \cap \blacktriangledown(n, Y) \\ \emptyset & \text{otherwise} \end{cases}$$

Now consider a path $\rho \in Out(v, \mathfrak{S})$ for $v \in Y$. Since $\rho_1 = v \in Y$, let $\mathcal{X} = \{k \geq 2 : \rho_k \notin Y\}$. If $\mathcal{X}$ is empty, then each $\rho_i \in \llbracket \psi \rrbracket$, otherwise let $j \geq 2$ be the smallest integer such that $\rho_j \notin Y$. Suppose to reach a contradiction that $\rho_{j-1} \notin \llbracket \varphi \rrbracket$. Since $\rho_{j-1} \in Y$ and $\rho_j \notin \llbracket \varphi \rrbracket$ we must have that $\rho_j \in \llbracket \psi \rrbracket \cap \blacktriangledown(n, Y)$. Thus, given any $\pi \in Out(\rho_j, \mathfrak{S})$ we obtain that $\pi_2 \in Y$. Since $\mathfrak{S}$ is memoryless, we obtain a contradiction because $\pi_2 = \rho_j$ for some $\pi \in Out(\rho_{j-1}, \mathfrak{S})$. □

As we have anticipated, we now show that the set of formulae that are true under the satisfaction relation and the memoryless satisfaction relation coincides.

Lemma 1. *For any formula φ, for any model $\mathfrak{M}$ and state s: if $\mathfrak{M}, s \models \varphi$ then $\mathfrak{M}, s \models_r \varphi$.*

Proof. The proof is by induction on the structure of φ. Remark that for the case $\varphi = \langle \nmid_n \rangle \psi_1 \mathbin{\mathsf{R}} \psi_2$, we showed in Theorem 1 how given any v belonging to a fix-point of the function $\mathsf{R}^n_{\psi_1,\psi_2}$, it is possible to construct a n-memoryless strategy $\mathfrak{S}$ such that all paths in $Out(v, \mathfrak{S})$ satisfies $\psi_1 \mathbin{\mathsf{R}} \psi_2$. As a consequence, we only detail the cases $\varphi = \langle \nmid_n \rangle \mathsf{X}\, \psi$ and $\varphi = \langle \nmid_n \rangle (\psi_1 \mathbin{\mathsf{U}} \psi_2)$. The proof for the $\langle \nmid_n \rangle (\psi_1 \mathbin{\mathsf{U}} \psi_2)$ case is inspired by the one presented in [26].

If $\varphi = \langle \nmid_n \rangle \mathsf{X}\, \psi$, $\mathfrak{M}, s \models \varphi$ iff there is a n-strategy $\mathfrak{S}$ such that $\langle \mathfrak{M}, \rho_2 \rangle \models \psi$ for all $\rho \in Out(s, \mathfrak{S})$. By induction hypothesis, $\mathfrak{M}, \rho_2 \models \psi$. Consider the n-strategy $\mathfrak{S}'$ such that $\mathfrak{S}'(h) = \mathfrak{S}(s)$ if $last(h) = s$ and $\mathfrak{S}'(h) = \emptyset$ otherwise. This latter n-strategy is memoryless, and for any $\tau \in Out(s, \mathfrak{S}')$, $\tau_2 = \rho_2$ which establish the wanted result.

If $\varphi = \langle \nmid \rangle \psi_1 \mathbin{\mathsf{U}} \psi_2$. Let $\mathfrak{S}$ be a n-strategy such that $\psi_1 \mathbin{\mathsf{U}} \psi_2$ holds for any path $\pi \in Out(s, \mathfrak{S})$. Let $\mathcal{X} = \{\pi_{\leq k} : \pi \in Out(s, \mathfrak{S}) \wedge \mathfrak{M}, \pi_k \models \psi_2 \wedge \mathfrak{M}, \pi_j \models \psi_1 \text{ for all } 1 \leq j < k\}$ and $\mathcal{Y}$ its prefix-closure. Given

$v \in S$, let h_v be a sequence in $\mathcal{Y}$ such that $last(h_v) = v$ and $\nexists h^\star \in \mathcal{Y}$ such that $h_v \sqsubset h^\star$ and $last(h^\star) = v$. Define a n-strategy $\mathfrak{S}_{h_v}$ by:

$$\mathfrak{S}_{h_v}(h) = \begin{cases} \mathfrak{S}(h_v \cdot h'') & \text{if } h = h' \cdot h'' \text{ and } last(h') = v \\ \mathfrak{S}(h) & \text{otherwise} \end{cases}$$

Such a n-strategy associates the same set of edges to any history ending in v and we clearly have that any path in its outcome set verifies $\psi_1 \,\mathsf{U}\, \psi_2$. Repeat the above procedure for any $s' \in S$. After at most $|S|$ steps, we obtain a memoryless n-strategy $\mathfrak{S}'$ in which any path verifies $\psi_1 \,\mathsf{U}\, \psi_2$. □

By the above lemma and by the fact that any memoryless n-strategy is a n-strategy, one immediately obtains the following.

Theorem 2. *For any formula φ, for every model $\mathfrak{M}$ and state s, we have that $\mathfrak{M}, s \models_r$ if and only if $\mathfrak{M}, s \models \varphi$.*

5 Relationship With Other Logics

5.1 Obstruction Logic and Computation Tree Logic

We here prove that OL extends Computation Tree Logic (CTL)[17] with a reduction to a fragment of our logic. We define the 0-fragment of OL to be the set of OL formulae in which the grade of any strategic operator is 0. We denote by OL^0 such a fragment. Let $(-)^\bullet$ be the function from OL^0 to CTL formulae that substitutes each strategic operator $\langle \nvdash_0 \rangle$ with the universal path operator A of CTL, i.e., the function recursively defined as follows:

$$\begin{array}{lcl}
(\top)^\bullet & = & \top \\
(p)^\bullet & = & p \\
(\neg\varphi)^\bullet & = & \neg(\varphi)^\bullet \\
(\varphi_1 \wedge \varphi_2)^\bullet & = & (\varphi_1)^\bullet \wedge (\varphi_2)^\bullet \\
(\langle \nvdash_0 \rangle \mathsf{X}\, \varphi)^\bullet & = & \mathsf{AX}\, (\varphi)^\bullet \\
(\langle \nvdash_0 \rangle (\varphi_1 \,\mathsf{U}\, \varphi_2))^\bullet & = & \mathsf{A}((\varphi_1)^\bullet \,\mathsf{U}\, (\varphi_2)^\bullet) \\
(\langle \nvdash_0 \rangle (\varphi_1 \,\mathsf{R}\, \varphi_2))^\bullet & = & \mathsf{A}(\varphi_1)^\bullet \,\mathsf{R}\, (\varphi_2)^\bullet
\end{array}$$

Remark that the function $(-)^\bullet$ induces a bijection between OL and CTL formulae.

Theorem 3. *For every model $\mathfrak{M}$, state s, and formula $\varphi \in OL^0$, we have that $\mathfrak{M}, s \models \varphi$ if and only if $\mathfrak{M}, s \models_{CTL} (\varphi)^\bullet$, where $\models_{CTL}$ is the CTL satisfaction relation.*

5.2 Obstruction Logic and Alternating-time Temporal Logic

Here, we compare OL with ATL. In particular, we show that given an OL formula φ and a model $\mathfrak{M}$ that satisfies it, there is a CGS (exponential in the size of $\mathfrak{M}$) that satisfies an ATL translation of φ.

First, define a rooted OL model as a pair $\langle \mathfrak{M}, s \rangle$ where $\mathfrak{M}$ is an OL model and s is one of its states. Given a natural number n, let $S^{\leq n}$ be the subset of $S \times 2^R$ defined by $(s, E) \in S^{\leq n}$ iff either $E = \emptyset$ or each $e \in E$ has s as source and $(\sum_{e \in E} C(e)) \leq n$. If $\langle \mathfrak{M}, s \rangle$ is a rooted OL model and n is a natural number, then $\mathfrak{C}^n_{\mathfrak{M}} = \langle \mathsf{Ap}, \mathsf{Ag}, Q, q_i, act_D, act_T, P, \delta, \mathcal{V} \rangle$ is the CGS, where:

- Ap is a set of atomic formulae labeling states of $\mathfrak{M}$;
- $\mathsf{Ag} = \{D, T\}$ where D is the Demon and T is the Traveler;
- $Q = Q_D \cup Q_T$ is a set of states, where $Q_D = S$ and $Q_T = S^{\leq n}$. Moreover, $q_I = s$ is the initial state. The set Q_D is the set of states where is the Demon's turn to move, while Q_T is the set of states in which is the Traveler's turn to move;
- the set of actions act_D of the Demon is equal to the set of subset of R appearing in $S^{\leq n}$ plus the idle action $\star$. More precisely $act_D = \{E \in 2^R : \exists q \in S^{\leq n} \wedge q = \langle s, E \rangle\} \cup \{\star\}$;
- the set of actions act_T of the Traveler is $R \cup \{\star\}$. We denote by $act = act_D \cup act_T$;
- the protocol function $P : Q \times \mathsf{Ag} \to 2^{act} \setminus \emptyset$ is defined as follows. For every $q \in Q_D$, we have that $P(q, i)$ is equal to $X_q = \{E \in 2^R : \langle q, E \rangle \in S^{\leq n}\}$ if $i = D$, and $\{\star\}$ otherwise. For every $q \in Q_T$, we have that if $q = \langle s, E \rangle$ then $P(q, i)$ is equal to $\{e \in R : e \notin E \wedge s' \in S\}$ if $i = T$ and it is equal to $\{\star\}$ otherwise;
- the transition function $\delta : Q \times act_D \times act_T \to Q$ is defined as follows: $\delta(q, E, \star) = \langle q, E \rangle$ iff $q \in Q_D$ and $\delta(q, \langle s, s' \rangle, \star) = s'$ iff $q = \langle s, E \rangle \in Q_T$ and $\langle s, s' \rangle \notin E$;
- the labeling function $V : S \to 2^{\mathsf{Ap}}$ is defined by $V(q) = \mathcal{L}(q)$ for any $q \in Q_D$ and $V(q) = \emptyset$ for any $q \in Q_T$.

Remark that given a model $\mathfrak{M}$ and a natural number n, the CGS $\mathfrak{C}^n_{\mathfrak{M}}$ can have a number of states that is **exponential** in the number of states of $\mathfrak{M}$. Consider the function from OL formulae to ATL formulae, inductively defined by:

$$\begin{array}{lcl}
(\top)^{\mathsf{A}} & = & \top \\
(p)^{\mathsf{A}} & = & p \\
(\neg\varphi)^{\mathsf{A}} & = & \neg(\varphi)^{\mathsf{A}} \\
(\varphi_1 \wedge \varphi_2)^{\mathsf{A}} & = & (\varphi_1)^{\mathsf{A}} \wedge (\varphi_2)^{\mathsf{A}} \\
(\langle \nvdash_n \rangle \mathsf{X}\, \varphi)^{\mathsf{A}} & = & \langle\!\langle D \rangle\!\rangle \mathsf{X}\, (\varphi)^{\mathsf{A}} \\
(\langle \nvdash_n \rangle (\varphi_1 \,\mathsf{U}\, \varphi_2))^{\mathsf{A}} & = & \langle\!\langle D \rangle\!\rangle (\varphi)^{\mathsf{A}} \,\mathsf{U}\, (\psi)^{\mathsf{A}} \\
(\langle \nvdash_n \rangle (\varphi_1 \,\mathsf{R}\, \varphi_2)))^{\mathsf{A}} & = & \langle\!\langle D \rangle\!\rangle (\varphi)^{\mathsf{A}} \,\mathsf{R}\, (\psi)^{\mathsf{A}}
\end{array}$$

Given a CGS $\mathfrak{C}^n_{\mathfrak{M}}$ as the one defined above, and a path ρ of the CGS, we write ρ^D for the subsequence of ρ containing only states that are in Q_D. If Δ is an ATL strategy and $q \in Q_D$ is a state, then $Out^D(q, \Delta)$ denotes the set of sequences $\{\rho \in Q_D^\omega : \rho = \pi^D \text{ for some } \pi \in Out(q, \Delta)\}$. For an ATL formula ψ, we write $\mathfrak{C}^n_{\mathfrak{M}}, q \models_D \psi$ iff either:

1. ψ is a boolean formula and $\mathfrak{C}^n_{\mathfrak{M}}, q \models_{ATL} \psi$, where $\models_{ATL}$ is the standard ATL satisfiability relation or
2. ψ is a strategic formula $\langle\!\langle D \rangle\!\rangle \psi_1$ and there is a strategy Δ such that for all $\rho \in Out^D(q, \Delta)$ we have that ρ satisfies ψ_1 (where the specific clauses for the temporal connectives X, R and U can be easily obtained).

We can now prove the following the lemma.

Lemma 2. *Let φ be any OL formula that contains at most a strategic operator $\langle \nvdash_n \rangle$, we have that $\mathfrak{M}, s \models \varphi$ iff $\mathfrak{C}^n_{\mathfrak{M}}, s \models_D (\varphi)^{\mathsf{A}}$.*

Remark that we can use the classic bottom-up approach to generalize the above lemma to any OL strategic formula φ. We thus obtain the following result.

Theorem 4. *For each strategic formula φ whose main operator rank is n we have that $\mathfrak{M}, s \models \varphi$ iff $\mathfrak{C}^n_{\mathfrak{M}}, s \models_D (\varphi)^{\mathsf{A}}$.*

5.3 Obstruction Logic and Graded Mu Calculus

In this subsection, we compare OL with graded μ-calculus (GMC for short) introduced by Kupferman, Sattler, and Vardi in [29]. More precisely, we show how to translate each OL formula φ to a GMC formula $(\varphi)^\mu$ and that given an OL model $\mathfrak{M}$ such that $C(e) = 1$ for all $e \in R$, we have that $[\![\varphi]\!]^{\mathfrak{M}} = [\![(\varphi)^\mu]\!]^{\mathfrak{M}}$.

Since GMC is less known than CTL and ATL, we briefly introduce its syntax and semantics. GMC is a propositional modal logic augmented with least (μ) and greatest (ν) fix-point operators and with the two *graded modalities* $[\,n\,]$ and $\langle\,n\,\rangle$. More formally, let Ap be a non-empty at most countable set of atomic propositions, and $\mathcal{V}$ a non-empty at most countable set of propositional variables and suppose that Ap and $\mathcal{V}$ are disjoints. One can define formulae of the graded μ-calculus using the following grammar:

$$\varphi ::= \top \mid p \mid X \mid \neg\varphi \mid \varphi \wedge \varphi \mid [\,n\,]\varphi \mid \mu X.(\varphi)$$

where $p \in \mathsf{Ap}$, $X \in \mathcal{V}$, and $n \in \mathbb{N}^0$. In $\mu X.\varphi$ we require that the variable X appears in the scope of an even number of negations in φ. An occurrence of a propositional variable that is under the scope of μ is said to be bound. Free variables are variables that are not bound. A GMC formula is closed if all its variables are bound.

Let φ and ψ two formulae, and suppose that no variable that is bound in φ is free in ψ[1]. We write $\varphi[\psi/X]$ to denote the result of the substitution of ψ to each free occurrence of X in φ. The greatest fixpoint operator can be defined by $\nu X.\varphi = \neg(\mu X.(\neg(\varphi[\neg X/X])))$ and the graded diamond modality can be defined as $\langle\,n\,\rangle\varphi = \neg[\,n\,]\neg\varphi$.

Formulae of GMC are interpreted over Kripke Structures. In particular, given a Kripke structure $\mathfrak{K} = \langle S, R, \mathcal{L}\rangle$, an assignment $\alpha : \mathcal{V} \to 2^S$, is a function that sends propositional variables to subsets of S. Given an assignment α, a subset T of S, and a variable Y, $\alpha[X \to T]$ is the assignment defined by $\alpha[X \to T](Y) = T$ if $Y = X$ and $\alpha[X \to T](Y) = \alpha(Y)$ otherwise. Given a Kripke structure $\mathfrak{K}$, the denotation $[\![\varphi]\!]^{\mathfrak{K}}_{\alpha}$ of a formula φ under α in $\mathfrak{K}$ is the subset of S inductively defined on the structure of φ by the following clauses:

$$\begin{array}{lcl}
[\![\top]\!]^{\mathfrak{K}}_{\alpha} & = & S; \\
[\![p]\!]^{\mathfrak{K}}_{\alpha} & = & \{s \in S \mid p \in \mathcal{L}(s)\}; \\
[\![X]\!]^{\mathfrak{K}}_{\alpha} & = & \alpha(X) \\
[\![\neg\psi]\!]^{\mathfrak{K}}_{\alpha} & = & \overline{[\![\psi]\!]^{\mathfrak{K}}_{\alpha}} \\
[\![\psi \wedge \theta]\!]^{\mathfrak{K}}_{\alpha} & = & [\![\psi]\!]^{\mathfrak{K}}_{\alpha} \cap [\![\theta]\!]^{\mathfrak{K}}_{\alpha}; \\
[\![[\,n\,]\psi]\!]^{\mathfrak{K}}_{\alpha} & = & \{s \in Pre([\![\psi]\!]^{\mathfrak{K}}_{\alpha}) \,:\, |post(s) \cap [\![\neg\psi]\!]^{\mathfrak{K}}_{\alpha}| \leq n\} \\
[\![\mu X.\psi]\!]^{\mathfrak{K}}_{\alpha} & = & \bigcap\{T \subseteq S \,:\, [\![\psi]\!]^{\mathfrak{K}}_{\alpha[X\to T]} \subseteq T\}.
\end{array}$$

Thus, the meaning of a formula $\psi = [\,n\,]\,\varphi$ can be spelled out as ψ is true at s if there are at most n-successors of s in which φ is not true. Remark that if φ is closed, one can simply write $[\![\varphi]\!]^{\mathfrak{K}}$ as the denotation of φ does not depend on any assignment. Now, let $(-)^{\mu}$ be the function from OL formulae to GMC formulae defined as follows:

$$\begin{array}{lcl}
(\top)^{\mu} & = & \top \\
(p)^{\mu} & = & p \\
(\neg\varphi)^{\mu} & = & \neg(\varphi)^{\mu} \\
(\varphi_1 \wedge \varphi_2)^{\mu} & = & (\varphi_1)^{\mu} \wedge (\varphi_2)^{\mu} \\
(\langle\dashv_n\rangle \mathsf{X}\,\varphi)^{\mu} & = & [\,n\,](\varphi)^{\mu} \\
(\langle\dashv_n\rangle(\varphi_1 \,\mathsf{U}\, \varphi_2))^{\mu} & = & \mu X.((\varphi_2)^{\mu} \vee ((\varphi_1)^{\mu} \wedge [\,n\,]X)) \\
(\langle\dashv_n\rangle(\varphi_1 \,\mathsf{R}\, \varphi_2)))^{\mu} & = & \nu X((\varphi_2)^{\mu} \wedge ((\varphi_1)^{\mu} \vee [\,n\,]X))
\end{array}$$

Remark that $(\varphi)^{\mu}$ is a closed GMC formula for every formula φ, thus the function $(-)^{\mu}$ has as image a proper fragment of GMC. Now: let us call *unary* an OL Model $\mathfrak{M} = \langle S, R, \mathcal{L}, C\rangle$ such that $C(e) = 1$ for all $e \in R$. We have the following

Lemma 3. *For any OL formula φ, if $\mathfrak{M}$ is a unary model, then we have that: $[\![\langle\dashv_n\rangle \mathsf{X}\,\varphi]\!]^{\mathfrak{M}} = \{s \in Pre([\![\varphi]\!]^{\mathfrak{M}}) \,:\, |post(s) \cap [\![\neg\varphi]\!]^{\mathfrak{M}}| \leq n\}$*

From the above lemma, one can easily prove the following by induction on the structure of φ using Theorem 1.

[1] One can always respect this constraint by renaming the bound variables of φ.

Theorem 5. *If $\mathfrak{M}$ is a unary OL model then for every OL formula φ we have that $[\![\varphi]\!]^{\mathfrak{M}} = [\![(\varphi)^{\mu}]\!]^{\mathfrak{M}}$.*

Note that the above theorem is false when the model is not unary. For instance, consider a model where $S = \{s1, s2\}$, $R = \{\langle s1, s1\rangle, \langle s1, s2\rangle, \langle s2, s2\rangle\}$ with $\mathcal{L}(s1) = \{q\}$, $\mathcal{L}(s2) = \{p\}$ and where the cost of each arc is 2. At s_1, the OL formula $\langle\dashv_1\rangle \mathsf{X}\,p$ is false, while its translation in GMC $[\,1\,]\,p$ is true.

6 Model Checking

Here, we show that the global model-checking problem for OL is decidable in polynomial-time. To show this result, we provide 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 an extension of the one for CTL.

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

```
1: for all $\varphi \in Sub(\varphi)$ do
2:     switch $\varphi$ do
3:         case $\varphi = \top$
4:             $[\![\varphi]\!] \leftarrow S$
5:         case $\varphi = p$
6:             $[\![\varphi]\!] \leftarrow \{s \in S \,:\, p \in \mathcal{L}(s)\}$
7:         case $\varphi = \neg\varphi_1$
8:             $[\![\varphi]\!] \leftarrow S \setminus [\![\varphi_1]\!]$
9:         case $\varphi = \varphi_1 \wedge \varphi_2$
10:            $[\![\varphi]\!] \leftarrow [\![\varphi_1]\!] \cap [\![\varphi_2]\!]$
11:        case $\varphi = \langle\dashv_n\rangle \mathsf{X}\,\varphi_1$
12:            $[\![\varphi]\!] \leftarrow \blacktriangledown(n, [\![\varphi_1]\!])$
13:        case $\varphi = \langle\dashv_n\rangle(\varphi_1 \,\mathsf{U}\, \varphi_2)$
14:            $X \leftarrow \emptyset$; $Y \leftarrow [\![\varphi_2]\!]$
15:            while $Y \neq X$ do
16:                $X \leftarrow Y$
17:                $Y \leftarrow [\![\varphi_2]\!] \cup ([\![\varphi_1]\!] \cap \blacktriangledown(n, X))$
18:            $[\![\varphi]\!] \leftarrow Y$
19:        case $\varphi = \langle\dashv_n\rangle(\varphi_1 \,\mathsf{R}\, \varphi_2)$
20:            $X \leftarrow [\![\top]\!]$; $Y \leftarrow [\![\varphi_2]\!]$
21:            while $X \neq Y$ do
22:                $X \leftarrow Y$
23:                $Y \leftarrow [\![\varphi_2]\!] \cap ([\![\varphi_1]\!] \cup \blacktriangledown(n, X))$
24:            $[\![\varphi]\!] \leftarrow Y$
```

Theorem 6. *The model checking problem for Obstruction Logic (OL) is P-complete.*

Proof. For the lowerbound, recall that CTL corresponds to the OL^0 fragment of OL. Then, the results follows immediately from the P-hardness of model checking CTL [17].

For the upperbound, Algorithm 1 shows a procedure for model checking OL, which manipulates set of states of S. The procedure is inspired by the model checking for CTL [17] and ATL [4]. However, we use two additional procedures $\blacktriangleright$ and $\blacktriangledown$ linked to the pre-image function Pre. In detail, our algorithm uses the following functions:

- The function Sub returns an ordered sequence, w.r.t. their complexities, of syntactic sub-formulas of a given formula φ.
- The function Pre is the same as for CTL [17].

- The function $\blacktriangleright(s, n, A)$ takes in input a state s, a natural numbers n, and a subset of states A. Such a function returns true if $(\sum_{s' \in A} \mathsf{C}(\langle s, s' \rangle)) < n$. If we represent the graph via an adjacent matrix, we can calculate such function in a linear number of steps w.r.t. the size of A.
- The function $\blacktriangledown(n, A)$ takes in input a natural number n and a subset of states A. The function returns the subset A' of $Pre(A)$, such that $\blacktriangleright(s', n, \overline{A})$ for all $s' \in A'$. The worst possible case is when $Pre(A) = S$, and one needs to call $|S|$-times the function $\blacktriangleright$. So, we are quadratic in S, i.e. polynomial.

Algorithm 1 works bottom-up on the structure of the formula; the cases of interest are for strategic formulas. For $\varphi = \langle \nplus_n \rangle \mathsf{X}\, \varphi_1$, the procedure calls function $\blacktriangledown(n, [\![\varphi]\!])$ to compute the subset of set of states of $Pre([\![\varphi_1]\!])$ that are "bound" to end up in satisfaction set. As regard $\varphi = \langle \nplus_n \rangle (\varphi_1 \,\mathsf{U}\, \varphi_2)$, the procedure computes the least fixed-point. We observe that, since it is monotone, such a fixed-point always exists. A similar reasoning can be done for $\varphi = \langle \nplus_n \rangle \varphi_1 \,\mathsf{R}\, \varphi_2$.

From the above, our procedure runs in polynomial-time in the size of the model and formula. 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 1. □

7 Related Work

In the past years, many works focused on the strategic abilities of agents playing in a dynamic game model. For instance, the authors of [35] consider the problem of planning paths in a dynamic but predictable environment. The authors of this work, have been shown that determining whether a given agent can construct a path in the graph reaching its goal is exponential in the size of the graph and number of players, while we provide a polynomial-time algorithm to solve our games. Furthermore, they don't give the ability to the agents to select a specific subset of successors to be occupied, while in our approach the Demon is able to do that. In our games the Demon can prevent the Traveler to occupy a given position only temporarily, while in [35] the occupation is permanent. Moreover, in our games subset of edges are deactivated with respect to quantitative information.

Sabotage games and Sabotage Modal Logic [38, 31, 5] are another line of research that our work is related to. 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 which an agent has the ability to erase edges. To reason about sabotage games, van Benthem introduced Sabotage Modal Logic (SML). Determining whether the player who's moving as a winning strategy for reaching a certain position by playing a Sabotage Game is PSPACE-hard [32] and the model-checking problem for Sabotage Modal Logic is PSPACE-complete [31]. Our version of games is incomparable with Sabotage games since we give the ability to temporarily select subsets of edges while in Sabotage games the saboteur can erase only one edge at each turn. On this respect, our work is related to [13] in which the authors use an extended version of Sabotage Modal Logic, called Subset Sabotage Modal Logic (SSML), in which the deactivation of particular subsets of edges of a directed graph is allowed. The authors show that the model-checking problem for such a logic is decidable, but they do not specify the complexity class of such a decision problem. Furthermore, we recall that SSML is an extension of SML, but it does not include temporal operators as we do. Moreover, neither SML nor SSML take into account quantitative information about the cost of edges as we do.

The authors of [37] introduce Dynamic Escape Games (DEG, for short). Such games have a close resemblance to our games. A DEG is a variant of weighted two-player turn-based reachability games in which an agent has the ability to inhibit edges. Contrary to our approach, on this class of games the authors have proposed an optimized heuristic that provides partial results to check whether the reachability player has a strategy to reach one of his goal states.

In [1] the authors introduce NTL a temporal logic to reason about normative systems. A normative system is a Kripke structure in which certain transition are considered illegal, see [2] for a survey. Formally, in NTL one evaluates CTL formulae with respect to a Kripke model in which a set of arcs has been deleted according to a given assignment function. The assignment function on NTL in non-local e non-quantitative: any subset of arcs can be deleted by the assignment, and there is no notion of deletion cost. Moreover, OL model checking is in P while NTL model checking is in NP.

Module checking [30, 9, 8] is another line of work which is related to our logic. Although there is a similarity between module checking and OL model checking for the 1-fragment on unary models, the two approaches are orthogonal. In OL each state of the model can be seen as a state that is controlled by the environment (the Demon). Furthermore, in OL we ask whether there is a winning strategy for the environment and not whether all environment strategies are winning. This difference is found in the fact that the model checking problem for OL is polynomial, while the module checking problem (even for "simple" logics such as CTL) is at least exponential.

From the cybersecurity side, several works propose game-theoretic solutions for finding an optimal defense policy based on attack graphs. Most of these approaches do not use formal verification, but rather try to solve the game using analytic and optimization techniques, e.g., [19, 20, 36, 39]. The works in [10, 14, 12] share some ideas with ours on the cybersecurity side. However, the authors do not use dynamic models. In addition, the already mentioned work in [11, 13] propose a logical approach to play on Attack Graphs.

8 Conclusions

In this paper, we introduced Obstruction Logic, a logic that allows to reason about two-player games with temporal goals in which one of the players has the power to modify, locally and temporarily, the game structure. Then, we showed how to express cybersecurity properties via OL. We studied the formal properties of OL and proved that its model-checking problem is solvable in polynomial-time.

In the future, we can explore different directions. One natural extension, would be to consider many-player games, between a Demon and *coalitions* of Travelers. At any step a possible continuation of a play would be determined, by the Demon's deactivation, the synchronous actions of the considered coalition of Travelers T, and all possible actions of Travelers not in T. We believe that such an extension would entertain with the logic ATL, the same relationship that OL entertain with CTL. Another extension we would like to study is to permit the Demon to *permanently* deactivate edges of the directed graph. In this case, Demon's actions could impact the topology of the graph in a non-local fashion: the Demon could choose to erase an edge situated anywhere in the graph. The logic so obtained, would reassemble to a temporal version of the already cited Sabotage Modal Logic [38]. Finally, we would like to study the above scenarios in the context of imperfect information. Unfortunately, this context is in general undecidable [18]. To overcome this problem, we could use an approximation to perfect information [6], a notion of bounded memory [7], or some hybrid technique [21, 22].

References

[1] Thomas Ågotnes, Wiebe van der Hoek, Juan A. Rodríguez-Aguilar, Carles Sierra, and Michael J. Wooldridge, 'On the logic of normative systems', in *IJCAI 2007, Proceedings of the 20th International Joint Conference on Artificial Intelligence 2007*, ed., Manuela M. Veloso, pp. 1175–1180, (2007).

[2] Natasha Alechina, Brian S. Logan, and Mehdi M. Dastani, 'Modeling norm specification and verification in multiagent systems', *FLAP*, **5**, 457–490, (2018).

[3] R. Alur, T. A. Henzinger, and O. Kupferman, 'Alternating-time temporal logic', in *FOCS97*, pp. 100–109, (1997).

[4] R. Alur, T.A. Henzinger, and O. Kupferman, 'Alternating-time temporal logic', *J. ACM*, **49**(5), 672–713, (2002).

[5] G. Aucher, J. Van Benthem, and D. Grossi, 'Modal logics of sabotage revisited', *Journal of Logic and Computation*, **28**(2), 269 – 303, (March 2018).

[6] Francesco Belardinelli, Angelo Ferrando, and Vadim Malvone, 'An abstraction-refinement framework for verifying strategic properties in multi-agent systems with imperfect information', *Artif. Intell.*, **316**, 103847, (2023).

[7] Francesco Belardinelli, Alessio Lomuscio, Vadim Malvone, and Emily Yu, 'Approximating perfect recall when model checking strategic abilities: Theory and applications', *J. Artif. Intell. Res.*, **73**, 897–932, (2022).

[8] Laura Bozzelli and Aniello Murano, 'On the complexity of ATL and atl* module checking', in *Proceedings Eighth International Symposium on Games, Automata, Logics and Formal Verification, GandALF 2017*, eds., Patricia Bouyer, Andrea Orlandini, and Pierluigi San Pietro, volume 256 of *EPTCS*, pp. 268–282, (2017).

[9] Laura Bozzelli, Aniello Murano, and Adriano Peron, 'Module checking of pushdown multi-agent systems', in *Proceedings of the 17th International Conference on Principles of Knowledge Representation and Reasoning, KR 2020*, eds., Diego Calvanese, Esra Erdem, and Michael Thielscher, pp. 162–171, (2020).

[10] Elie Bursztein and Jean Goubault-Larrecq, '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*, ed., Iliano Cervesato, volume 4846 of *LNCS*, pp. 212–227. Springer, (2007).

[11] Davide Catta, Jean Leneutre, and Vadim Malvone, 'Subset sabotage games & attack graphs', in *Proceedings of the 23rd Workshop "From Objects to Agents"*, volume 3261, pp. 209–218. CEUR-WS.org, (2022).

[12] Davide Catta, Jean Leneutre, and Vadim Malvone, 'Towards a formal verification of attack graphs', in *Proceedings of SPIRIT 2022*, volume 3345 of *CEUR Workshop Proceedings*. CEUR-WS.org, (2022).

[13] Davide Catta, Jean Leneutre, and Vadim Malvone, 'Attack graphs & subset sabotage games', *Intelligenza Artificiale*, **17**(1), 77–88, (2023).

[14] Davide Catta, Antonio Di Stasio, Jean Leneutre, Vadim Malvone, and Aniello Murano, 'A game theoretic approach to attack graphs', in *ICAART 2023,*, pp. 347–354. SCITEPRESS, (2023).

[15] Jin-Hee Cho, Dilli Sharma, Hooman Alavizadeh, Seunghyun Yoon, Noam Ben-Asher, Terrence Moore, Dan Kim, Hyuk Lim, and Frederica Nelson, 'Toward proactive, adaptive defense: A survey on moving target defense', *IEEE Communications Surveys & Tutorials*, **PP**, 1–1, (01 2020).

[16] E. M. Clarke, O. Grumberg, and D. A. Peled, *Model Checking*, The MIT Press, Cambridge, Massachusetts, 1999.

[17] E.M. Clarke and E.A. Emerson, 'Design and Synthesis of Synchronization Skeletons Using Branching-Time Temporal Logic.', in *81*, LNCS 131, pp. 52–71. Springer, (1981).

[18] C. Dima and F.L. Tiplea, 'Model-checking ATL under imperfect information and perfect recall semantics is undecidable', *CoRR*, **abs/1102.4225**, (2011).

[19] Karel Durkota, Viliam Lisý, Branislav Bosanský, and Christopher Kiekintveld, 'Approximate solutions for attack graph games with imperfect information', in *Decision and Game Theory for Security - 6th International Conference, GameSec 2015*, eds., M. H. R. Khouzani, Emmanouil A. Panaousis, and George Theodorakopoulos, volume 9406 of *LNCS*, pp. 228–249. Springer, (2015).

[20] Karel Durkota, Viliam Lisý, Branislav Bosanský, and Christopher Kiekintveld, 'Optimal network security hardening using attack graph games', in *Proceedings of the Twenty-Fourth International Joint Conference on Artificial Intelligence, IJCAI 2015*, eds., Qiang Yang and Michael J. Wooldridge, pp. 526–532. AAAI Press, (2015).

[21] Angelo Ferrando and Vadim Malvone, 'Towards the combination of model checking and runtime verification on multi-agent systems', in *20th International Conference, PAAMS 2022*, eds., Frank Dignum, Philippe Mathieu, Juan Manuel Corchado, and Fernando de la Prieta, volume 13616 of *LNCS*, pp. 140–152. Springer, (2022).

[22] Angelo Ferrando and Vadim Malvone, '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*, eds., Noa Agmon, Bo An, Alessandro Ricci, and William Yeoh, pp. 793–801. ACM, (2023).

[23] W. Jamroga and A. Murano, 'Module checking of strategic ability', in *AAMAS 2015*, pp. 227–235, (2015).

[24] N. R. Jennings and M. Wooldridge, 'Application of intelligent agents', in *Agent Technology: Foundations, Applications, and Markets*, Springer-Verlag, (1998).

[25] Kerem Kaynar, 'A taxonomy for attack graph generation and usage in network security', *J. Inf. Secur. Appl.*, **29**(C), 27–56, (August 2016).

[26] Michal Knapik, Étienne André, Laure Petrucci, Wojciech Jamroga, and Wojciech Penczek, 'Timed ATL: forget memory, just count', *J. Artif. Intell. Res.*, **66**, 197–223, (2019).

[27] O. Kupferman, M.Y. Vardi, and P. Wolper, 'An Automata Theoretic Approach to Branching-Time ModelChecking.', *Journal of the ACM*, **47**(2), 312–360, (2000).

[28] O. Kupferman, M.Y. Vardi, and P. Wolper, 'Module Checking.', *Information and Computation*, **164**(2), 322–344, (2001).

[29] Orna Kupferman, Ulrike Sattler, and Moshe Y. Vardi, 'The complexity of the graded μ-calculus', in *Automated Deduction - CADE-18, 18th International Conference on Automated Deduction 2002*, ed., Andrei Voronkov, volume 2392 of *LNCS*, pp. 423–437. Springer, (2002).

[30] Orna Kupferman and Moshe Y. Vardi, 'Module checking', in *Computer Aided Verification, 8th International Conference, CAV '96, New Brunswick, NJ, USA, July 31 - August 3, 1996, Proceedings*, eds., Rajeev Alur and Thomas A. Henzinger, volume 1102 of *LNCS*, pp. 75–86. Springer, (1996).

[31] Christof Löding and Philipp Rohde, 'Model checking and satisfiability for sabotage modal logic', in *FST TCS 2003: Foundations of Software Technology and Theoretical Computer Science*, eds., Paritosh K. Pandya and Jaikumar Radhakrishnan, volume 2914 of *LNCS*, pp. 302–313. Springer, (2003).

[32] Christof Löding and Philipp Rohde, 'Solving the sabotage game is pspace-hard', in *Mathematical Foundations of Computer Science 2003, 28th International Symposium, MFCS 2003*, eds., Branislav Rovan and Peter Vojtás, volume 2747 of *LNCS*, pp. 531–540. Springer, (2003).

[33] A. Lomuscio, H. Qu, and F. Raimondi, 'MCMAS: A model checker for the verification of multi-agent systems', in *Proceedings of the 21th International Conference on Computer Aided Verification (CAV09)*, volume 5643 of *LNCS*, pp. 682–688. Springer, (2009).

[34] F. Mogavero, A. Murano, G. Perelli, and M. Y. Vardi, 'Reasoning about strategies: On the model-checking problem', *ACM Transactions in Computational Logic*, **15**(4), 34:1–34:47, (2014).

[35] Aniello Murano, Giuseppe Perelli, and Sasha Rubin, 'Multi-agent path planning in known dynamic environments', in *PRIMA 2015: Principles and Practice of Multi-Agent Systems - 18th International Conference*, eds., Qingliang Chen, Paolo Torroni, Serena Villata, Jane Yung-jen Hsu, and Andrea Omicini, volume 9387 of *LNCS*, pp. 218–231. Springer, (2015).

[36] Thanh H. Nguyen, Mason Wright, Michael P. Wellman, and Satinder Baveja, 'Multi-stage attack graph security games: Heuristic strategies, with empirical game-theoretic analysis', MTD '17, p. 87–97, New York, NY, USA, (2017). Association for Computing Machinery.

[37] Antonio Di Stasio, Paolo Domenico Lambiase, Vadim Malvone, and Aniello Murano, 'Dynamic escape game', in *Proceedings of the 17th International Conference on Autonomous Agents and MultiAgent Systems, AAMAS 2018*, eds., Elisabeth André, Sven Koenig, Mehdi Dastani, and Gita Sukthankar, pp. 1806–1808. ACM, (2018).

[38] J. van Benthem, *An Essay on Sabotage and Obstruction*, 268–276, Springer Berlin Heidelberg, Berlin, Heidelberg, 2005.

[39] Yunxiao Zhang and Pasquale Malacaria, 'Bayesian stackelberg games for cyber-security decision support', *Decis. Support Syst.*, **148**, 113599, (2021).