When Natural Strategies Meet Fuzziness and Resource-Bounded Actions

Marco Aruta[1], **Francesco Improta**[1], **Vadim Malvone**[2], **Aniello Murano**[1]

[1]University of Naples Federico II, Naples, Italy
[2]LTCI, Télécom Paris, Institut Polytechnique de Paris, Palaiseau, France
marco.aruta@unina.it, francesco.improta@unina.it, aniello.murano@unina.it, vadim.malvone@telecom-paris.fr

Abstract

In formal strategic reasoning for Multi-Agent Systems (MAS), agents are typically assumed to (i) employ arbitrarily complex strategies, (ii) execute each move at zero cost, and (iii) operate over fully crisp game structures. These idealized assumptions stand in stark contrast with human decision-making in real-world environments. The *natural strategies* framework along with some of its recent variants, partially addresses this gap by restricting strategies to concise rules guarded by regular expressions. Yet, it still overlook both the cost of each action and the uncertainty that often characterizes human perception of facts over the time. In this work, we introduce HumanATL$[\mathcal{F}]$, a logic that builds upon natural strategies employing both fuzzy semantics and resource-bound actions: each action carries a real-valued cost drawn from a non-refillable budget, and atomic conditions and goals have degrees in [0,1]. We give a formal syntax and semantics, and prove that model checking is in P when both the strategy complexity k and resource budget b are fixed, NP-complete if just one strategic operator over Boolean objectives is allowed, and Δ_2^P-complete when k and b vary. Moreover, we show that recall-based strategies can be decided in PSPACE. We implement our algorithms in VITAMIN, an open source model-checking tool for MAS and validate them on an adversarial resource-aware drone rescue scenario.

Code — https://github.com/MarcoAruta/HumanATLFTool
Extended version — https://github.com/MarcoAruta/Paper

Introduction

Formal strategic reasoning in Multi-Agent Systems (MAS) plays a central role in verifying and synthesizing behaviors in distributed and adversarial settings. A milestone in this area is *Alternating-time Temporal Logic* (ATL), introduced by Alur, Henzinger, and Kupferman (Alur, Henzinger, and Kupferman 2002), which provides a formal framework to express and verify the strategic abilities of coalitions of agents. ATL and its extensions have greatly advanced the formal aspects of strategic reasoning, enabling model checking techniques that scale to complex MAS (Ågotnes 2006; Schnoor 2009; Knapik et al. 2019; Jamroga et al. 2025). Over the years, theoretical developments and practical implementations have further refined these foundations, improving expressiveness, decidability, and verification algorithms that bridge logic, game theory, and automated reasoning (Chatterjee, Henzinger, and Piterman 2010; Mogavero et al. 2014).

Although ATL comes with a powerful strategic-reasoning capability for MAS, it assumes idealized agents with unlimited reasoning capacity, making it unsuitable for modeling the bounded rationality typical of human decision-making. This has spurred the community to look for a new strategic reasoning framework that better represents (and allows to reproduce) bounded, human-like reasoning. This has led to *natural strategies* (Jamroga, Malvone, and Murano 2019), which allows to define human-reproducible strategies represented by means of rules guarded by regular expressions.

While natural strategies allow to capture the important aspect of bounded rationality, it measures only the combinatorial cost of evaluating propositional conditions; it does not take into account that in real-world strategic reasoning settings, humans (and in general resource-limited AI agents) cannot maintain arbitrarily complex strategies. In realistic scenarios, each strategy action consumes tangible resources, whether cognitive effort or physical energy. As an example, consider a fleet of delivery drones operating in a smart city: every "ascend" or "move" maneuver depletes a fixed portion of the battery. A path that seems admissible under Boolean or regular-expression guards may prove infeasible if the sum of individual action costs exceeds the drone's remaining charge. At the same time, decision makers frequently employ graded judgments rather than strict true–false distinctions: goals such as "maintain safe distance" or "minimize detection risk" are naturally expressed in terms of satisfaction levels. By endowing a strategic logic with *fuzzy* semantics (i.e. the truth degrees in $[0, 1]$) one can capture these intermediate satisfaction levels and reason about partial achievements of objectives. We contend that a unified treatment of human-feasible strategy complexity, action-level resource consumption, and fuzzy outcome evaluation delivers a more faithful model of strategic reasoning in human-like agents.

Our Contribution. We introduce HumanATL$[\mathcal{F}]$, a Human-like logic that extends ATL on three key quanti-

tative dimensions. First, strategy descriptions are human-readable natural strategies: each rule is a simple guarded-action pair whose syntactic complexity is capped by a parameter k. Second, every action draws from a finite, non-refillable budget b, ensuring plans respect explicit resource constraints. Third, objectives are evaluated under fuzzy semantics, assigning truth degrees in $[0, 1]$ to capture partial fulfillment of the aforementioned goals. On this foundation, we develop the full syntax and fuzzy, cost-aware semantics of HumanATL$[\mathcal{F}]$ and then analyze the computational complexity of the model-checking problem. The latter established three key results: the verification is in P when both the strategy complexity bound k and the available budget b are fixed; following that, even a single strategic operator over Boolean objectives induces NP-completeness; and finally, that model checking becomes Δ_2^P-complete when k and b are treated as part of the input. In the case of recall-based strategies, we show that model checking remains tractable by exploring a bounded unfolding of the game graph that respects the fixed complexity bound, demonstrating that it is in PSPACE. To demonstrate practicality, we implement our algorithms within the open-source VITAMIN tool (Ferrando and Malvone 2024b) and evaluate their performance on a suite of benchmarks, including an adversarial drone-rescue scenario designed to highlight the interplay between fuzzy satisfaction, energy limitations, and natural strategy construction. This comprehensive framework aims to offer the first end-to-end approach to strategic verification that unifies interpretability, quantitative reasoning, and cost-awareness within a single, executable formalism.

Related Works. The seminal quantitative Alternating-Time Temporal Logic first provided by Jamroga in 2008 (Jamroga 2008) helped to blossom interest in the interaction between logics for strategic reasoning and fuzziness. The other foundational work that inspired our approach is the contribution of Alechina et al. (Alechina et al. 2010) where they reason about the availability of resources and the lack of a straightforward way of reasoning about resource requirements in ATL (Alur, Henzinger, and Kupferman 2002). Subsequent works explored the impact of restrictions such as memory bounds and imperfect information on expressiveness and decidability (Mogavero et al. 2014; Chen et al. 2013; Huang and van der Meyden 2014; Cermák et al. 2014; Cermák, Lomuscio, and Murano 2015; Gutierrez et al. 2018; Kurpiewski et al. 2021; Lomuscio, Qu, and Raimondi 2017). In parallel, temporal and strategic logics were extended to quantitative settings: finite-trace LTL$_F$ with averaging semantics (Almagor, Boker, and Kupferman 2016), threshold-based CTL$_F$ (Ma et al. 2024), continuous-truth ATL$[\mathcal{F}]$ (Ferrando et al. 2024), payoff-oriented Strategy Logic (SL$[\mathcal{F}]$) (Bouyer et al. 2023), as well as timed extensions (Brihaye et al. 2007; Henzinger and Prabhu 2006), multi-valued semantics (Jamroga et al. 2020; Belardinelli, Ferrando, and Malvone 2023), weighted and discounted objectives (Aminof et al. 2016, 2018; Ferrando and Malvone 2024a; Bulling and Goranko 2022; **?**; **?**; Laroussinie, Markey, and Oreiby 2006; Vester 2015; Almagor, Boker, and Kupferman 2014; De Alfaro et al. 2005; Mittelmann, Murano, and Perrussel 2023), probabilistic frameworks (Belardinelli et al. 2024; Chen and Lu 2007; Aminof et al. 2019; Chatterjee et al. 2009), and resource-constrained strategies (Nguyen et al. 2018; Catta, Ferrando, and Malvone 2024). Separately, the paradigm of natural strategies bounded the complexity of the strategy through simple representations based on rules, to improve interpretability without losing the determinability (Jamroga, Malvone, and Murano 2019). Foundational resource logics have also tracked consumable budgets in plays (Alechina et al. 2009, 2011, 2015), but none of these approaches is suitable for a more human-like representation of strategies to catch fuzzy achievements of goals using non-refillable resource budgets.

Outline. Section 2 defines the background notions for the reader. Section 3 provides the syntax and semantics of HumanATL$[\mathcal{F}]$. Section 4 presents the model-checking algorithms analysis. Section 5 describes the implementation. Section 6 reports the evaluation with a drone-rescue case study. Section 7 concludes the paper. All proofs are provided in the Technical Appendix of the supplementary material.

Preliminaries

We begin by introducing resource-bounded fuzzy Concurrent Game Structures (rfCGS), fuzzy operators, and the foundation for interpreting HumanATL$[\mathcal{F}]$ formulas and for defining its model checking problem.

Resource-Bounded Fuzzy Concurrent Game Structures

A Concurrent Game Structure (CGS) (Alur, Henzinger, and Kupferman 2002) is a formal model that effectively represents systems where multiple agents interact with each other and their environment. It is a graph where nodes denote system states, and edges show possible transitions between states. These transitions are non-deterministic, meaning that agents can choose from multiple possible actions in a given state and each chosen action leads to a new state. The overall outcome of the game emerges from the collective agents choices. Building on this foundation, our field often addresses the inadequacy of the CGS in verifying quantitative goals by employing a Weighted Concurrent Game Structure (wCGS) (Bouyer et al. 2023; Belardinelli et al. 2022). In a wCGS, every state is further characterized by fuzzy truth values (weights) assigned to propositions, quantifying the value of each property within the game. It is worth to note that, we extend the wCGS with a resource function res and a $consume$ function to handle action costs. We refer to the resulting model as a resource-bounded fuzzy Concurrent Game Structure (rfCGS).

Definition 1. *A rfCGS is a tuple* $G = (\mathrm{Ag}, \mathrm{Ap}, \{\mathrm{Act}_a\}_{a\in\mathrm{Ag}}, St, st_I, \ell, d, t, res, consume)$ *where:*

- Ag *is a not empty finite set of agents.*
- Ap *is a not empty finite set of atomic propositions (atoms).*

- *For every* $a \in \text{Ag}$, Act_a *is a not empty finite set of* actions. *Let* $Act = \bigcup_{a \in \text{Ag}} Act_a$ *be the set of all actions, and* $ACT = \prod_{a \in \text{Ag}} Act_a$ *the set of all joint actions.*
- St *is a finite set of states.*
- $st_I \in St$ *is an initial state.*
- $\ell : St \times \text{Ap} \to [0, 1]$ *is a weight function.*
- $d : \text{Ag} \times St \to 2^{\text{Act}}$ *is an availability function that defines a non-empty set of actions available to agents at each state.*
- t *is a transition function which assigns the outcome state* $st' = t(st, \boldsymbol{c})$ *to each state* st *and tuple of actions* $\boldsymbol{c} \in \prod_{a \in \text{Ag}} d(a, st)$ *that can be executed by the agents in* st.
- $res : \text{Ag} \to \mathbb{N}$ *is the resource function.*
- $consume : \text{Ag} \times \text{Act} \to \mathbb{N}$ *is the consume function.*

Obviously, res defines the maximum total action cost that agent a can incur during execution. With $consume(a, \alpha)$ we specify the cost for agent a to perform action α. Given a joint action $\boldsymbol{c}$ and a coalition of agents A, we use $\boldsymbol{c}_A$ to denote the projection of $\boldsymbol{c}$ onto the actions of the agents in A, and $\boldsymbol{c}_{\text{Ag} \setminus A}$ to denote the projection of $\boldsymbol{c}$ onto the actions of the agents outside A.

Fuzzy Operators

We characterize the Boolean operators $\wedge$, $\vee$ and $\neg$ with the quantitative counterparts functions $min\{x, y\}$, $max\{x, y\}$, and $1 - x$. Specifically, $\varphi_1 \vee \varphi_2$ yields the maximum of the values of φ_1 and φ_2, $\varphi_1 \wedge \varphi_2$ yields their minimum, and $\neg\varphi$ equals 1 minus the value of φ; consequently, the implication $\varphi_1 \to \varphi_2$ is defined as the maximum of φ_2 and $(1 - \varphi_1)$. These three truth functions correspond to the original fuzzy logic semantics formulated by Joseph Goguen (Goguen 1969).

Natural Strategies

We now review the definition of natural strategies and their complexities (Jamroga, Malvone, and Murano 2019), introducing memoryless and recall-based strategies.

Memoryless Strategies. We introduce the notion of a *natural memoryless strategy* (nr-strategy) s_a for an agent a, defined as a rule-based, condition-action representation. Formally, a natural strategy is given by an ordered list of guarded actions, i.e., pairs (φ_i, α_i) such that:

1. $\varphi_i \in \text{Bool}(\text{Ap})$ is a propositional condition on states of the rfCGS,
2. $\alpha_i \in d_a(q)$ for every state $q \in St$ where $q \models \varphi_i$.

We require that the last pair is of the form $(\top, \alpha)$, ensuring a default action is always available. The collection of all natural memoryless strategies for agent a is denoted by Σ_a^{nr}. We denote by $length(s_a)$ the number of guarded actions in s_a, by $cond_i(s_a)$ and $act_i(s_a)$ the ith condition and action respectively, and by $match(q, s_a)$ the smallest index $i \leq length(s_a)$ for which $q \models cond_i(s_a)$ and $act_i(s_a) \in d_a(st)$. Moreover, we define $dom(\varphi) = \{p \in \text{Ap} \mid p \in \varphi\}$ and $dom(s_a) = \bigcup_{i=1}^{length(s_a)} dom(cond_i(s_a))$. A *collective natural strategy* for a group of agents $A = \{a_1, ..., a_{|A|}\}$ is the tuple $s_A = (s_{a_1}, ..., s_{a_{|A|}})$, with the set of such strategies denoted by Σ_A^{nr}. The outcome function $out(st, s_A)$ returns all paths that can occur when the agents in A execute strategy s_A starting from state st. Formally, for $st \in St$:

$$
\begin{aligned}
out(st, s_A) = \{\pi \in \Pi \mid \pi[0] = st \wedge \forall_{i \geq 0} \exists_{\alpha_1, ..., \alpha_{|\mathbb{Agt}|}} : \\
(a \in A \Rightarrow \alpha_a = act_{match(\pi[i], s_a)}(s_a)) \wedge \\
(a \notin A \Rightarrow \alpha_a \in d_a(\pi[i])) \wedge \\
(\pi[i+1] = t(\pi[i], \alpha_1, ..., \alpha_{|\mathbb{Agt}|}))\}.
\end{aligned}
$$

Note that $out(st, s_A)$ encompasses *all* paths consistent with s_A, without imposing any assumptions on the strategies or behavior of the opponents.

Recall-based Strategies. Memoryless strategies are often insufficient when decisions depend on the history of the game. Agents with memory can base their choices on past states, which can be represented using automaton states (Vester 2013). However, we propose a more intuitive approach: using regular expressions over propositional formulas. So, let $Reg(L)$ be the set of regular expressions over the language L with standard operations concatenation $\cdot$, nondeterministic choice $\bigcup$, and Kleene star $*$. A natural strategy with recall (nR-strategy) s_a for agent a is a sequence of pairs from $Reg(\text{Bool}(\text{Ap})) \times \text{Act}$, namely a pair (r, a) with r being a regular expression over Bool(Ap), and a an action available in last(h) for all histories h consistent with r. Formally, given a r and the language $L(r)$ on words generated by r, an $h = q_0...q_n$ is consistent with r iff $\exists\, d \in L(r)$ such that $|h| = |d|$ and $\forall_{0 \leq j \leq n}\ h[j] \models d[j]$. For an individual agent a, the set of all such strategies is denoted Σ_a^{nR}. To decide which rule applies to a given h the matching function $match(h[0, j], s_a)$ is the smallest $n \leq length(s_a)$ such that $\forall_{0 \leq m \leq j} h[j] \models cond_n(s_a)[m]$ and $act_n(s_a) \in d_a(\pi[m])$. The function $out(q, s_a)$ continues to denote the set of all paths starting from a state q that are consistent with the collective strategy s_a.

Natural Strategy Complexity The *complexity* $compl(s_a)$ of a natural strategy is determined by the size of its representation. In the case of nr-strategies, complexity is measured by the total number of symbols in the Boolean conditions, while for nR-strategies, it is quantified by the total size of all the regular expressions used. A collective strategy (s_A) is equal to the sum of its individual strategies complexities (s_a), formally $compl_{\Sigma(s_A)} = \sum_{i=1,...,n} compl_{\Sigma}(s_{a_i})$. Hereafter in this paper, we will refer to each strategy type using the abbreviations nr and nR. By adopting natural strategies, we establish HumanATL$[\mathcal{F}]$ combining them with a quantitative approach.

Model Path. In a rfCGS G, a path π represents an infinite sequence of states. The set of paths over St is denoted by St^ω. For a joint natural strategy s_A, consisting of one strategy for each agent in coalition A, a path π is s_A-compatible if, for every $j \geq 1$, $\pi_{j+1} = \text{out}(\pi_j, c)$ for some joint action c such that for every $i \in A$, $c_i = s_i(\pi_{\leq j})$, and for every $i \in A$, $c_i \in d(i, \pi_j)$. The set of all s_A-compatible paths from s is denoted by out(s, s_A).

HumanATL[$\mathcal{F}$]

In this section, we provide a more precise formalization of human-like strategies, that takes into account a cost over the actions and then give the formal syntax and semantics of HumanATL[$\mathcal{F}$] and illustrate its use on the concrete example drawn from the aforementioned drone scenario.

Human Resource-Bounded Strategies. We extend the definition of natural strategies by incorporating resource constraints on the feasibility of actions. As stated above, a natural strategy for an agent a is a sequence of guarded actions (φ, α). We define cost over action integrating it as (φ, α_τ), where φ is a Boolean formula over atomic propositions, defining the guard under which the rule applies; $\alpha \in d(a, s)$ is an action available to agent a in any state st such that $\mathcal{G}, st \models \varphi$; $\tau \in \mathbb{N}$ is the fixed resource cost for executing action α. The association of costs and actions are defined via the resource consumption function $consume : \text{Ag} \times \text{Act} \to \mathbb{N}$, where $consume(a, \alpha)$ specifies the cost for agent a to perform action α. Each rule in a natural strategy must satisfy $\tau = consume(a, \alpha)$, ensuring that costs are consistently assigned based solely on the agent and the action, independent of the system state. Each agent $a \in \text{Ag}$ is initially endowed with a finite quantity of resources, given by the function: $res : \text{Ag} \to \mathbb{N}$, which defines the maximum total cost that agent a can incur during actions. In particular, for each rule (φ, α_τ) in the strategy of agent a, it must hold that the collection of total action cost $b \leq res(a)$. Before the presentation of HumanATL[$\mathcal{F}$], we introduce some notation that will be used throughout the paper. We denote the length of a tuple v as $|v|$, its j-th element as v_j, and its last element $v_{|v|}$ as $last(v)$. For $j \leq |v|$, let $v_{\geq j}$ be the suffix $v_j, ..., v_{|v|}$ of v starting from v_j and $v_{\leq j}$ the prefix $v_1, ..., v_j$ of v.

HumanATL[$\mathcal{F}$] Syntax. The grammar of HumanATL[$\mathcal{F}$] is given by the following definition:

Definition 2. *Formulas φ in* HumanATL[$\mathcal{F}$] *are defined as follows:*

$$\varphi ::= p \mid f[\varphi, ..., \varphi] \mid \langle\langle A \rangle\rangle^{\leq k}_{\leq b} \mathbf{X}\varphi \mid \langle\langle A \rangle\rangle^{\leq k}_{\leq b} (\varphi \mathbf{U} \varphi) \mid \langle\langle A \rangle\rangle^{\leq k}_{\leq b} (\varphi_1 \mathbf{R} \varphi_2)$$

*where $p \in \text{Ap}$, $A \in 2^{\text{Ag}}$, $k \in \mathbb{N}$ is the complexity bound for the strategies of agents in A, $b \in \mathbb{N}$ is the resource bound over actions, and $f \in \mathcal{F}$, where $\mathcal{F} \subseteq \{f : [0,1]^m \to [0,1] \mid m \in \mathbb{N}\}$ represents the set of computable functions. Temporal operators **G** and **F** are classically derived.*

HumanATL[$\mathcal{F}$] Semantics. Formulas of HumanATL[$\mathcal{F}$] are evaluated over a weighted rfCGS $G = (S, ..., \ell, ...)$ with atomic propositions in $[0, 1]$. Let $\pi = (st_1, st_2, ...)$ be a path. The satisfaction degree $[\![\varphi]\!]^G_\pi$ is defined by:

- $[\![p]\!]^G_\pi = \ell(st_1, p)$
- $[\![f[\varphi_1, ..., \varphi_m]]\!]^G_\pi = f([\![\varphi_1]\!]^G_\pi, ..., [\![\varphi_m]\!]^G_\pi)$
- $[\![\langle\langle A \rangle\rangle^{\leq k}_{\leq b} \mathbf{X}\psi]\!]^G_\pi = \max_{s \in \Sigma^{k,b}_A} (\min_{\pi' \in \text{out}(st_1, s)} ([\![\psi]\!]^G_{\pi'_{\geq 2}}))$
- $[\![\langle\langle A \rangle\rangle^{\leq k}_{\leq b} \mathbf{G}\psi]\!]^G_\pi = \max_{s \in \Sigma^{k,b}_A} (\min_{\pi' \in \text{out}(st_1, s)} (\min_{j \geq 1} ([\![\psi]\!]^G_{\pi'_{\geq j}})))$
- $[\![\langle\langle A \rangle\rangle^{\leq k}_{\leq b} (\psi_1 \mathbf{U} \psi_2)]\!]^G_\pi =$
  $= \max_{s \in \Sigma^{k,b}_A} (\min_{\pi' \in \text{out}(st_1, s)} (\max_{j \geq 1} (\min([\![\psi_2]\!]^G_{\pi'_{\geq j}}, \min_{i < j} ([\![\psi_1]\!]^G_{\pi'_{\geq i}})$
- $[\![\langle\langle A \rangle\rangle^{\leq k}_{\leq b} (\psi_1 \mathbf{R} \psi_2)]\!]^G_\pi =$
  $= \max_{s \in \Sigma^{k,b}_A} (\min_{\pi' \in \text{out}(st_1, s)} (\min_{j \geq 1} (\max([\![\psi_2]\!]^G_{\pi'_{\geq j}}, \max_{i < j} ([\![\psi_1]\!]^G_{\pi'_{\geq i}})$

$\Sigma^{k,b}_A$ is the set of all A–natural strategies with complexity at most k and total actions cost at most b, and $\text{out}(st_1, s)$ denotes the outcome paths from st_1 under strategy s.

Example 1. *Consider the coalition $\langle\langle carrier, drone \rangle\rangle$ operating on the model from Figure 3 at the carrier position pawn, with strategy complexity bound $k = 2$, the total energy budget $b = 5$, and the atomic propositions*

$$p = safe, \quad q = (dist \leq 0.5),$$

where p holds in the green "Rescue Zone" and q flags any state whose carrier–villain distance drops below 0.5. We have the HumanATL[$\mathcal{F}$] *formula*

$$\varphi = \langle\langle carrier, drone \rangle\rangle^{\leq 2}_{\leq 5} (\neg q \, \mathbf{U} \, p),$$

meaning the coalition has a natural strategy of at most two guard-action rules and spends no more than five energy units in total, so that the carrier reaches the Rescue Zone while the villain never comes within 0.5.

A suitable joint strategy of complexity 2 is:

Rule 1: ***if*** $(x \leq 0.6 \wedge y \geq 0.3)$ ***then***
$(carrier : move\_right, \ drone : move\_right)$,

Rule 2: ***otherwise if*** $(x > 0.6)$ ***then***
$(carrier : ascend, \ drone : ascend)$.

Here each agent's action costs 1 energy unit per step, so a joint step costs 2. Starting from $(0, 0)$, the coalition (i) executes Rule 1 twice, moving both agents to column $x = 0.6$ (cost $2 \times 2 = 4$), then (ii) applies Rule 2 once to ascend into the Rescue Zone $[0.3, 0.6] \times [0.3, 0.6]$ (cost 2), for a total cost of 6. To respect the budget $b = 5$, one can instead perform a single application of Rule 1 (cost 2) followed by a single application of Rule 2 (cost 2), then a final move by the carrier alone (cost 1) while the drone hovers (still described by Rule 2) bringing the total to $2 + 2 + 1 = 5$. Throughout, $dist > 0.5$ is maintained, so φ is satisfied.

Model Checking

In this section, we study algorithms and the complexity of the model-checking problem for HumanATL[$\mathcal{F}$] with both *nr*-strategies and *nR*-strategies, i.e., HumanATL$_\text{r}$[$\mathcal{F}$] and HumanATL$_\text{R}$[$\mathcal{F}$]. We analyze both constant and variable bounds k on the size of natural strategies.

HumanATL$_\text{r}$[$\mathcal{F}$] Model Checking

In this subsection, we establish that the model checking problem for HumanATL$_\text{r}$[$\mathcal{F}$] remains computationally efficient when the complexity bound k and the resource bound b are fixed. The model checking algorithm runs in linear time with respect to the size of the model, and the overall complexity remains polynomial in the size of the model and the length of the formula. The formal result follows.

Algorithm 1: HumanATL[F] Memoryless Model Checking

procedure HUMANATLFMEMORYLESSMODELCHECK(model, $\langle\langle A\rangle\rangle^{\leq k}_{\leq b}\psi$)
  for all s in GenerateStrategyCandidates(model, A, k, b) **do**
    $M_s \leftarrow$ PruneModel$(model, s)$
    if CTL[F]ModelChecking(M_s, ψ) **then**
      return TRUE
    end if
  end for
  return FALSE
end procedure

Theorem 1. *The model checking problem for* HumanATL$_r[\mathcal{F}]$ *with fixed complexity bound k and fixed resource bound b is in P with respect to the size of the model and the length of the formula.*

A summarization of the verification procedure is provided through algorithm 1.

We now examine the complexity of HumanATL$_r[\mathcal{F}]$ when the bounds in the strategic modalities are treated as variables. We begin by establishing NP-completeness for formulas that consist of a single strategic operator followed by a simple temporal subformula. Then, we modify the argument to demonstrate that model checking for the entire HumanATL$_r[\mathcal{F}]$ is Δ_2^P-complete.

Proposition 1. *The model check for* 1HumanATL$_r[\mathcal{F}]$ *(i.e., with a single strategic operator) is in NP with respect to the size of the model, the length of the formula, the variable value of the bound k, and the variable value of the bound b.*

Algorithm 2 establishes this result. We emphasize that this result is stated in terms of the value of k and b, or equivalently, the size of its unary encoding. This perspective will be maintained throughout the paper, and the analysis of model-checking complexity with respect to the binary representation of k and b is left for future research.

Theorem 2. 1HumanATL$_r[\mathcal{F}]$ *is NP-complete with respect to the size of the model, the length of the formula, and the variable value of k.*

Given the above result, we can present the general case.

Algorithm 2: 1HumanATL$_r[\mathcal{F}]$ Model Checking

procedure 1HUMANATLFMODELCHECKING(M, $\langle\langle A\rangle\rangle^{\leq k}_{\leq b}\gamma$)
  Guesss_A in GenerateStrategyCandidates(M, A, k, b)
    $M_s \leftarrow$ PruneModel$(model, s)$
    if CTL[F]ModelChecking(M_s, ψ) **then**
      return TRUE
    end if
  EndGuess
  return FALSE
end procedure

Theorem 3. *Model Checking* HumanATL$_r[\mathcal{F}]$ *is Δ_2^P-complete with respect to the size of the model, the length of the formula, the maximal bound k, and maximal cost b.*

HumanATL$_R[\mathcal{F}]$ Model Checking

We now examine the HumanATL$_R[\mathcal{F}]$ model-checking problem. Bounded tree unfoldings represent outcome sets of recall strategies.

Proposition 2. *Let M be a rfCGS with a finite state set St_M, and let $s_A = (s_{a_1}, ..., s_{a_n}) \in \Sigma_A^{nR}$ be a natural resource-bounded strategy with recall for coalition A with complexity $k = \mathrm{compl}(s_A)$. Suppose that every action has an associated cost, and each agent starts with a fixed, non-refillable amount of resource $res(a) \in \mathbb{N}$. Assume that goal is a temporal formula of the form $\varphi\,\mathcal{U}\,\psi$. Then, to decide whether s_A enforces the objective from a state $st \in \mathrm{St}_M$, it is sufficient to consider the tree unfolding of the executions prescribed by s_A up to a depth $L = |\mathrm{St}_M| \cdot 2^{2k^2} \cdot \prod_{a\in A}(r_a + 1)$.*

Based on the above proposition, we can now present the decision procedure as a concrete algorithm. Algorithm 3 implements the bounded-depth unfolding up to $L = |\mathrm{St}_M| \cdot 2^{2k^2} \prod_{a\in A}(r_a + 1)$ and checks the $\varphi\,\mathcal{U}\,\psi$ objective under s_A.

We now establish that model checking for HumanATL$_R[\mathcal{F}]$ can be performed using only polynomial space.

Theorem 4. *Model Checking of* HumanATL$_R[\mathcal{F}]$ *is in PSPACE with respect to the size of the model, the formula length, the maximal complexity bound k, and the maximal cost bound b.*

Implementation

In this section, we present the algorithms that implement the proposed HumanATL$[\mathcal{F}]$ logic. The implementation is designed to be consistent with the theoretical foundations established in the previous sections. Figure 1 illustrates our HumanATL$[\mathcal{F}]$ model-checking workflow, implemented in Python and organized into three phases: Strategy Generation, Model Pruning, and Model Checking. In the Strategy

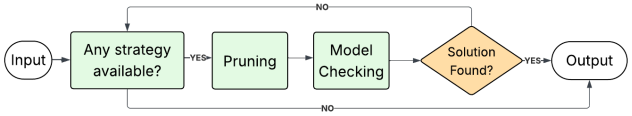

Figure 1: HumanATL$[\mathcal{F}]$ Model Checking process

Generation phase, we enumerate each agent's guarded actions by pairing guards (boolean or regex conditions) with available moves, form complete strategy profiles via Cartesian product, and enforce bounded deviation: for every strategy profile, each alternative action must also respect the specified complexity bound. A guarded action (φ, a) is accepted only if φ holds in all states where a is enabled and the sum of action costs (via `get_action_cost()`) does not exceed the agent's resource budget b. During Model Pruning, we refine the input model for each valid collective strategy s_A by removing all transitions that do not conform to

Algorithm 3: Bounded-Depth Unfolding for $\varphi \mathcal{U} \psi$ under s_A with resources

procedure CHECKOBJECTIVEUNDERSTRATEGY($M, s_A, q, \varphi, \psi, \{r_a\}_{a \in A}$)
  // Build DFAs for each regex condition in s_A
  for all regex r in s_A **do**
    $D_r \leftarrow$ REGEXTODFA(r)
  end for
  // Compute the combined depth bound including resources
  $L \leftarrow |\,\mathrm{St}_M| \times 2^{2k^2} \times \prod_{a \in A}(r_a + 1)$
  // Explore the outcome tree up to depth L
  Initialize queue Q; enqueue $(s = q, \{q_r = D_r.init\}_r,\ d = 0)$
  while Q not empty **do**
    $(s,\ (q_r)_r, d) \leftarrow Q$.dequeue()
    if $d > L$ **then**
      continue
    end if
    if $\neg\varphi(s)$ **then**
      return FALSE
    end if
    if $\psi(s)$ **then**
      continue
    end if
    for all joint action m prescribed by s_A at $(s, (q_r)_r)$ **do**
      $s' \leftarrow$ NEXTSTATE(M, s, m)
      for all regex r **do**
        $q'_r \leftarrow D_r.\delta(q_r,$ OBSERVE$(s, m))$
      end for
      Enqueue $(s',\ (q'_r)_r,\ d + 1)$ into Q
    end for
  end while
  return TRUE
end procedure

s_A, producing a smaller model tailored for efficient verification. In model checking for HumanATL$[\mathcal{F}]$, we adopt a binary abstraction on top of the underlying fuzzy semantics: although HumanATL$[\mathcal{F}]$ formulas are interpreted over the continuous range $[0, 1]$, we introduce meta-truth values *True* and *False* to indicate whether a formula is considered satisfied in a given state. After computing the fuzzy satisfaction degrees, we treat a formula as *True* exactly in those states whose degree meets or exceeds a designated threshold (in our implementation, effectively those target states where the computed degree equals 1.0[1]), and as *False* otherwise. These meta-values do not replace the fuzzy degrees but serve as a practical decision criterion for pruning and strategy validation. More specifically, the representation of fuzziness in our framework is realized via an explicit fuzzy Kripke structure (FKS) abstraction. From the original input model, we extract a mapping $R : S \times S \to [0, 1]$ by assigning to each joint action label a normalized degree in the interval $(0, 1)$ - using a simple Gougen fuzzy-logic scheme where

$$\mu(a) = \frac{\mathrm{index}(a)}{|\mathsf{Labels}| + 1},$$

and propagate these degrees into the transition relation. Atomic propositions are also annotated with fuzzy truth values in $V : S \times \mathrm{Ap} \to [0, 1]$ according to the input labelling. We then employ classical fuzzy-CTL fixed-point algorithms (EX, AX, EU, EG, AU, AF, AG) instantiated with Gouguen's minimum and maximum, as well as standard negation and implication operators, to compute a real-valued satisfaction map $[[\varphi]] : S \to [0, 1]$. The degree of satisfiability for each state is then used both to inform the meta-truth abstraction and to drive the decision whether a strategy yields a winning outcome. This integrated approach ensures that our implementation faithfully realizes the fuzzy semantics of HumanATL$[\mathcal{F}]$ with resource bound actions while maintaining practical efficiency. A key aspect of our implementation is that, beyond indicating whether a solution exists for the given (pruned) model, the algorithm also synthesizes the optimal winning strategy leading to that solution. Specifically, if a solution validating the formula is found, the tool returns both the set of states that satisfy it and the minimal-complexity strategy that guarantees its satisfaction. This ensures that, rather than merely verifying existence, our framework actively constructs and returns an executable strategy for the agents. If the verification does not yield a solution for the pruned model, the algorithm backtracks to the Strategy Generation phase, projects new strategies onto the original model, and re-attempts verification. This iterative cycle continues until a valid strategy is synthesized or no further strategies remain viable.

Experiments

We evaluated our HumanATL$[\mathcal{F}]$ verifier on an HPOmen 15-ax213ng (Intel i7-7700HQ 3.8GHz, 16GB RAM), implemented in Python 3.9 (PyCharm) and integrated into the open-source VITAMIN checker (Ferrando and Malvone 2024c). VITAMIN provides a robust baseline, supporting memoryless and bounded-recall strategies with bounded complexity, against which we compared our module. Unlike MCMAS (Lomuscio, Qu, and Raimondi 2017), its implementation handles natural strategies. To validate correctness, we ran 100 tests per temporal operator, using identical inputs for both ATL$[\mathcal{F}]$ and our HumanATL$[\mathcal{F}]$ formulas, with a complexity bound up to 5. Across nearly 1000 trials, HumanATL$[\mathcal{F}]$ matched the established ATL$[\mathcal{F}]$ results, confirming implementation fidelity. For performance, we conducted 1000 additional tests, varying model size (number of states), complexity bound, resource bound, and agent count. Execution time grew with both the number of agents and the complexity bound, since each agent's strategy set expands with its action space. For bounded-recall strategies, we fixed the tree depth at 5 to avoid excessive unwind time[2]. Recall-based histories added overhead compared to memoryless runs. Figure 2 illustrates average-case execution times on sparse vs. dense transition matrices with

[1] obviously, if the user needs a narrower truth range it possible to associate more fuzzy-values to the true meta value

[2] Depths >5 caused noticeable delays during tree traversal.

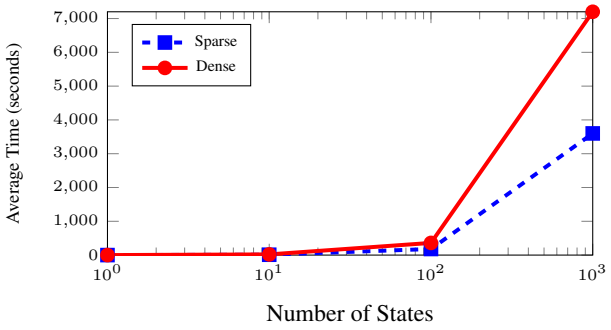

Figure 2: Average case: Average time table comparison using dense and sparse transition matrix and $|A| = 3$

three agents, k=5, and a varying resource bound (as it does not significantly impact performance under our implementation). Dense matrices (higher branching) incur substantially longer runtimes, while sparse matrices execute more quickly. We additionally measured worst-case behavior on fully nondeterministic models (maximally dense matrices) and average-case behavior on classical incomplete multi-agent models, observing that filtering invalid strategies yields more efficient, typical runtimes.

Case Study: Resource-Bound Drone Battle

To demonstrate the practical expressiveness of HumanATL[$\mathcal{F}$], we consider a case study grounded in autonomous aerial navigation, a domain where strategy design must inherently balance correctness, readability, and resource feasibility. Our scenario is set in an war-like environment where battery recharging is impossible, emphasizing the critical importance of judicious energy management. In our example, a carrier drone is tasked with transporting a critical artifact to a designated rescue zone while continuously avoiding proximity to a hostile pursuer (the villain drone). The spatial environment is discretized along both axes into the finite domain $\{0, 0.3, 0.6, 1\}$, effectively modeling a grid-based operational area. Each drone can move in discrete steps, consuming irreplaceable battery power at each action. The environment is formally encoded as a rfCGS, enriched with action costs and fuzzy valued propositions such as `dist` (distance between drones) and `safe` (degrees of positional security), both ranging over $[0, 1]$.

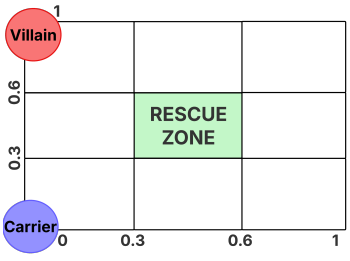

Figure 3: Drone Battle scenario: the carrier and villain occupy grid positions and evaluate `dist` and `safe` on each state.

Each guarded rule in a strategy not only specifies a boolean or regex condition, but also consumes energy when its action is done. For instance, an "ascend" maneuver might cost 2 battery units, "right" costs 3, and "idle" costs 1. Before pruning, every candidate strategy is checked so that the sum of its per-use action costs does not exceed the drone's battery budget (e.g. $b = 5$). Only those strategies that both satisfy their distance-based conditions and remain within budget pass to the pruning phase. Strategically, the carrier uses two guarded rules ($k = 2$) to avoid the villain and reach safety: $s_{\text{carrier}} = \{(\neg(\,\text{dist} < 0.5\,), \rightarrow),\ (\top, \uparrow)\}$. The first rule tells the carrier to move right when the villain is too close, the second to ascend otherwise; the combined energy cost along the path (e.g. $3 + 2 = 5$) exactly matches the budget. The villain's one-rule strategy ($k = 1$) with $b = 4$ is specified as $s_{\text{villain}} = \{(\text{dist} \geq 0.5, \downarrow)\}$, pursuing the carrier whenever distance permits. After pruning, the resulting game graph (Figure 4) retains only states reachable under the selected strategies and within the specified resource bounds. This focused model highlights exactly the critical interactions. Finally, the tool synthesizes and returns this same pair of strategies as the winning coalition profile, namely $s^*_{\text{carrier}} = \{(\neg(\text{dist} < 0.5), \rightarrow), (\top, \uparrow)\}$ and $s^*_{\text{villain}} = \{(\text{dist} \geq 0.5, \downarrow)\}$, which is guaranteed to satisfy $\langle\langle \mathit{carrier} \rangle\rangle^{\leq 2}_{\leq 5} \neg(\text{dist} < 0.5)\ \mathbf{U}\ \mathit{safe}$.

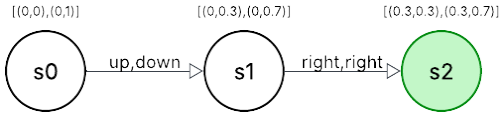

Figure 4: Pruned Drone Battle model: shows only transitions whose joint actions meet both strategic guards and energy budgets.

Conclusion

In this paper, we have addressed the problem of modeling human-like strategies in MAS by explicitly accounting for bounded rationality, non-zero action costs, and uncertain/noisy perceptions over the time. We have encoded strategies as concise, rule-based controllers with fuzzy semantics for predicates and real-valued action costs drawn from a non-refillable budget. This enables verification of safety and performance under realistic constraints, better orchestration of human-AI handoffs, and auditable explanations. The approach is especially relevant in search-and-rescue robotics, cybersecurity, healthcare and mixed traffic, and critical-infrastructure operations. Technically, we introduced HumanATL[$\mathcal{F}$], provided its syntax and cost-aware fuzzy semantics, and established complexity results: model checking is in P when complexity bound k and budget b are fixed, NP-complete in case of a single strategic operator over Boolean objectives, and Δ_2^P-complete when k and b can vary; for recall-based strategies, a bounded unfolding of the game graph yields a PSPACE decision procedure. Our implementation in VITAMIN and its evaluation on a suite of benchmarks demon-

strate that these guarantees translate into practical performance. Future work will address scalability via neurosymbolic methods involving LLM-based strategy generation. Also, naturalness and expressiveness will be refined to better reflect human-like reasoning through knowledge operators, belief systems, and strategic hierarchies.

Acknowledgments

This research is supported by the following projects: ANR project NOGGINS ANR-24-CE23-440, PRIN 2020 RIPER - CUP E63C22000400001, ECS00000037-MUSA-INFANT, and the ECS00000009-Tech4You-APLAND.

References

Ågotnes, T. 2006. Action and knowledge in alternating-time temporal logic. *Synthese*, 149(2): 375–407.

Alechina, N.; Logan, B.; Nga, N. H.; and Rakib, A. 2009. Expressing Properties of Coalitional Ability under Resource Bounds. In *Logic, Rationality, and Interaction*, 1–14. Springer Berlin Heidelberg. ISBN 978-3-642-04893-7.

Alechina, N.; Logan, B.; Nga, N. H.; and Rakib, A. 2010. Resource-bounded alternating-time temporal logic. In *Proc. of the 9th International Conference on Autonomous Agents and Multiagent Systems*, AAMAS 2010, 481–488. International Foundation for Autonomous Agents and Multiagent Systems. ISBN 9780982657119.

Alechina, N.; Logan, B.; Nga, N. H.; and Rakib, A. 2011. Logic for coalitions with bounded resources. *J. Log. Comput.*, 21(6): 907–937.

Alechina, N.; Logan, B.; Nguyen, H. N.; Raimondi, F.; and Mostarda, L. 2015. Symbolic Model-checking for Resource-Bounded ATL. In *Proc. of the 2015 International Conference on Autonomous Agents and Multiagent Systems, AAMAS 2015*, 1809–1810. ACM.

Almagor, S.; Boker, U.; and Kupferman, O. 2014. Discounting in LTL. In *TACAS*, volume 8413 of *Lecture Notes in Computer Science*, 424–439. Springer.

Almagor, S.; Boker, U.; and Kupferman, O. 2016. Formally reasoning about quality. *Journal of the ACM (JACM)*, 63(3): 1–56.

Alur, R.; Henzinger, T. A.; and Kupferman, O. 2002. Alternating-time temporal logic. *Journal of the ACM (JACM)*, 49(5): 672–713.

Aminof, B.; Kwiatkowska, M.; Maubert, B.; Murano, A.; and Rubin, S. 2019. Probabilistic Strategy Logic. In *Proc. of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI 2019*, 32–38. ijcai.org.

Aminof, B.; Malvone, V.; Murano, A.; and Rubin, S. 2016. Graded Strategy Logic: Reasoning about Uniqueness of Nash Equilibria. In *Proc. of the 2016 International Conference on Autonomous Agents & Multiagent Systems, AAMAS 2016*, 698–706. ACM.

Aminof, B.; Malvone, V.; Murano, A.; and Rubin, S. 2018. Graded modalities in strategy logic. *Information and Computation*, 261: 634–649.

Belardinelli, F.; Ferrando, A.; and Malvone, V. 2023. An abstraction-refinement framework for verifying strategic properties in multi-agent systems with imperfect information. *Artif. Intell.*, 316: 103847.

Belardinelli, F.; Jamroga, W.; Malvone, V.; Mittelmann, M.; Murano, A.; and Perrussel, L. 2022. Reasoning about Human-Friendly Strategies in Repeated Keyword Auctions. In *Proceedings of the 21st International Conference on Autonomous Agents and Multiagent Systems*, AAMAS 2022, 62–71. International Foundation for Autonomous Agents and Multiagent Systems. ISBN 9781450392136.

Belardinelli, F.; Jamroga, W.; Mittelmann, M.; and Murano, A. 2024. Verification of Stochastic Multi-Agent Systems with Forgetful Strategies. In *Proceedings of the 23rd International Conference on Autonomous Agents and Multiagent Systems, AAMAS 2024*, 160–169. International Foundation for Autonomous Agents and Multiagent Systems / ACM.

Bouyer, P.; Kupferman, O.; Markey, N.; Maubert, B.; Murano, A.; and Perelli, G. 2023. Reasoning about Quality and Fuzziness of Strategic Behaviors. *ACM Transactions on Computational Logic*, 24(3): 1–38.

Brihaye, T.; Laroussinie, F.; Markey, N.; and Oreiby, G. 2007. Timed Concurrent Game Structures. In *Proc. of the 18th International Conference on Concurrency Theory, CONCUR 2007*, volume 4703 of *Lecture Notes in Computer Science*, 445–459. Springer.

Bulling, N.; and Goranko, V. 2022. Combining quantitative and qualitative reasoning in concurrent multi-player games. *Auton. Agents Multi Agent Syst.*, 36(1): 2.

Catta, D.; Ferrando, A.; and Malvone, V. 2024. Resource Action-based Bounded ATL: a New Logic for MAS to express a cost over the actions. In *Proc. of the 25th International Conference on Principles and Practice of Multi-Agent Systems PRIMA 2024*, volume 15395 of *Lecture Notes in Computer Science*, 206–223. Springer.

Cermák, P.; Lomuscio, A.; Mogavero, F.; and Murano, A. 2014. MCMAS-SLK: A Model Checker for the Verification of Strategy Logic Specifications. In *Proc. of the 26th International Conference on Computer Aided Verification CAV 2014*, volume 8559 of *Lecture Notes in Computer Science*, 525–532. Springer.

Cermák, P.; Lomuscio, A.; and Murano, A. 2015. Verifying and Synthesising Multi-Agent Systems against One-Goal Strategy Logic Specifications. In *Proc. of the Twenty-Ninth AAAI Conference on Artificial Intelligence, AAAI'15*, 2038–2044. AAAI Press.

Chatterjee, K.; de Alfaro, L.; Faella, M.; and Legay, A. 2009. Qualitative Logics and Equivalences for Probabilistic Systems. *Log. Methods Comput. Sci.*, 5(2).

Chatterjee, K.; Henzinger, T. A.; and Piterman, N. 2010. Strategy logic. *Information and Computation*, 208(6): 677–693.

Chen, T.; Forejt, V.; Kwiatkowska, M. Z.; Parker, D.; and Simaitis, A. 2013. PRISM-games: A Model Checker for Stochastic Multi-Player Games. In *Proc. of the 19th International Conference on Tools and Algorithms for the Construc-*

tion and Analysis of Systems TACAS 2013, volume 7795 of *Lecture Notes in Computer Science*, 185–191. Springer.

Chen, T.; and Lu, J. 2007. Probabilistic Alternating-time Temporal Logic and Model Checking Algorithm. In *In Proc. of the 4th International Conference on Fuzzy Systems and Knowledge Discovery, FSKD 2007*, 35–39. IEEE Computer Society.

De Alfaro, L.; Faella, M.; Henzinger, T. A.; Majumdar, R.; and Stoelinga, M. 2005. Model checking discounted temporal properties. *Theoretical Computer Science*, 345(1): 139–170.

Ferrando, A.; Luongo, G.; Malvone, V.; and Murano, A. 2024. Theory and Practice of Quantitative ATL. In *In Proc. of the International Conference on Principles and Practice of Multi-Agent Systems PRIMA 2024:*, volume 15395 of *Lecture Notes in Computer Science*, 231–247. Springer. Winner of the Best Paper Award at PRIMA 2024.

Ferrando, A.; and Malvone, V. 2024a. Hands-on VITAMIN: A Compositional Tool for Model Checking of Multi-Agent Systems. In Alderighi, M.; Baldoni, M.; Baroglio, C.; Micalizio, R.; and Tedeschi, S., eds., *Proceedings of the 25th Workshop "From Objects to Agents", Bard (Aosta), Italy, July 8-10, 2024*, volume 3735 of *CEUR Workshop Proceedings*, 148–160. CEUR-WS.org.

Ferrando, A.; and Malvone, V. 2024b. VITAMIN: A Compositional Framework for Model Checking of Multi-Agent Systems. *CoRR*, abs/2403.02170.

Ferrando, A.; and Malvone, V. 2024c. VITAMIN: A Tool for Model Checking of MAS. In Collier, R.; Ricci, A.; and Nallur, V., eds., *Multi-Agent Systems - 21st European Conference, EUMAS 2024, Dublin, Ireland, August 26-28, 2024, Proceedings*, volume to appear of *Lecture Notes in Computer Science*. Springer.

Goguen, J. A. 1969. The logic of inexact concepts. *Synthese*, 19(3/4): 325–373.

Gutierrez, J.; Najib, M.; Perelli, G.; and Wooldridge, M. J. 2018. EVE: A Tool for Temporal Equilibrium Analysis. In Lahiri, S. K.; and Wang, C., eds., *Automated Technology for Verification and Analysis - 16th International Symposium, ATVA 2018, Los Angeles, CA, USA, October 7-10, 2018, Proceedings*, volume 11138 of *Lecture Notes in Computer Science*, 551–557. Springer.

Henzinger, T. A.; and Prabhu, V. S. 2006. Timed Alternating-Time Temporal Logic. In *Proc. of the 4th International Conference on Formal Modeling and Analysis of Timed Systems, FORMATS 2006*, volume 4202 of *Lecture Notes in Computer Science*, 1–17. Springer.

Huang, X.; and van der Meyden, R. 2014. Symbolic Model Checking Epistemic Strategy Logic. In Brodley, C. E.; and Stone, P., eds., *Proceedings of the Twenty-Eighth AAAI Conference on Artificial Intelligence, July 27 -31, 2014, Québec City, Québec, Canada*, 1426–1432. AAAI Press.

Jamroga, W. 2008. A temporal logic for stochastic multi-agent systems. In *Pacific Rim International Conference on Multi-Agents*, 239–250. Springer.

Jamroga, W.; Konikowska, B.; Kurpiewski, D.; and Penczek, W. 2020. Multi-valued Verification of Strategic Ability. *Fundam. Informaticae*, 175(1-4): 207–251.

Jamroga, W.; Kwiatkowska, M.; Penczek, W.; Petrucci, L.; and Sidoruk, T. 2025. Probabilistic Timed ATL. In *Proceedings of the 24th International Conference on Autonomous Agents and Multiagent Systems*, 1051–1059.

Jamroga, W.; Malvone, V.; and Murano, A. 2019. Natural strategic ability. *Artif. Intell.*, 277.

Knapik, M. J.; André, É.; Petrucci, L.; Jamroga, W.; and Penczek, W. 2019. Timed ATL: forget memory, just count. *Journal of Artificial Intelligence Research*, 66: 197–223.

Kurpiewski, D.; Pazderski, W.; Jamroga, W.; and Kim, Y. 2021. STV+Reductions: Towards Practical Verification of Strategic Ability Using Model Reductions. In Dignum, F.; Lomuscio, A.; Endriss, U.; and Nowé, A., eds., *AAMAS '21: 20th International Conference on Autonomous Agents and Multiagent Systems, Virtual Event, United Kingdom, May 3-7, 2021*, 1770–1772. ACM.

Laroussinie, F.; Markey, N.; and Oreiby, G. 2006. Model-Checking Timed. In *Proc. of the 4th International Conference on Formal Modeling and Analysis of Timed Systems, FORMATS 2006*, volume 4202 of *Lecture Notes in Computer Science*, 245–259. Springer.

Lomuscio, A.; Qu, H.; and Raimondi, F. 2017. MCMAS: an open-source model checker for the verification of multi-agent systems. *Int. J. Softw. Tools Technol. Transf.*, 19(1): 9–30.

Ma, Z.; Li, X.; Liu, Z.; Huang, R.; and He, N. 2024. Model checking fuzzy computation tree logic of multi-agent systems based on fuzzy interpreted systems. *Fuzzy Sets Syst.*, 485: 108966.

Mittelmann, M.; Murano, A.; and Perrussel, L. 2023. Discounting in Strategy Logic. In *Proceedings of the Thirty-Second International Joint Conference on Artificial Intelligence, IJCAI 2023, 19th-25th August 2023, Macao, SAR, China*, 225–233. ijcai.org.

Mogavero, F.; Murano, A.; Perelli, G.; and Vardi, M. Y. 2014. Reasoning About Strategies: On the Model-Checking Problem. *ACM Trans. Comput. Log.*, 15(4): 34:1–34:47.

Nguyen, H. N.; Alechina, N.; Logan, B.; and Rakib, A. 2018. Alternating-time temporal logic with resource bounds. *J. Log. Comput.*, 28(4): 631–663.

Schnoor, H. 2009. Probabilistic ATL with incomplete information.

Vester, S. 2013. Alternating-time temporal logic with finite-memory strategies. *arXiv preprint arXiv:1307.4476*.

Vester, S. 2015. On the Complexity of Model-Checking Branching and Alternating-Time Temporal Logics in One-Counter Systemss. In Finkbeiner, B.; Pu, G.; and Zhang, L., eds., *Automated Technology for Verification and Analysis*, 361–377. Cham: Springer International Publishing. ISBN 978-3-319-24953-7.