# Reasoning about Real-Time and Probability on Obstruction Logic

Jean Leneutre, Vadim Malvone and James Ortiz

*LTCI, Télécom Paris, Institut Polytechnique de Paris, Palaiseau, France*

**Abstract**

This paper discusses Timed Obstruction Temporal Logic (TOTL) and Probabilistic Obstruction Temporal Logic (POTL), extensions of Obstruction Logic for systems with critical timing and probabilistic behavior. TOTL focuses on real-time system properties, while POTL handles uncertainty and stochastic events alongside temporal constraints. These logics are especially useful in cybersecurity, where both time and probability are crucial. We demonstrate their applicability through case studies in cybersecurity games based on attack graphs.

**Keywords**

Timed Systems, Dynamic games, Attack Graphs, Markov Chain, Strategic Reasoning

## 1. Introduction

Digital systems are becoming increasingly complex and challenging, especially when timing and uncertainty are critical. Multi-Agent Systems (MAS), consisting of interacting autonomous agents, are increasingly used to model complex scenarios in various domains, including cybersecurity, distributed systems, and automated control, where timing and uncertainty are prevalent [2]. In such systems, agents may cooperate or compete to achieve their goals, and their interactions are often analyzed using game theory. Game theory provides a mathematical framework for understanding strategic interactions among rational agents. In MAS, this often involves dynamic games played in arenas, where agents make decisions based on the current state and the expected actions of others. These games become even more complex when we introduce timing constraints and probabilities into the model, reflecting the real-world conditions where the timing of actions and the uncertainty of outcomes significantly influence the agents' strategies. Alternating-Time Temporal Logic (ATL) [3] and Strategy Logic (SL)[4] are formalisms that have been developed to reason about the capabilities of coalitions of players in such multi-agent systems. ATL extends traditional temporal logic by allowing the expression of properties that depend on the strategies available to different agents or groups of agents. For instance, ATL can specify whether a coalition of players can guarantee reaching a particular goal through cooperative actions, regardless of the opposition they face. Similarly, SL provides a framework to reason about the existence and effectiveness of strategies in dynamic games, where agents must consider both their objectives and the strategies of others. In these logics, the game model is typically static, meaning that while players' actions may change their positions within the arena, they do not alter the overall structure of the game itself. This approach differs from dynamic games, where the game model can change in response to the player's actions. Such dynamic games are especially relevant in areas like cybersecurity and planning, where the ability to adjust the game environment in real time to address new threats or capitalize on emerging opportunities is essential. Obstruction Logic (OL) [5] is a formalism designed to analyze games with temporal objectives in dynamic models. The games involve two players, the

*AI4CC-IPS-RCRA-SPIRIT 2024: International Workshop on Artificial Intelligence for Climate Change, 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 25-28th, 2024, Bolzano, Italy [1].*

*Corresponding author.

†These authors contributed equally.

jean.leneutre@telecom-paris.fr (J. Leneutre); vadim.malvone@telecom-paris.fr (V. Malvone); james.ortizvega@telecom-pari.fr (J. Ortiz)

0000-0003-4810-0791 (J. Leneutre); 0000-0001-6138-4229 (V. Malvone); 00000-0001-5407-963X (J. Ortiz)

Demon and the Traveler, who interact on a directed graph with edges associated with deactivation costs. The Demon attempts to obstruct the Traveler by deactivating certain edges within a budget, while the Traveler navigates the graph. The game progresses in rounds, with the Demon winning if the Traveler's path satisfies a specific temporal property. OL is particularly useful in cybersecurity, where it can model scenarios involving attackers and defenders, enabling the design of dynamic defense strategies using concepts like Attack Graphs (AG) [6] and Moving Target Defense (MTD) mechanisms. However, these logics often fall short when it comes to dealing with the complex time constraints, probabilistic behaviors, and dynamic models that are a hallmark of cybersecurity.

To address these challenges, we discuss two extensions of the obstruction logic: Timed Obstruction Temporal Logic (TOTL) and Probabilistic Obstruction Temporal Logic (POTL). TOTL builds on the foundation of temporal logic by incorporating constructs that allow precise modeling of time-dependent behavior. This makes TOTL particularly suitable for real-time systems where the timing of actions and events must be meticulously managed. POTL, on the other hand, adds probabilistic reasoning to the framework. This allows for the analysis of systems where outcomes are influenced by both deterministic and stochastic factors. By integrating probability with temporal logic, POTL allows for a more nuanced understanding of system behavior under uncertainty. This is essential in domains such as cybersecurity, where the likelihood of certain events must be carefully assessed.

Together, TOTL and POTL provide a unified framework for reasoning about systems that are both time-sensitive and probabilistically complex. This paper explores the syntax and semantics of these logics and illustrates their power and flexibility with practical examples. Through examples, we show how these logics can be applied to real-world scenarios, such as the formal verification of security protocols and the optimization of defense strategies in the face of sophisticated cyber threats. By leveraging the capabilities of TOTL and POTL, system designers and security analysts can better manage the trade-offs between time, probability, and system performance, ultimately leading to more secure and reliable digital infrastructures.

**Structure of the work.** In Section 2, we present the syntax and the semantics of our logics, called Timed Obstruction Logic (TOL) and Probabilistic Obstruction Temporal Logic (POTL). In Section 3, we present our case study in the cybersecurity context. In Section 4, we compare our approach to related work. Finally, Section 5 concludes and presents possible future directions.

## 2. Model and Logic

In this section, we define the syntax and semantics of our Probabilistic Obstruction Temporal Logic (POTL) and Timed Obstruction Temporal Logic (TOTL).

### 2.1. Syntax and Semantics of POTL

Here, we introduce the Probabilistic Obstruction Temporal Structure (POTS), the type of model that we use to verify POTL properties. Let us start with some probabilistic concepts. A probability distribution over a finite set assigns probabilities to each element such that the total sum is 1. A probability space, which consists of a sample space, is a set of possible events (a $\sigma$-algebra), and a probability measure assigns probabilities to these events.

**Definition 1 (Markov Chain).** *A Markov Chain (MC) is a pair $\mathcal{H} = (Q, \mathbf{P})$ where $Q$ is a (countable) set of states and $\mathbf{P}$: $Q \times Q \to [0, 1]$ is a transition probability function such that for each state $q \in Q$, $\Sigma_{q' \in Q} \mathbf{P}(q, q') = 1$. If $Q$ is finite, we can consider $\mathbf{P}$ to be a transition matrix.*

**Definition 2 (Probabilistic Kripke Structure (PKS)).** *A PKS over a set $\mathsf{AP}$ of atomic propositions is a tuple $\mathcal{G} = \langle Q, q_0, \mathbf{P}, \mathcal{L} \rangle$ where $(Q, \mathbf{P})$ is a MC, $q_0 \in Q$ is the initial state, and $\mathcal{L} : S \to 2^{\mathsf{AP}}$ is a labeling function assigning a set of atomic propositions to any state $q \in Q$.*

**Definition 3 (Probabilistic Obstruction Temporal Structure).** *A POTS (model for short) is given by a tuple* $\mathcal{M} = (Q, q_0, \mathbf{P}, \mathcal{L}, \mathsf{C})$ *where* $\mathcal{G} = (Q, q_0, \mathbf{P}, \mathcal{L})$ *is a PKS and* $\mathsf{C} : Q \times Q \to \mathbb{N}$ *is a function assigning to any pairs* $(q, q')$ *a natural number* $n \in \mathbb{N}$.

A path $\pi$ over $\mathcal{M}$ is a finite or infinite sequence of states $\pi = q_0, q_1, q_2, \ldots$ starting from the initial state $q_0$, where each step satisfies $\mathbf{P}(q_i, q_{i+1}) > 0$ for all $i \in \mathbb{N}$. We use $\pi_i$ to denote the $i$-th state $q_i$ in $\pi$, $\pi_{\leq i}$ for the prefix $q_0, \ldots, q_i$, and $\pi_{\geq i}$ for the suffix $q_i, q_{i+1} \ldots$. The set of all finite paths from a state $q \in Q$ in the model $\mathcal{M}$ is denoted by $\mathsf{Paths}^{+}_{\mathcal{M},q}$, and the set of all infinite paths is $\mathsf{Paths}^{*}_{\mathcal{M},q}$. A *history* $h$ is any finite path prefix, with $H$ denoting the set of all histories, and $last(h)$ representing the last state of a history $h$. Given a finite path $\hat{\pi} = q_0, q_1, \ldots, q_n$ of states, the cylinder set of $\hat{\pi}$, denoted $\mathsf{Cyl}(\hat{\pi}) = \{\pi \in \mathsf{Paths}^{*}_{\mathcal{M},q_0} \mid \hat{\pi} \in \mathsf{Prefix}(\pi)\}$, is the set of infinite paths $\pi = q_0, q_1, \cdots, q_n, \cdots$, where $\hat{\pi}$ is a prefix of $\pi$. The set of infinite paths is supposed to be equipped with the $\sigma$-algebra generated by the cylinder sets of the finite paths and the probability measure given by $\mathsf{Pr}^{q_0}_{\mathcal{M}}(\mathsf{Cyl}(\hat{\pi})) = \prod_{i=0}^{n-1} \mathbf{P}(q_i, q_{i+1})$.

**Definition 4.** *Let* $\mathcal{M}$ *be a model,* $Q$ *be states in* $\mathcal{M}$, $\mathsf{C}$ *is the function cost and* $n$ *be a natural number, a* $n$*-strategy is a function* $\mathfrak{S} : H \to 2^{Q \times Q}$ *that, given a history* $h$*, returns a subset* $T \in Q \times Q$*, such that: (i)* $T \subset E(last(h))$ *and (ii)* $(\sum_{e \in T} \mathsf{C}(e)) \leq n$*. A memoryless n-strategy is a n-strategy* $\mathfrak{S}$ *such that for all histories* $h$ *and* $h'$ *if* $last(h) = last(h')$ *then* $\mathfrak{S}(h) = \mathfrak{S}(h')$ *and* $E(last(h))$ *denotes its outgoing edges, where* $E(e) = \{e \in Q \times Q \mid e = (q, q')$ *for some* $q' \in Q$ *and* $\mathbf{P}(q, q') > 0\}$.

A path $\pi$ is compatible with an n-strategy $\mathfrak{S}$ if for all $i \geq 1$ we have that $(\pi_i, \pi_{i+1}) \notin \mathfrak{S}(\pi_{\leq i})$. The set of outcomes of an $n$-strategy $\mathfrak{S}$ and state $q$ is denoted as $\mathsf{Out}(q, \mathfrak{S})$ and it returns the set of all paths that can result from a strategy $\mathfrak{S}$ and a state $q$.

**Definition 5.** *Let* $\mathsf{AP}$ *be an at most countable set of atomic formulas (or atoms). Formulas of Probabilistic Obstruction Temporal Logic (POTL, for short) are defined by the following grammar:*

$$\varphi ::= \top \mid p \mid \neg\varphi \mid \varphi \wedge \varphi \mid \langle\!\downarrow\!\!{}^{\bowtie k}_{n}\rangle\theta$$

$$\theta ::= \mathsf{X}\,\varphi \mid \varphi\,\mathsf{U}^{\leq m}\varphi \mid \varphi\,\mathsf{U}\,\varphi \mid \varphi\,\mathsf{R}^{\leq m}\varphi \mid \varphi\,\mathsf{R}\,\varphi$$

*where* $p \in \mathsf{AP}$ *is an atomic formula,* $k \in [0, 1]$ *is a rational constant,* $n$ *(the grade) and* $m$ *are any natural number in* $\mathbb{N}$*, and* $\bowtie \in \{\leq, <, >, \geq\}$.

In this syntax, state formulas $\varphi$ are evaluated over states, and path formulas $\theta$ are evaluated over paths. Model properties are expressed as state formulas, with path formulas used only in the probabilistic path operator $\langle\!\downarrow\!\!{}^{\bowtie k}_{n}\rangle\theta$. Temporal operators like $X$ (next), $\mathsf{U}^{\leq m}$ (bounded until), $\mathsf{U}$ (until), $\mathsf{R}^{\leq m}$ (bounded release), and $\mathsf{R}$ (release), are allowed in path formulas. The parameter $n$ is the grade of the strategic operator. Boolean connectives $\bot$, $\vee$, and $\rightarrow$ are defined as usual. The formula $\langle\!\downarrow\!\!{}^{\bowtie k}_{n}\rangle\varphi$ with $\varphi$ means "there is a demonic strategy such that all paths compatible with the strategy satisfy $\varphi$ with a probability $\bowtie k$", where a demonic strategy disables arcs. POTL formulas are interpreted over POTS, and we can now define their semantics precisely.

**Definition 6.** *The satisfaction relation between a model* $\mathcal{M}$*, a state* $q$ *of* $\mathcal{M}$*, and a formula* $\varphi$ *is defined by induction on the structure of* $\varphi$*:*

- $\mathcal{M}, q \models \langle\!\downarrow\!\!{}^{\bowtie k}_{n}\rangle\theta$ *iff there is a n-strategy* $\mathfrak{S}$ *such that* $Pr^{q}_{\mathcal{M}}(\{\pi \in Out(q, \mathfrak{S}) \mid \mathcal{M}, \pi \models \theta\}) \bowtie k$.

*The satisfaction relation* $\mathcal{M}, \pi \models \varphi$ *between a model* $\mathcal{M}$*, a path* $\pi \in Paths^{*}_{\mathcal{M},q}$ *of* $\mathcal{M}$*, and path formula* $\theta$ *is defined as follows:*

- $\mathcal{M}, \pi \models \mathsf{X}\,\varphi$ *iff* $\mathcal{M}, \pi_2 \models \varphi$.
- $\mathcal{M}, \pi \models \varphi_1\,\mathsf{U}^{\leq m}\varphi_2$ *iff there is an* $0 \leq i \leq m$ *such that* $\mathcal{M}, \pi_i \models \varphi_2$ *and* $\mathcal{M}, \pi_j \models \varphi_1$ *for all* $0 \leq j < i$.
- $\mathcal{M}, \pi \models \varphi_1\,\mathsf{U}\,\varphi_2$ *iff there is an* $i \geq 0$ *such that* $\mathcal{M}, \pi_i \models \varphi_2$ *and* $\mathcal{M}, \pi_j \models \varphi_1$ *for all* $0 \leq j < i$.

- $\mathcal{M}, \pi \models \varphi_1 \, \mathsf{R}^{\leq m} \varphi_2$ *iff either* $\mathcal{M}, \pi_i \models \varphi_2$ *for all* $0 \leq i \leq m$ *or there is an* $0 \leq i \leq m$ *such that* $\mathcal{M}, \pi_i \models \varphi_1$ *and* $\mathcal{M}, \pi_j \models \varphi_2$ *for all* $0 \leq j \leq i$.
- $\mathcal{M}, \pi \models \varphi_1 \, \mathsf{R} \, \varphi_2$ *iff either* $\mathcal{M}, \pi_i \models \varphi_2$ *for all* $i \geq 0$ *or there is an* $i \geq 0$ *such that* $\mathcal{M}, \pi_i \models \varphi_1$ *and* $\mathcal{M}, \pi_j \models \varphi_2$ *for all* $0 \leq j \leq i$.

Let $\varphi$ be a formula and $\mathcal{M}$ be a model. Then $\mathsf{Sat}(\varphi, \mathcal{M})$ denotes the set of states of $\mathcal{M}$ that satisfy $\varphi$, i.e., $\mathsf{Sat}(\varphi, \mathcal{M}) = \{q \in Q \mid \mathcal{M}, q \models \varphi\}$. Two formulas $\varphi$ and $\psi$ are equivalent (denoted by $\varphi \equiv \psi$) if, for all models $\mathcal{M}$, $\mathsf{Sat}(\varphi, \mathcal{M})$ = $\mathsf{Sat}(\psi, \mathcal{M})$. The semantics of the obstruction probabilistic operator $\langle \dotplus_n{}^{\bowtie k} \rangle$ refers to the probability for the sets of paths where a path formula holds. To ensure that this is well-defined, we need to establish that the events specified by POTL path formulas are measurable. Since the set $\{\pi \in \mathsf{Out}(q, \mathfrak{S}) \mid \mathcal{M}, \pi \models \varphi\}$ for POTL path formula $\varphi$ can be considered as a countable union of cylinder sets, its measurability is ensured. This follows from the following lemma.

**Lemma 1.** *For each POTL path formula* $\varphi$ *and state* $q$ *of a model* $\mathcal{M}$*, the set* $\{\pi \in \mathit{Out}(q, \mathfrak{S}) \mid \mathcal{M}, \pi \models \varphi\}$ *is measurable.*

## 2.2. Syntax and Semantics of TOTL

Here, we present the Weighted Timed Automata (WTA), the type of model that we use to verify TOTL properties. A WTA is an extension of a TA [7] with weight/cost information at both locations and edges, and it can be used to address several interesting questions [8, 9]. In WTA (also TA) are used non-negative real-valued variables called *clocks* to represent the continuous time domain [7].

**Definition 7 (Clock constraints and invariants).** *Let X be a finite set of variables ranging over* $\mathbb{R}_{\geq 0}$*, called clocks. The set* $\Phi(X)$ *of clock constraints over the set of clocks X is given by the following grammar:*

$$\phi := true \mid x \sim c \mid \phi_1 \ \wedge \ \phi_2$$

*where* $x \in X$*,* $c \in \mathbb{N}$*, and* $\sim \in \ \{<, \ >, \ \leq, \ \geq, \ =\}$.

Clock invariants $\Delta(X)$ are clock constraints in which $\sim \in \{<, \leq\}$.

**Definition 8 (Clock valuations).** *Given a finite set of clocks X, a clock valuation function,* $\nu : X \to \mathbb{R}_{\geq 0}$ *assigning to each clock* $x \in X$ *a non-negative value* $\nu(x)$*. We denote* $\mathbb{R}^X_{\geq 0}$ *the set of all valuations. For a clock valuation* $\nu \in \mathbb{R}^X_{\geq 0}$ *and a time value* $d \in \mathbb{R}_{\geq 0}$*,* $\nu + d$ *is the valuation satisfied by* $(\nu + d)(x) = \nu(x) + d$ *for each* $x \in X$*. Given a clock subset* $Y \subseteq X$*, we denote* $\nu[Y \leftarrow 0]$ *the valuation defined as follows:* $\nu[Y \leftarrow 0](x) = 0$ *if* $x \in Y$ *and* $\nu[Y \leftarrow 0](x) = \nu(x)$ *otherwise.*

**Definition 9 (Clock valuations).** *Given a finite set of clocks X, a clock valuation function,* $\nu : X \to \mathbb{R}_{\geq 0}$ *assigning to each clock* $x \in X$ *a non-negative value* $\nu(x)$*. We denote* $\mathbb{R}^X_{\geq 0}$ *the set of all valuations. For a clock valuation* $\nu \in \mathbb{R}^X_{\geq 0}$ *and a time value* $d \in \mathbb{R}_{\geq 0}$*,* $\nu + d$ *is the valuation satisfied by* $(\nu + d)(x) = \nu(x) + d$ *for each* $x \in X$*. Given a clock subset* $Y \subseteq X$*, we denote* $\nu[Y \leftarrow 0]$ *the valuation defined as follows:* $\nu[Y \leftarrow 0](x) = 0$ *if* $x \in Y$ *and* $\nu[Y \leftarrow 0](x) = \nu(x)$ *otherwise.*

**Definition 10 (Weighted Timed Automata (WTA)).** *Let* $X$ *be a finite set of clocks and* $\mathsf{AP}$ *a finite set of atoms. A WTA is a tuple* $\mathcal{A}$ = $(L, l_0, X, \Sigma, T, I, C, \mathcal{L}, F)$*, where:* $L$ *is a finite set of locations,* $l_0 \in L$ *is an initial location,* $X$ *is a finite set of clocks,* $\Sigma$ *is a finite set of actions,* $T \subseteq L \times \Sigma \times \Phi(X) \times 2^X \times L$ *is a finite set of edges (or transitions),* $I : L \to \Delta(X)$ *is a function that associates to each location a clock invariant,* $C$*:* $T \to \mathbb{N}_{\geq 0}$ *is a function that labels the elements of* $T$*,* $\mathcal{L} : L \to 2^{\mathsf{AP}}$ *is a labeling function for the locations,* $F \subseteq L$ *is a set of goal locations.*

In WTA, costs are explicitly defined in its syntax, however, they do not influence the discrete behavior of the system. Since there is no cost constraint, the semantics of a WTA is similar to that of a TA. It is thus given as a Timed Transition System (TTS). In a $\mathsf{TTS}(\mathcal{A})$ = $(Q, q_0, \Sigma_\Delta, E, C, \mathcal{L}, Q_F)$, the states

$Q$ are pairs of locations and clock valuations, with initial state $q_0$=$(l_0, \nu_0)$, where all clocks $x \in X$ are initially 0, $\Sigma_\Delta = \Sigma \uplus \mathbb{R}_{\geq 0}$, the transition relation $E$ includes discrete transitions $(l, \nu) \xrightarrow{a}_c (l', \nu')$ based on satisfying guards and clock resets, and delay transitions $(l, \nu) \xrightarrow{d} (l, \nu + d)$ for valid time elapses, the labeling function $\mathcal{L}$ assigns atomic propositions to states based on their location and clock valuations and $Q_F \subseteq F \times \mathbb{R}^X_{\geq 0}$.

A path $\rho$ in $\mathsf{TTS}(\mathcal{A})$ is an infinite sequence of consecutive delays and discrete transitions. A finite path fragment of $\mathcal{A}$ is a run in $\mathsf{TTS}(\mathcal{A})$ starting from the initial state $q_0 = (l_0, \nu_0)$, with delay and discrete transitions alternating along the path: $\rho = q_0 \xrightarrow{d_0} q'_1 \xrightarrow[w_0]{a_0} q_1 \xrightarrow{d_1} q'_2 \xrightarrow[w_1]{a_1} q_2 \dots q_{n-1} \xrightarrow{d_{n-1}} q'_n \xrightarrow[w_n]{a_n} q_n \dots$. We write $\rho_i$ to denote the $i$-th element $q_i = (l_i, \nu_i)$ of $\rho$, $\rho_{\leq i}$ to denote the prefix $q_0, \dots, q_i$ of $\rho$, and $\rho_{\geq i}$ to denote the suffix $q_i, q_{i+1} \dots$ of $\rho$. A *history* is any finite prefix of some path. We use $H$ to denote the set of histories. Let $\mathcal{A}$ be a WTA. Here, we will use $T'$ to indicate the set of deactivated edges in $\mathcal{A}$. We will also use $\mathsf{ind}(T')$ to indicate the set of edges induced by the deactivated edges in $T'$ in the $\mathsf{TTS}(\mathcal{A})$.

**Definition 11.** *Let $\mathcal{A}$ be a WTA and $n$ be a natural number. Given a model TTS($\mathcal{A}$), a $n$-strategy is a function $\mathfrak{S} : H \to 2^T$ that, given a history $h$, returns a subset $T'$ such that: (i) $T' \subset T(last(h))$, (ii) ind$(T') \subset E(last(h))$, and (iii) $(\sum_{t \in T'} \mathsf{C}(t)) \leq n$. 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 $\rho$ is compatible with a $n$-strategy if for all $i \geq 1$, $(\rho_i, \sigma, \rho_{i+1}) \notin \mathfrak{S}(\rho_{\leq i})$, where $\sigma \in \Sigma$.

Given a state $q = (l, \nu)$ and a $n$-strategy $\mathfrak{S}$, $Out(q, \mathfrak{S})$ refers to the set of pathways starting from $q$ that are consistent with $\mathfrak{S}$.

**Definition 12.** *Let $\mathcal{A}$ be a WTA, $\mathsf{AP}$ a set of atomic propositions (or atoms), a set $X$ of clocks of $\mathcal{A}$, and $J$ a non-empty set of clocks of the formula, where $X \cap J = \emptyset$. Formulas of Timed Obstruction Temporal Logic (TOTL) are defined by the following grammar:*

$$\varphi ::= \top \mid p \mid \neg\varphi \mid \varphi_1 \wedge \varphi_2 \mid \phi \mid \langle\!\!+_n\rangle(\varphi_1 \,\mathsf{U}\, \varphi_2) \mid \langle\!\!+_n\rangle(\varphi_1 \,\mathsf{R}\, \varphi_2) \mid j.\varphi$$

*where $p \in \mathsf{AP}$ is an atomic formula, $j \in J$, $n \in \mathbb{N}_{\geq 0}$ represents* the grade *of the strategic operator, and $\phi \in \Phi(X \cup J)$.*

It is possible to compare a formula clock with an automata clock using clock constraints $\phi$, which apply to both. Boolean connectives $\bot$, $\vee$, and $\rightarrow$ are defined as usual. In the formula $j.\varphi$, the clock $j$ is called freeze identifier, which means $j$ starts at 0 in the current state, and $\varphi$ must hold from that point. This can express timing requirements like punctuality or bounded response. For example, $j.\langle\!\!+_n\rangle((\varphi_1 \wedge j \leq 7)\,\mathsf{U}\,\varphi_2)$ means there is a demonic strategy ensuring $\varphi_1$ holds until $\varphi_2$ becomes valid within 7 time units. The intuitive meaning of a formula $\langle\!\!+\rangle\varphi$ with $\varphi$ timed temporal formula is: there is a demonic strategy such that all paths of the TTS that are compatible with the strategy satisfy $\varphi$. Unlike OL, TOTL does not use the next operator because time is continuous, and there is no unique "next" time. However, TOTL allows timing constraints, called timed temporal formulas, which are interpreted over TTS. The semantics of TOTL formulas are now defined precisely.

**Definition 13 (TOTL Semantics).** *Let $\mathcal{A}$ be a WTA, a set $X$ of clocks of $\mathcal{A}$, $J$ a non-empty set of clocks of the formula, $p \in \mathsf{AP}$, $\phi \in \Phi(X \cup J)$, and $\mathcal{M}$ = TTS($\mathcal{A}$). An extended state over $Q$ is a triple $(l, \nu, \mu)$, where $q = (l, \nu) \in Q$ is a WTS state and $\mu$ a valuation for the formula clocks in $J$. The satisfaction relation between a TTS $\mathcal{M}$, TOTL formulas $\varphi$ and $\psi$, and an extended state $q_\mu = (l, \nu, \mu)$[1] of the formula, is given inductively as follows (Boolean operators are omitted because they are defined as usual):*

- $\mathcal{M}, q \models \phi$ *iff* $\nu \models \phi$.
- $\mathcal{M}, q \models \langle\!\!+_n\rangle(\varphi\,\mathsf{U}\,\psi)$ *iff there is a $n$-strategy $\mathfrak{S}$ such that for all $\rho \in Out(q, \mathfrak{S})$ there is a $j \in \mathbb{N}$ such that $\mathcal{M}, \rho_j \models \psi$ and for all $0 \leq k < j$, $\mathcal{M}, \rho_k \models \varphi$.*

[1]To facilitate reading, from this point onward we will use only the symbol $q$ for an extended state.

- $\mathcal{M}, q \models \langle\!\langle n \rangle\!\rangle(\varphi \,\mathsf{R}\, \psi)$ *iff there is a* $n$*-strategy* $\mathfrak{S}$ *such that for all* $\rho \in Out(q, \mathfrak{S})$ *we have that either* $\mathcal{M}, \rho_i \models \psi$ *for all* $i \in \mathbb{N}$ *or there is a* $k \in \mathbb{N}$ *such that* $\mathcal{M}, \rho_k \models \varphi$ *and* $\mathcal{M}, \rho_i \models \psi$ *for all* $0 \leq i \leq k$.
- $\mathcal{M}, q \models j.\varphi$ *iff* $\mathcal{M}, (l, \nu[j \leftarrow 0], \mu) \models \varphi$.

Two formulas $\varphi$ and $\psi$ are semantically equivalent (denoted by $\varphi \equiv \psi$) iff for any model $\mathcal{M}$ and extended state $s$ of $\mathcal{M}$, $\mathcal{M}, q \models \varphi$ iff $\mathcal{M}, q \models \psi$. The relationship between WTA and TTS is defined as follows.

**Proposition 1.** *Let* $\mathcal{A}$ *be a WTA and* $\varphi \in$ *TOTL, then* $\mathcal{A} \models \varphi$ *iff* $TTS(\mathcal{A}) \models \varphi$.

Let $\varphi$ be a formula, the set of extended states satisfying $\varphi$ is independent of the valuation $\mu$ for the formula clocks. Thus, for any state $q = (l, \nu)$ in a TTS and valuations $\mu$, $\mu'$ for the formula clocks, we can get that $\mathcal{M}, (l, \nu, \mu) \models \varphi$ iff $\mathcal{M}, (l, \nu, \mu') \models \varphi$. Therefore, when $\varphi$ is closed, it makes sense to talk about a state $q$ that satisfies $\varphi$. Let $\varphi$ be any formula, $(X \cup J)$ a set of clocks (formula and automaton) and $\mathcal{A}$ be a WTA, then $\mathrm{Sat}(\varphi)$ denotes the set of extended states of $\mathcal{M} = \mathrm{TTS}(\mathcal{A})$ verifying, $\varphi$, i.e., $\mathrm{Sat}(\varphi) = \{q \in Q \mid \mathcal{M}, q \models \varphi\}$.

## 3. Case Study

In this section, we consider two case study related to cybersecurity scenario.

### 3.1. Probabilistic Scenario

Probability theory is well-suited for cybersecurity risk analysis because it provides a framework for understanding and quantifying uncertainty. To illustrate this, we will consider the following general cybersecurity scenario. Let $\mathcal{G}$ be an AG and we want to check if there are MTD response strategies that will satisfy certain security goals.

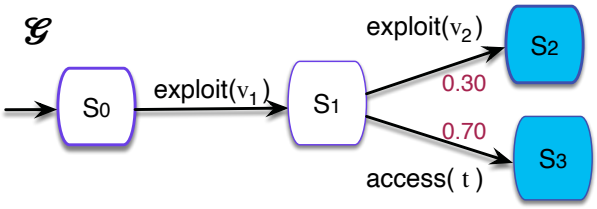


**Figure 1:** Example of an AG $\mathcal{G}$ from [10].

Consider the AG in Fig. 1 with four states: $S_0$, $S_1$, $S_2$, and $S_3$. Each state represents a state of the attacker in the system. If the attacker is in $S_0$ or $S_1$, he can perform one or two of the following actions: exploit vulnerability $v_1$, exploit vulnerability $v_2$, and access device $t$. If the attacker succeeds in exploiting $v_1$, he will transition to state $S_1$. Here, we assume that depending on the attacker's preferences, there are 70% chance that the attacker will attempt to access equipment $t$ and a 30% chance that he will attempt to exploit $v_2$. In Table 1, there are the three possible actions the attacker

| Action | Countermeasure | Cost | Efficiency |
|---|---|---|---|
| exploit($v_1$) | $c_1$ | 5 | 47.5% |
| access($t$) | $c_2$ | 1 | 22.5% |
| exploit($v_2$) | $c_3$ | 3 | 24.7% |

**Table 1**
Actions and Attack countermeasure

can deploy, with their respective countermeasures, cost, and effectiveness. Let Fig. 2 depict the POTS $\mathcal{M}$, constructed using the information from the attack graph presented in [10]. Notice that, in contrast

to [10], here we remove the actions because we do not have any actions in our POTS model. Therefore, the probabilities present in each state of the model are divided by the number of outgoing actions of that state. In Fig. 2 the yellow line (do nothing), indicates that no countermeasure will be deployed. The red lines ($c_1$ in Table 1), refer to a defensive countermeasure aimed at protecting the system against the attack attempt. However, $c_1$ has an efficiency of 47.5%. Therefore, an attacker attempting to exploit($v_1$) has a 5% chance of success. The violet lines ($c_2$) are a defensive countermeasure against accessing equipment $t$ and have an efficiency of 22.5%. The orange lines ($c_3$) are a defensive countermeasure against exploiting vulnerability $v_2$ and have an efficiency of 24.7%. Finally, green lines refer to the deployment of countermeasures $c_2$ and $c_3$ at the same time. Let us take the case where the defender chooses to deploy the countermeasure $c_3$ (orange lines) in state $S_1$, the attacker can either succeed or fail in his attack attempt. The efficiency of $c_3$ is 24.7%. Therefore, the probability that the attacker fails in his attack attempt is 0.07425 (exploit($v_2$) $\times$ efficiency($c_3$)). Otherwise, the probability of success is of 0.00075.

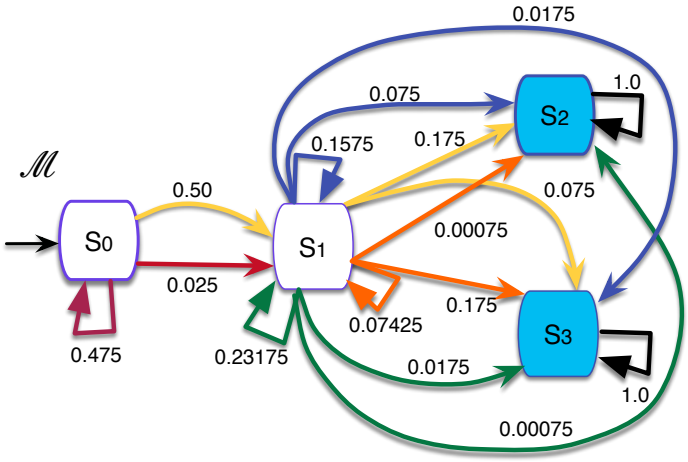


**Figure 2:** The POTS $\mathcal{M}$ from $\mathcal{G}$.

Let $r_2$ and $r_3$ be the atomic propositions for the states, $S_2$ and $S_3$. We can express, via POTL formulas, the following security objective:

- *There is a defender strategy with a cost* 4 *such that the attacker reaches the state satisfying* $r_2$ *or the state satisfying* $r_3$ *with a probability less than a given threshold* 0.1. *The following POTL formula captures the objective:* $\varphi_1 := \langle\!\!\dashv\!\!\!\!+_4^{<0.1}\rangle \mathsf{F}\,(r_2 \vee r_3)$.
- *There exists a defender strategy with cost 5 such that the probability that the attacker reaches state satisfying* $r_3$ *is less than 0.2.* The following POTL formula captures the objective: $\varphi_1 := \langle\!\!\dashv\!\!\!\!+_5^{<0.2}\rangle \mathsf{F}\, r_3$.

### 3.2. Timed Scenario

Based on the concepts of AG presented in Section 1, we want to determine if there are MTD strategies that can satisfy specific security objectives. To do this, we assume: (1) The defender always knows the attacker's current state in the AG. (2) At any moment, there is a unique current state for the attacker in the AG. (3) When the attacker's state is detected, the defender can activate one or more MTDs to temporarily remove outgoing edges from that state. (4) The total cost of deactivated edges is below a given threshold. (5) If the attacker launches an attack from its current state and the corresponding edge hasn't been removed, the attack succeeds, moving the attacker to the next state. (6) if the corresponding edge has been removed, the attack fails, and the attacker remains in the current state. In the model in Fig. 3 assume that reaching states $s_1$, $s_3$, or $s_5$ gives the attacker root privileges on a critical server $s$. In addition, if the attacker completes attack steps $a_6$ or $a_7$ (reaching state $s_5$), then the defender will obtain information on the identity of the attacker. Let $a$ be an atomic proposition that expresses 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 the server $s$. We can express, via TOL formulas, the following security objectives:

- *The attacker will never be able to obtain root privileges on server s unless the defender can obtain information about his identity within* 3 *time units*: that is, either we want the attacker to never reach a state satisfying $r_s$ *or* if the attacker reaches such a state, the defender wants to be able to identify it within 3 time units ($a$). By using $t_1$ as a variable, the following TOL formula captures the objective: $\varphi_1 := j.\langle\!\langle t_1 \rangle\!\rangle \mathsf{G}\,(r_s \vee (r_s \rightarrow \langle\!\langle t_1 \rangle\!\rangle \mathsf{F}(\mathsf{j} \leq 3 \wedge a)))$.
- *While the defender has not obtained information about the attacker identity within* 5 *time units, the attacker has not root privilege on the server s*: that is, we want $r_s$ to be false *until* we have identified the attacker ($a$) within 5 time units, if such an identification ever happens. Thus, by using $t_2$ as a variable for a given threshold, we can write our objective by using the until connective: $\varphi_2 := j.\langle\!\langle t_2 \rangle\!\rangle(\neg r_s \wedge j \leq 5\,\mathsf{U}\,a)$.

Suppose that $t_1$ and $t_2$ are respectively 3 and 4. Let $\mathcal{M} = \mathsf{TTS}(\mathcal{A})$, we have that $\mathcal{M}, s_0 \models \varphi_1 \wedge \varphi_2$. To satisfy $\varphi_1$ 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 $\mathcal{M}$. Remark that for any path $\pi \in Out(s_0, \mathfrak{S}_1)$ and any $i \in \mathbb{N}$ we have that $\mathcal{M}, \pi_i \models r_s$ iff $\pi_i \in \{s_1, s_3, s_5\}$. Thus, we must establish that $\mathcal{M}$ satisfies $\langle\!\langle 3 \rangle\!\rangle \mathsf{F}(j \leq 3 \wedge a)$ on $s_1$ (resp. $s_3$ and $s_5$). To do so, we remark that $Out(s_1, \mathfrak{S}_1)$ (resp. $Out(s_3, \mathfrak{S}_1)$ and $Out(s_5, \mathfrak{S}_1)$) only contains the path $s_1, s_3, s_5^\omega$ (resp. $s_3, s_5^\omega$ and $s_5^\omega$) and that $\mathcal{M}, s_5 \models a$. Thus, we have obtained that there is a strategy (i.e. $\mathfrak{S}_1$) such that for all $\pi \in Out(s_0, \mathfrak{S}_1)$ and all $i \in \mathbb{N}$ either $\mathcal{M}, \pi_i \models \neg r_s$ or if $\mathcal{M}, \pi_i \models r_s$ then there is a strategy ($\mathfrak{S}_1$ itself) such that $\mathcal{M}, \rho_j \models a$ for some $j \geq 1$ and for all $\rho \in Out(\pi_i, \mathfrak{S}_1)$, as we wanted. Remark that if $t_1 < 3$ then it is impossible to satisfy $\varphi_1$ in $\mathcal{M}$ at $s_0$. For the specification $\varphi_2 = j.\langle\!\langle 4 \rangle\!\rangle(\neg r_s \wedge j \leq 5\,\mathsf{U}\,a)$, 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.

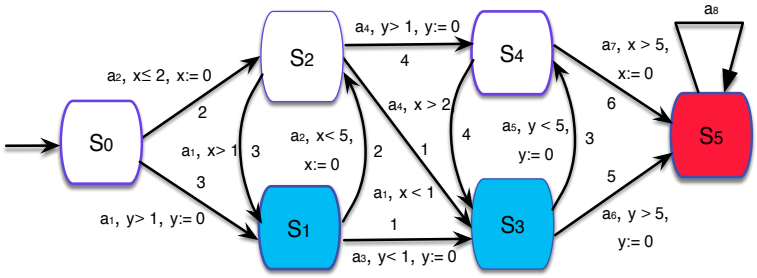


**Figure 3:** A WTA from [5] where states $s_1$, $s_3$ and $s_5$ represent the goals of the attacker and the blue nodes satisfy $r_s$, the red node satisfies both $a$ and $r_s$, and the white ones satisfy neither $r_s$ nor $a$.

## 4. Related Work

Many papers have focused on the strategic capabilities of agents playing within dynamic game models. In this section, we compare our approach with some of these papers. Previous research on sabotage games by van Benthem led to Sabotage Modal Logic (SML) for graph reachability problems, with a PSPACE-complete model checking problem [11]. Unlike sabotage games, where only one edge can be removed at a time, our approach allows temporary deactivation of edge subsets, similar to Subset Sabotage Modal Logic (SSML) [12], which lacks temporal operators and cost considerations. Our Timed Obstruction Logic (TOL) extends Obstruction Logic (OL) [5] by incorporating real-time elements, setting it apart from other strategic logics like ATL [3] and SCTL [13], which do not account for edge costs, real-time, or dynamic models. Timed Game Automata (TGA) [8] allow players to choose transitions and wait times, with extensions like TATL [14] and STCTL [13] incorporating timing requirements. However, these models do not address dynamic changes. Probabilistic logics like PSL [15], PATL, and

PATL* [16] extend ATL to handle stochastic games and probabilistic strategies. Further studies [17] explore probabilistic $\mu$-calculus and strategies under incomplete information [18]. However, these logics do not combine probabilistic reasoning with dynamic models.

## 5. Conclusion

Timed Obstruction Temporal Logic (TOTL) and Probabilistic Obstruction Temporal Logic (POTL) provide powerful frameworks for reasoning about systems where timing and probability are critical factors. By extending traditional temporal logics to account for both timed constraints and probabilistic behaviors, TOTL and POTL allow for the specification and verification of complex properties in systems like smart grids, where security, reliability, and performance are paramount. In our case study, we demonstrated how TOTL and POTL could be applied to analyze and enhance the security of a smart grid under cyber-attack. The ability to model both time-sensitive and probabilistic aspects of defense strategies enables system designers and security analysts to develop more robust and cost-effective solutions. There are several directions we would like to explore for future work. First of all, extending TOTL and POTL to support reasoning about multi-agent systems, where multiple defenders and attackers interact, would provide a more comprehensive framework for analyzing complex, distributed environments. Additionally, integrating TOTL and POTL with machine learning techniques could lead to more adaptive and intelligent defense strategies. For example, reinforcement learning could be used to optimize the selection of defense strategies based on real-time feedback from the system. Finally, we would like to implement these two logics in the VITAMIN tool [19]. This tool supports a variety of specifications, including Obstruction Logic (OL) [12] and Obstruction ATL (OATL) [20], like Alternating-time Temporal Logic (ATL) [3], ATL with Fuzzy functions (ATLF) [21], Natural ATL (NatATL) [22], Natural SL (NatSL) [23], Resource-Bounded ATL (RB-ATL) [24, 25], Resource Action-based Bounded ATL [26], and Capacity ATL (CapATL) [27].

## References


[1] D. Aineto, R. De Benedictis, M. Maratea, M. Mittelmann, G. Monaco, E. Scala, L. Serafini, I. Serina, F. Spegni, E. Tosello, A. Umbrico, M. Vallati (Eds.), Proceedings of the International Workshop on Artificial Intelligence for Climate Change, the Italian workshop on Planning and Scheduling, the RCRA Workshop on Experimental evaluation of algorithms for solving problems with combinatorial explosion, and the Workshop on Strategies, Prediction, Interaction, and Reasoning in Italy (AI4CC-IPS-RCRA-SPIRIT 2024), co-located with 23rd International Conference of the Italian Association for Artificial Intelligence (AIxIA 2024), CEUR Workshop Proceedings, CEUR-WS.org, 2024.

[2] A. Lomuscio, H. Qu, 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), 2009.

[3] R. Alur, T. Henzinger, O. Kupferman, Alternating-time temporal logic, J. ACM 49 (2002) 672–713.

[4] F. Mogavero, A. Murano, G. Perelli, M. Y. Vardi, Reasoning about strategies: On the model-checking problem, ACM Transactions in Computational Logic (2014). URL: http://doi.acm.org/10.1145/2631917. doi:10.1145/2631917.

[5] D. Catta, J. Leneutre, V. Malvone, Obstruction logic: A strategic temporal logic to reason about dynamic game models, in: ECAI 2023 - 26th European Conference on Artificial Intelligence, 2023. URL: https://doi.org/10.3233/FAIA230292. doi:10.3233/FAIA230292.

[6] K. Durkota, V. Lisý, B. Bosanský, C. Kiekintveld, Optimal network security hardening using attack graph games, in: Q. Yang, M. J. Wooldridge (Eds.), Proceedings of the Twenty-Fourth International Joint Conference on Artificial Intelligence, IJCAI 2015, AAAI Press, 2015, pp. 526–532. URL: http://ijcai.org/Abstract/15/080.

[7] R. Alur, D. Dill, A theory of timed automata, Theoretical computer science 126 (1994) 183–235.

[8] P. Bouyer, U. Fahrenberg, K. G. Larsen, N. Markey, Quantitative analysis of real-time systems using priced timed automata, Communications of the ACM (2011). URL: http://www.lsv.fr/Publis/PAPERS/PDF/BFLM-cacm11.pdf. doi:10.1145/1995376.1995396.

[9] R. Alur, S. La Torre, G. J. Pappas, Optimal paths in weighted timed automata, in: Computation and Control, 2001, pp. 49–62.

[10] Z. Ismail, Optimal defense strategies to improve the security and resilience of Smart Grids, Theses, Télécom ParisTech, 2016. URL: https://pastel.hal.science/tel-03752359.

[11] C. Löding, P. Rohde, Model checking and satisfiability for sabotage modal logic, in: FST TCS 2003: Foundations of Software Technology and Theoretical Computer Science, 2003. URL: https://doi.org/10.1007/978-3-540-24597-1_26. doi:10.1007/978-3-540-24597-1_26.

[12] 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:10.3233/IA-221080.

[13] J. Arias, W. Jamroga, W. Penczek, L. Petrucci, T. Sidoruk, Strategic (timed) computation tree logic, (2023). arXiv:2302.13405.

[14] T. A. Henzinger, V. S. Prabhu, Timed alternating-time temporal logic, in: 4th International Conferences on Formal Modelling and Analysis of Timed Systems (FORMATS'06), 2006.

[15] B. Aminof, M. Kwiatkowska, B. Maubert, A. Murano, S. Rubin, Probabilistic strategy logic, in: S. Kraus (Ed.), Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI 2019, Macao, China, August 10-16, 2019, 2019, pp. 32–38. URL: https://doi.org/10.24963/ijcai.2019/5. doi:10.24963/IJCAI.2019/5.

[16] X. Huang, C. Luo, A logic of probabilistic knowledge and strategy, in: Proceedings of the 2013 International Conference on Autonomous Agents and Multi-Agent Systems, AAMAS '13, International Foundation for Autonomous Agents and Multiagent Systems, Richland, SC, 2013.

[17] F. Song, Y. Zhang, T. Chen, Y. Tang, Z. Xu, Probabilistic alternating-time $\mu$-calculus, in: Proceedings of the Thirty-Third AAAI Conference on Artificial Intelligence and Thirty-First Innovative Applications of Artificial Intelligence Conference and Ninth AAAI Symposium on Educational Advances in Artificial Intelligence, AAAI'19/IAAI'19/EAAI'19, 2019. URL: https://doi.org/10.1609/aaai.v33i01.33016179. doi:10.1609/aaai.v33i01.33016179.

[18] F. Belardinelli, W. Jamroga, M. Mittelmann, A. Murano, Strategic abilities of forgetful agents in stochastic environments, in: P. Marquis, T. C. Son, G. Kern-Isberner (Eds.), Proceedings of the 20th International Conference on Principles of Knowledge Representation and Reasoning, KR 2023, 2023, pp. 726–731. URL: https://doi.org/10.24963/kr.2023/71. doi:10.24963/KR.2023/71.

[19] A. Ferrando, V. Malvone, VITAMIN: A compositional framework for model checking of multi-agent systems, CoRR abs/2403.02170 (2024). URL: https://doi.org/10.48550/arXiv.2403.02170. doi:10.48550/ARXIV.2403.02170. arXiv:2403.02170.

[20] D. Catta, J. Leneutre, V. Malvone, A. Murano, Obstruction alternating-time temporal logic: A strategic logic to reason about dynamic models, in: M. Dastani, J. S. Sichman, N. Alechina, V. Dignum (Eds.), Proceedings of the 23rd International Conference on Autonomous Agents and Multiagent Systems, AAMAS 2024, Auckland, New Zealand, May 6-10, 2024, International Foundation for Autonomous Agents and Multiagent Systems / ACM, 2024, pp. 271–280. URL: https://dl.acm.org/doi/10.5555/3635637.3662875. doi:10.5555/3635637.3662875.

[21] A. Ferrando, G. Luongo, V. Malvone, A. Murano, Theory and practice of quantitative atl, in: R. Arisaka, V. S. Anguix, S. Stein, R. Aydogan, L. van der Torre, T. Ito (Eds.), PRIMA 2024: Principles and Practice of Multi-Agent Systems - 25th International Conference, Kyoto, Japan, November 18-24, 2024, Proceedings, volume to appear of *Lecture Notes in Computer Science*, Springer, 2024.

[22] W. Jamroga, V. Malvone, A. Murano, Natural strategic ability, Artif. Intell. 277 (2019). URL: https://doi.org/10.1016/j.artint.2019.103170. doi:10.1016/j.artint.2019.103170.

[23] F. Belardinelli, W. Jamroga, V. Malvone, M. Mittelmann, A. Murano, L. Perrussel, Reasoning about human-friendly strategies in repeated keyword auctions, in: AAMAS 2022, 2022, pp. 62–71.

[24] H. N. Nguyen, N. Alechina, B. Logan, A. Rakib, Alternating-time temporal logic with resource bounds, J. Log. Comput. 28 (2018) 631–663.

[25] A. Ferrando, V. Malvone, Hands-on VITAMIN: A compositional tool for model checking of

multi-agent systems, in: M. Alderighi, M. Baldoni, C. Baroglio, R. Micalizio, S. Tedeschi (Eds.), Proceedings of the 25th Workshop "From Objects to Agents", Bard (Aosta), Italy, July 8-10, 2024, volume 3735 of *CEUR Workshop Proceedings*, CEUR-WS.org, 2024, pp. 148–160. URL: https://ceur-ws.org/Vol-3735/paper_12.pdf.

[26] D. Catta, A. Ferrando, V. Malvone, Resource action-based bounded atl: a new logic for mas to express a cost over the actions, in: R. Arisaka, V. S. Anguix, S. Stein, R. Aydogan, L. van der Torre, T. Ito (Eds.), PRIMA 2024: Principles and Practice of Multi-Agent Systems - 25th International Conference, Kyoto, Japan, November 18-24, 2024, Proceedings, volume to appear of *Lecture Notes in Computer Science*, Springer, 2024.

[27] G. Ballot, V. Malvone, J. Leneutre, Y. Laarouchi, Strategic reasoning under capacity-constrained agents, in: M. Dastani, J. S. Sichman, N. Alechina, V. Dignum (Eds.), Proceedings of the 23rd International Conference on Autonomous Agents and Multiagent Systems, AAMAS 2024, Auckland, New Zealand, May 6-10, 2024, International Foundation for Autonomous Agents and Multiagent Systems / ACM, 2024, pp. 123–131. URL: https://dl.acm.org/doi/10.5555/3635637.3662859. doi:10.5555/3635637.3662859.