Reasoning About Dynamic Game Models Using Obstruction Logic (short paper)

Davide Catta, Jean Leneutre, Vadim Malvone

Reasoning About Dynamic Game Models Using Obstruction Logic*

Davide Catta[1,*,†], Jean Leneutre[2,†] and Vadim Malvone[2,†]

[1]Università degli studi di Napoli "Federico II", Naples, Italy
[2]LTCI, Télécom Paris, Institut Polytechnique de Paris, Palaiseau, France

Abstract

Games played within dynamic models have been explored in various domains, including cybersecurity and planning. Our paper introduces Obstruction Logic, a formalism designed for analyzing specific games featuring temporal objectives, which unfold within dynamic models. These games involve players whose actions can impact the underlying game model. We demonstrate how this logic can be employed to express significant properties within the realm of cybersecurity games, particularly those defined on attack graphs. An expanded version of our research has been published in ECAI 2023.

Keywords

Strategic Reasoning, Dynamic games, Attack Graphs

1. Introduction

Multi-agent systems capture the behavior of two or more rational agents engaged in interactions with each other, whether through cooperation or adversarial interactions, all with the aim of achieving a specific objective [2]. Typically, this behavior is represented using a combination of temporal or modal logic and game theory. In this framework, agents are treated as players in games played on directed graphs known as arenas, and their goals are defined using logical formulas. For example, logics like ATL and Strategy Logic [3, 4] provide the means to express the idea that a coalition of players can reach a particular goal through cooperative actions.

In these logics, the game model, within which players participate, is regarded as a fixed entity. While players' actions affect their positions within the arena, they do not alter the underlying structure of the arena itself. In contrast, dynamic games, where the game model is subject to change, have been examined in various contexts, including cybersecurity and planning [5, 6, 7, 8].

This paper presents a logic designed for the analysis of a specific category of games with temporal objectives played within a dynamic model. These games involve two key players: the Demon and the Traveler, and are conducted on a directed graph. Each edge e in the graph is associated with a deactivation cost $C(e)$. The game unfolds in rounds, with each round

IPS-RCRA-SPIRIT 2023: Italian Workshop on Planning and Scheduling, RCRA Workshop on Experimental evaluation of algorithms for solving problems with combinatorial explosion, and SPIRIT Workshop on Strategies, Prediction, Interaction, and Reasoning in Italy. November 7-9th, 2023, Rome, Italy [1]

*Corresponding author.
†These authors contributed equally.

consisting of a move by the Demon followed by a move by the Traveler. In the Demon's move, given a node v in the graph and a natural number n, the Demon deactivates a proper subset E of the edges incident to v in such a way that the sum of the deactivation costs of the edges in E is less or equal to n. Subsequently, the Traveler selects a node v' connected to v, such that the edge $\langle v, v' \rangle$ does not belong to E. A new round commences from the last node chosen by the Traveler, and the edges disabled in the previous round are reactivated. The Demon wins the play if the infinite sequence of nodes subsequently chosen by the Traveler satisfies a given temporal property. To reason about the existence of "demonic" strategies for this kind of games, we propose Obstruction Logic (or simply OL). In OL, one can quantify over the existence of demonic strategies that allows the Demon to temporally obstruct some reachable states. OL 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 capturing the interactions between an attacker whose possible actions are modeled using an *Attack Graph* [9], and a defender able to dynamically deploy *Moving Target Defense (MTD)* mechanisms [10, 11] based on this attack graph.

Related Work In the past years, many works focused on the strategic abilities of agents playing in a dynamic game model [5, 6, 12]. We compare some of this works to OL.

Sabotage games and Sabotage Modal Logic [13, 14, 15] are a 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). 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 [8] 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. 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.

In [12] 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 [16] 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.

The works in [17, 18, 19] share some ideas with ours on the cybersecurity side. However, the authors do not use dynamic models.

2. Syntax, Semantics, and Main Properties

In this section, we introduce the syntax and semantics of our logic. 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 non-negative integer. The number n is called *the grade* of the strategic operator. In what follows we use greek letters φ and ψ (eventually indexed by natural numbers), to denote arbitrary formulas. The boolean connectives $\bot$, $\vee$, and $\rightarrow$ and the temporal connectives F, W, and G can be defined as usual. Formulae of OL will be interpreted over obstruction models. The definition follows.

Definition 1. *An obstruction model $\mathfrak{M}$ (model for short) is given by $\langle S, R, \mathscr{L}, \mathscr{C}\rangle$ where $\langle S, R, \mathscr{L}\rangle$ is a Kripke structure in which the accessibility relation R is serial, and $\mathrm{C} : R \rightarrow \mathbb{N}$ is a function assigning to any $\langle s, s'\rangle \in R$ a positive integer n.*

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. A *history* is any finite prefix of some path. We use H to denote the set of histories.

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". We define an arc-removing strategy as follows.

Definition 2. *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 a history h, returns a subset E of R such that: (i) $E \subset R(last(h))$[1] , (ii) $(\sum_{e\in E} \mathrm{C}(e)) \leq n$. A memoryless n-strategy is a n-strategy $\mathfrak{S}$ such that for all histories h and h' if $last(h) = last(h')$ then $\mathfrak{S}(h) = \mathfrak{S}(h')$.*

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 $\mathfrak{S}$.

Definition 3. *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*[2]*:*

- $\mathfrak{M}, s \models \langle\dagger_n\rangle \mathsf{X}\,\varphi$ *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 $1 \leq i \leq k$.*

[1] *last*(h) denotes the last element of the finite sequence h.

[2] the clauses for $\top$, atomic propositions, and boolean connectives are the usual ones and thus omitted.

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 $\mathfrak{M}, s \models \psi$. OL enjoys some important theoretical properties.

Theorem 1. *[20] Formulae that are true under the satisfaction relation and the memoryless satisfaction relation coincides, that is: for any formula φ, for every model $\mathfrak{M}$ and state s, we have that $\mathfrak{M}, s \models_r$ if and only if $\mathfrak{M}, s \models \varphi$.*

Let $[\![\varphi]\!]^{\mathfrak{M}} = \{s \in S \mid \mathfrak{M}, s \models \varphi\}$, we can characterize $[\![\langle\dagger_n\rangle \mathsf{X}\,\varphi]\!]^{\mathfrak{M}}$ as follows: given $X \subseteq S$ we write $s \in \dagger(n, X)$ iff the following holds:

$$s \in Pre(X) \wedge \Big(\sum_{s' \in \overline{X}} \mathsf{C}\langle s, s'\rangle\Big) \leq n$$

where $Pre(X) = \{s \mid \exists s' \in X \wedge \langle s, s'\rangle \in R\}$. Given the above characterization of $\dagger(-,-)$, we can easily prove that $s \in [\![\langle\dagger_n\rangle \mathsf{X}\,\varphi]\!]^{\mathfrak{M}}$ iff $s \in \dagger(n, [\![\varphi]\!]^{\mathfrak{M}})$. Remark that computing $\dagger(n, X)$ is polynomial in $|S|$ for any $X \subseteq S$. Given this last result, we can prove the following.

Theorem 2. *[20] Given a model $\mathfrak{M}$ and two formulae φ and ψ let $\mathsf{U}^n_{\varphi,\psi}$ and $\mathsf{R}^n_{\varphi,\psi}$ the two monotone functions from 2^S into itself defined by:*

$$\mathsf{U}^n_{\varphi,\psi}(X) = [\![\psi]\!]^{\mathfrak{M}} \cup ([\![\varphi]\!]^{\mathfrak{M}} \cap \dagger(n, X)) \tag{1}$$

$$\mathsf{R}^n_{\varphi,\psi}(X) = [\![\psi]\!]^{\mathfrak{M}} \cap ([\![\varphi]\!]^{\mathfrak{M}} \cup \dagger(n, X)) \tag{2}$$

we have that $[\![\langle\dagger_n\rangle(\varphi\,\mathsf{U}\,\psi)]\!]^{\mathfrak{M}}$ is the least fix-point of $\mathsf{U}^n_{\varphi,\psi}$ and that $[\![\langle\dagger_n\rangle(\varphi\,\mathsf{R}\,\psi)]\!]^{\mathfrak{M}}$ is the greatest fix-point of $\mathsf{R}^n_{\varphi,\psi}$.

Given the above theorem, an algorithm that computes the set of states satisfying a formula φ is obtained by slightly modifying the classical labeling algorithm of CTL [21]. Moreover, it exists a simple syntactic embedding from CTL to a proper fragment of OL (see [20] for details). We thus obtain the following.

Theorem 3. *[20] The model checking problem for Obstruction Logic (OL) is PTIME-complete, that is: given a finite model $\mathfrak{M}$ a state s of $\mathfrak{M}$, and formula φ deciding whether $\mathfrak{M}, s \models \varphi$ is PTIME-complete and can be computed quadratic time in the number $|S|$ of states of $\mathfrak{M}$.*

3. Case study

We here give an extended example of how OL can be used to reason about security scenarios modeled by means of attack graphs. 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, and 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 state of the attacker; an edge is labeled with the name of the

action. Figure 1(i) 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 $1 \leq i \leq 7$. A path in the graph corresponds to a sequence of atomic attacks. Attack graphs can be used to perform dynamic analysis to define an optimal security attack/response policy during an ongoing attack scenario, as we do here. We assume that the defender can dynamically prevent an attack using active defense mechanisms such as *Moving Target Defense (MTD)* mechanisms (see for instance [10] 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.

MTD mechanisms use 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 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. 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.

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.*

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. 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(i) as a model in the sense of Definition 1. 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(ii). If t_1 and t_2 are respectively 3 and 5, then, in the obtained model $\mathfrak{M}$ we have that $\mathfrak{M}, s_0 \models \varphi_1 \wedge \varphi_2$.

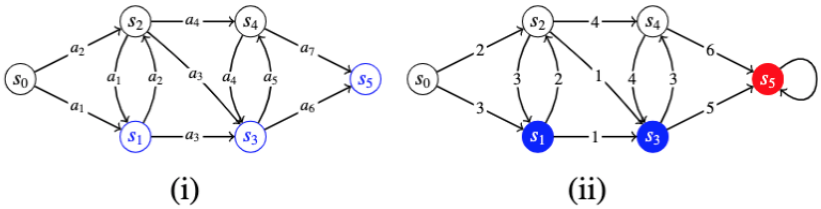

Figure 1: (i) An Attack Graph where states s_1, s_3, and s_5 represents the goals of the attacker. (ii) The model $\mathfrak{M}$ obtained from (i) where the blue nodes satisfy r_s, the red node satisfies both a and r_s, and the white ones satisfy neither r_s nor a.

4. Conclusions

We presented 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. 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. 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 [13]. Finally, we would like to study the above scenarios in the context of imperfect information. Unfortunately, this context is in general undecidable [22]. To overcome this problem, we could use an approximation to perfect information [23], a notion of bounded memory [24], or some hybrid technique [25, 26, 27].

References

[1] R. De Benedictis, M. Castiglioni, D. Ferraioli, V. Malvone, M. Maratea, E. Scala, L. Serafini, I. Serina, E.Tosello, A. Umbrico, M. Vallati, Preface to the Italian Workshop on Planning and Scheduling, RCRA Workshop on Experimental evaluation of algorithms for solving problems with combinatorial explosion, and SPIRIT Workshop on Strategies, Prediction, Interaction, and Reasoning in Italy (IPS-RCRA-SPIRIT 2023), in: Proceedings of the Italian Workshop on Planning and Scheduling, RCRA Workshop on Experimental evaluation of algorithms for solving problems with combinatorial explosion, and SPIRIT Workshop on Strategies, Prediction, Interaction, and Reasoning in Italy (IPS-RCRA-SPIRIT 2023) co-located with 22th International Conference of the Italian Association for Artificial Intelligence (AI* IA 2023), 2023.

[2] N. R. Jennings, M. Wooldridge, Application of intelligent agents, in: Agent Technology: Foundations, Applications, and Markets, Springer-Verlag, 1998.
[3] R. Alur, T. A. Henzinger, O. Kupferman, Alternating-time temporal logic, in: FOCS97, 1997, pp. 100–109.
[4] F. Mogavero, A. Murano, G. Perelli, M. Y. Vardi, Reasoning about strategies: On the model-checking problem, ACM Transactions in Computational Logic 15 (2014) 34:1–34:47. URL: http://doi.acm.org/10.1145/2631917. doi:P10.1145/2631917.
[5] A. Murano, G. Perelli, S. Rubin, Multi-agent path planning in known dynamic environments, in: Q. Chen, P. Torroni, S. Villata, J. Y. Hsu, A. Omicini (Eds.), PRIMA 2015: Principles and Practice of Multi-Agent Systems - 18th International Conference, volume 9387 of *LNCS*, Springer, 2015, pp. 218–231. URL: https://doi.org/10.1007/978-3-319-25524-8_14. doi:P10.1007/978-3-319-25524-8_14.
[6] A. D. Stasio, P. D. Lambiase, V. Malvone, A. Murano, Dynamic escape game, in: E. André, S. Koenig, M. Dastani, G. Sukthankar (Eds.), Proceedings of the 17th International Conference on Autonomous Agents and MultiAgent Systems, AAMAS 2018, ACM, 2018, pp. 1806–1808. URL: http://dl.acm.org/citation.cfm?id=3237984.
[7] D. Catta, J. Leneutre, V. Malvone, Subset sabotage games & attack graphs, in: Proceedings of the 23rd Workshop "From Objects to Agents", volume 3261, CEUR-WS.org, 2022, pp. 209–218. URL: http://ceur-ws.org/Vol-3261/paper16.pdf.
[8] D. Catta, J. Leneutre, V. Malvone, Attack graphs & subset sabotage games, Intelligenza Artificiale 17 (2023) 77–88. URL: https://doi.org/10.3233/IA-221080. doi:P10.3233/IA-221080.
[9] K. Kaynar, A taxonomy for attack graph generation and usage in network security, J. Inf. Secur. Appl. 29 (2016) 27–56.
[10] J.-H. Cho, D. Sharma, H. Alavizadeh, S. Yoon, N. Ben-Asher, T. Moore, D. Kim, H. Lim, F. Nelson, Toward proactive, adaptive defense: A survey on moving target defense, IEEE Communications Surveys & Tutorials PP (2020) 1–1.
[11] G. Ballot, V. Malvone, J. Leneutre, E. Borde, Reasoning about moving target defense in attack modeling formalisms, in: H. Okhravi, C. Wang (Eds.), Proceedings of the 9th ACM Workshop on Moving Target Defense, MTD 2022, Los Angeles, CA, USA, 7 November 2022, ACM, 2022, pp. 55–65. URL: https://doi.org/10.1145/3560828.3564009. doi:P10.1145/3560828.3564009.
[12] T. Ågotnes, W. van der Hoek, J. A. Rodríguez-Aguilar, C. Sierra, M. J. Wooldridge, On the logic of normative systems, in: M. M. Veloso (Ed.), IJCAI 2007, Proceedings of the 20th International Joint Conference on Artificial Intelligence 2007, 2007, pp. 1175–1180. URL: http://ijcai.org/Proceedings/07/Papers/190.pdf.
[13] J. van Benthem, An Essay on Sabotage and Obstruction, Springer Berlin Heidelberg, Berlin, Heidelberg, 2005, pp. 268–276. URL: https://doi.org/10.1007/978-3-540-32254-2_16. doi:P10.1007/978-3-540-32254-2$_1$6.
[14] C. Löding, P. Rohde, Model checking and satisfiability for sabotage modal logic, in: P. K. Pandya, J. Radhakrishnan (Eds.), FST TCS 2003: Foundations of Software Technology and Theoretical Computer Science, volume 2914 of *LNCS*, Springer, 2003, pp. 302–313. URL: https://doi.org/10.1007/978-3-540-24597-1_26. doi:P10.1007/978-3-540-24597-1_26.
[15] G. Aucher, J. V. Benthem, D. Grossi, Modal logics of sabotage revisited, Journal of Logic and Computation 28 (2018) 269 – 303. URL: https://hal.inria.fr/hal-01827076. doi:P10.1093/log-

com/exx034.
[16] N. Alechina, B. S. Logan, M. M. Dastani, Modeling norm specification and verification in multiagent systems, FLAP 5 (2018) 457–490.
[17] E. Bursztein, J. Goubault-Larrecq, A logical framework for evaluating network resilience against faults and attacks, in: I. Cervesato (Ed.), Advances in Computer Science - ASIAN 2007. Computer and Network Security, 12th Asian Computing Science Conference, volume 4846 of *LNCS*, Springer, 2007, pp. 212–227. URL: https://doi.org/10.1007/978-3-540-76929-3_20. doi:P10.1007/978-3-540-76929-3_20.
[18] D. Catta, A. D. Stasio, J. Leneutre, V. Malvone, A. Murano, A game theoretic approach to attack graphs, in: ICAART 2023„ SCITEPRESS, 2023, pp. 347–354. URL: https://doi.org/10.5220/0011776900003393. doi:P10.5220/0011776900003393.
[19] D. Catta, J. Leneutre, V. Malvone, Towards a formal verification of attack graphs, in: Proceedings of SPIRIT 2022, volume 3345 of *CEUR Workshop Proceedings*, CEUR-WS.org, 2022. URL: https://ceur-ws.org/Vol-3345/paper14_Spirit3.pdf.
[20] D. Catta, J. Leneutre, V. Malvone, Obstruction logic: A strategic temporal logic to reason about dynamic game models, in: K. Gal, A. Nowé, G. J. Nalepa, R. Fairstein, R. Radulescu (Eds.), ECAI 2023 - 26th European Conference on Artificial Intelligence, September 30 - October 4, 2023, Kraków, Poland - Including 12th Conference on Prestigious Applications of Intelligent Systems (PAIS 2023), volume 372 of *Frontiers in Artificial Intelligence and Applications*, IOS Press, 2023, pp. 365–372. URL: https://doi.org/10.3233/FAIA230292. doi:P10.3233/FAIA230292.
[21] E. Clarke, E. Emerson, Design and Synthesis of Synchronization Skeletons Using Branching-Time Temporal Logic., in: 81, LNCS 131, Springer, 1981, pp. 52–71.
[22] C. Dima, F. Tiplea, Model-checking ATL under imperfect information and perfect recall semantics is undecidable, CoRR abs/1102.4225 (2011). URL: http://arxiv.org/abs/1102.4225.
[23] F. Belardinelli, A. Ferrando, V. Malvone, An abstraction-refinement framework for verifying strategic properties in multi-agent systems with imperfect information, Artif. Intell. 316 (2023) 103847. URL: https://doi.org/10.1016/j.artint.2022.103847. doi:P10.1016/j.artint.2022.103847.
[24] F. Belardinelli, A. Lomuscio, V. Malvone, E. Yu, Approximating perfect recall when model checking strategic abilities: Theory and applications, J. Artif. Intell. Res. 73 (2022) 897–932. URL: https://doi.org/10.1613/jair.1.12539. doi:P10.1613/jair.1.12539.
[25] A. Ferrando, V. Malvone, Towards the combination of model checking and runtime verification on multi-agent systems, in: F. Dignum, P. Mathieu, J. M. Corchado, F. de la Prieta (Eds.), 20th International Conference, PAAMS 2022, volume 13616 of *LNCS*, Springer, 2022, pp. 140–152. URL: https://doi.org/10.1007/978-3-031-18192-4_12. doi:P10.1007/978-3-031-18192-4_12.
[26] A. Ferrando, V. Malvone, Towards the verification of strategic properties in multi-agent systems with imperfect information, in: N. Agmon, B. An, A. Ricci, W. Yeoh (Eds.), Proceedings of the 2023 International Conference on Autonomous Agents and Multiagent Systems, AAMAS 2023, ACM, 2023, pp. 793–801. URL: https://dl.acm.org/doi/10.5555/3545946.3598713. doi:P10.5555/3545946.3598713.
[27] A. Ferrando, V. Malvone, Give me a hand: How to use model checking for multi-agent systems to help runtime verification and vice versa (short paper), in: R. D. Benedictis,

N. Gatti, M. Maratea, A. Micheli, A. Murano, E. Scala, L. Serafini, I. Serina, A. Umbrico, M. Vallati (Eds.), Proceedings of the 10th Italian workshop on Planning and Scheduling (IPS 2022), RCRA Incontri E Confronti (RiCeRcA 2022), and the workshop on Strategies, Prediction, Interaction, and Reasoning in Italy (SPIRIT 2022) co-located with 21st International Conference of the Italian Association for Artificial Intelligence (AIxIA 2022), November 28 - December 2, 2022, University of Udine, Udine, Italy, volume 3345 of *CEUR Workshop Proceedings*, CEUR-WS.org, 2022. URL: https://ceur-ws.org/Vol-3345/paper16_Spirit5.pdf.