Reasoning about Bias in Multi-Agent Systems Verification

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

Chunyan Mu*
University of Aberdeen
Aberdeen, United Kingdom
Chunyan.Mu@abdn.ac.uk

ABSTRACT

This paper investigates formal analysis of *bias* properties in Multi-Agent Systems. We present a formal framework to reason about bias properties within these models. We propose a novel definition of bias based on *attribute interference* and formalise the notion of bias by trace equivalence regarding bias-sensitive lattice. Intuitively, a model is considered unbiased or fair if the input values of bias-sensitive attributes will not affect the system's output, ensuring that the system treats all bias-sensitive attributes equitably. To effectively capture and specify bias and fairness properties, we extend Alternating-time Temporal Logic (ATL) with new bias operators as bATL. Using model checking techniques, we propose algorithm to rigorously verify these properties within the game structure.

KEYWORDS

Multi-Agent Systems; Strategic Logics; Model Checking; Bias

ACM Reference Format:
Vadim Malvone and Chunyan Mu. 2026. Reasoning about Bias in Multi-Agent Systems Verification. In *Proc. of the 25th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2026), Paphos, Cyprus, May 25 – 29, 2026*, IFAAMAS, 9 pages. https://doi.org/10.65109/RTAL7233

1 INTRODUCTION

As intelligent systems become increasingly integrated into decision-making processes, ranging from hiring and lending to criminal justice and healthcare, the need for fair and transparent Multi-Agent Systems (MASs) is more critical than ever. These systems often involve interactions among humans, AI agents, and institutions, where bias can emerge from model assumptions or strategic behaviours. Formal analysis provides a principled and rigorous way to specify, detect, and reason about bias, enabling system designers to verify whether certain outcomes or behaviours are fair across different groups. However, this task is highly non-trivial: bias can be subtle, context-dependent, and temporally dynamic, and arises not just from individual actions but from complex agent interactions. Addressing these challenges requires expressive formal tools that can capture *strategic behaviour*, *temporal evolution*, and *structured notions* of fairness, which motivates the development of notions and formal specification tailored to bias reasoning in MASs.

We introduce a formal definition of *bias* in MASs, based on the concept of *interference* between bias-sensitive inputs and the public outputs of the system. Intuitively, a system is considered *unbiased* or *fair* if its outputs remain consistent when bias-sensitive attributes are varied, provided that non-bias attributes are held constant. This ensures that changes in bias-sensitive inputs, such as gender or race, do not influence the system's intermediate behaviours and/or final outcomes, thereby guaranteeing equitable treatment of all such attributes. Our definition, inspired by the notion of *noninterference* [15] from the information security community, provides a rigorous way to formalise state and path equivalence with respect to bias-sensitive properties. This approach is well-suited to MASs, where fairness and bias are inherently tied to how the system handles equivalently relevant states (i.e., bias-sensitive states). By ensuring that two states equivalent in terms of non-bias-sensitive attributes yield consistent system behaviour, noninterference eliminates the risk of unintended bias arising from structural or contextual discrepancies within the model. This provides a robust mathematical foundation for analysing and mitigating bias in dynamic and interactive systems. Although simpler definitions, such as identifying bias through paths where biased states affect outcomes, may appear straightforward, they lack the ability to capture the subtle interdependencies within MASs. Bias often arises not just from direct paths but from how states are *related* based on bias-sensitive attributes. Noninterference addresses this by ensuring fairness at a more structural and comprehensive level.

We then propose a logical framework, bias-aware Alternating-time Temporal Logic (bATL) and bATL *, to rigorously specify and reason about this notion of bias. Building on ATL (ATL*) [1, 6], bATL (bATL *) introduces bias-sensitive operators, enabling the formalisation of properties related to bias within MASs, such as those found in human–AI interaction models. The bATL (bATL *) framework is interpreted over concurrent game structures, representing interactions among agents.

Furthermore, we develop verification algorithms tailored to both bATL and bATL * logic, enabling the automatic verification of multiagent systems against formally specified bias-related properties. By embedding bias reasoning within a strategic temporal logic framework, our approach extends bias analysis to dynamic and strategic settings, providing a significant and novel contribution to the field of fair AI. We show that the model-checking problems for bATL and bATL * in our model are decidable, with complexities PTIME-complete and 2EXPTIME-complete, respectively.

Outline. The paper is organised as follows. Section 2 reviews the underlying model based on Concurrent Game Structures (CGSs) and the Alternating-time Temporal Logic (ATL*) framework. Section 3 presents the foundational model for formal bias analysis in MASs, including our novel definitions of *bias* and *bias policy*, together with their formalisation in this setting. Section 4 defines the bATL * logic for expressing and reasoning about bias. Section 5 and 6 describe the model checking procedure for bATL and bATL * and analyses their computational complexity respectively. Finally, Sections 7 and 8

*Corresponding author.

Proc. of the 25th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2026), C. Amato, L. Dennis, V. Mascardi, J. Thangarajah (eds.), May 25 – 29, 2026, Paphos, Cyprus. https://doi.org/10.65109/RTAL7233

discuss related work and provide concluding remarks & future works, respectively.

2 PRELIMINARIES

In this section we recall some preliminary notions. Given a set U, $\overline{U}$ denotes its complement, U^+ denotes the set of all finite sequence over U. 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 $v_j, \ldots, v_{|v|}$ of v starting from v_j and $v_{\leq j}$ the prefix $v_1, \ldots, v_j$ of v.

2.1 Model

We start by showing a formal model for Multi-Agent Systems (MASs) via concurrent game structures [1].

Definition 1. *A* concurrent game structure (CGS) *is a tuple* $M = \langle \mathsf{Ag}, \mathsf{Ap}, S, s_I, \{\mathsf{Act}_i\}_{i\in\mathsf{Ag}}, d, \delta, V\rangle$ *such that:*

- $\mathsf{Ag} = \{1, \ldots, m\}$ *is a nonempty finite set of agents.*
- Ap *is a nonempty finite set of atomic propositions (atoms).*
- $S \neq \emptyset$ *is a finite set of* states*, with a set of* initial states $s_I \subseteq S$.
- *For every* $i \in \mathsf{Ag}$, Act_i *is a nonempty finite set of* actions. *Let* $\mathsf{Act} = \bigcup_{i\in\mathsf{Ag}} \mathsf{Act}_i$ *be the set of all actions, and* $\mathsf{ACT} = \prod_{i\in Ag} \mathsf{Act}_i$ *the set of all joint actions.*
- *The* protocol function $d : \mathsf{Ag} \times S \to (2^{\mathsf{Act}} \setminus \{\emptyset\})$ *defines the availability of actions so that for every* $i \in \mathsf{Ag}$, $s \in S$, $d(i, s) \subseteq \mathsf{Act}_i$.
- *The* transition function $\delta : S \times \mathsf{ACT} \to S$ *assigns a successor state* $s' = \delta(s, \vec{a})$ *to each* $s \in S$*, for every joint action* $\vec{a} \in \mathsf{ACT}$ *such that* $a_i \in d(i, s)$ *for every* $i \in \mathsf{Ag}$.
- $V : S \to 2^{\mathsf{Ap}}$ *is the* labelling function.

By Definition 1 an CGS describes the interactions of a group Ag of agents, starting from a initial state $s \in s_I$, according to the transition function δ. The latter is constrained by the availability of actions to agents, as specified by the protocol function d.

2.2 Syntax

We recall ATL* [1] to reason about the strategic abilities of agents.

Definition 2. *State* (φ) *and path* (ψ) *formulas in ATL* are defined as follows:*

$$\varphi \quad ::= \quad q \mid \neg\varphi \mid \varphi \wedge \varphi \mid \langle\langle\Gamma\rangle\rangle\psi$$
$$\psi \quad ::= \quad \varphi \mid \neg\psi \mid \psi \wedge \psi \mid \bigcirc\psi \mid (\psi \mathbf{U} \psi)$$

where $q \in \mathsf{Ap}$ *and* $\Gamma \subseteq \mathsf{Ag}$. *Formulas in ATL* are all and only the state formulas.*

As usual, a formula $\langle\langle\Gamma\rangle\rangle\Phi$ is read as "the agents in coalition Γ have a strategy to achieve Φ". The meaning of temporal operators *next* $\bigcirc$ and *until* $\mathbf{U}$ is standard [3]. Operators *release* $\mathbf{R}$, *eventually* $\Diamond$, and *globally* $\Box$ can be introduced as usual.

Formulas in the ATL fragment are obtained from Definition 2 by restricting path formulas as follows:

$$\psi \quad ::= \quad \bigcirc\varphi \mid (\varphi \mathbf{U} \varphi) \mid (\varphi \mathbf{R} \varphi)$$

2.3 Semantics

First, we give the formal definition of strategy.

Definition 3. *A* perfect recall strategy *for agent* $i \in \mathsf{Ag}$ *is a function* $\sigma_i : S^+ \to \mathsf{Act}_i$ *such that for all* $h \in S^+$, $\sigma_i(h) \in d(i, last(h))$.

By Definition 3, any strategy for agent i has to return actions that are enabled for i. Furthermore, we obtain memoryless (or imperfect recall) strategies by considering the domain of σ_i in S, *i.e.* $\sigma_i : S \to \mathsf{Act}_i$. Given an CGS M, a *path* π is a finite or infinite sequence of states. We denote with S^ω the set of paths over S. Given a joint strategy $\vec{\sigma}_\Gamma$, comprising of one strategy for each agent in coalition Γ, a path π is $\vec{\sigma}_\Gamma$*-compatible* iff for every $j \geq 1$, $\pi_{j+1} = \delta(\pi_j, \vec{a})$ for some joint action $\vec{a}$ such that for every $i \in \Gamma$, $a_i = \sigma_i(\pi_{\leq j})$, and for every $i \in \overline{\Gamma}$, $a_i \in d(i, \pi_j)$. We denote with $\mathrm{out}(s, \vec{\sigma}_\Gamma)$ the set of all $\vec{\sigma}_\Gamma$-compatible paths from s.

Now, we have all the ingredients to give the semantics of ATL*.

Definition 4. *The satisfaction relation* $\models$ *for a CGS* M*, state* $s \in S$*, path* $\pi \in S^\omega$*, atom* $q \in \mathsf{Ap}$*, and ATL* formula* φ *is defined as (clauses for Boolean connectives are immediate and thus omitted):*

$(M, s) \models q$	*iff*	$q \in V(s)$
$(M, s) \models \langle\langle\Gamma\rangle\rangle\psi$	*iff*	*for some joint strategy* $\vec{\sigma}_\Gamma$*,* *for all* $\pi \in \mathrm{out}(s, \vec{\sigma}_\Gamma)$, $(M, \pi) \models \psi$
$(M, \pi) \models \varphi$	*iff*	$(M, \pi_1) \models \varphi$
$(M, \pi) \models \bigcirc\psi$	*iff*	$(M, \pi_{\geq 2}) \models \psi$
$(M, \pi) \models \psi \mathbf{U} \psi'$	*iff*	*for some* $k \geq 1$, $(M, \pi_{\geq k}) \models \psi'$*, and* *for all* $1 \leq j < k$, $(M, \pi_{\geq j}) \models \psi$

We say that formula φ is *true* in a CGS M, or $M \models \varphi$, iff $(M, s) \models \varphi$ for all $s \in s_I$. Now, we state the model checking problem.

Definition 5. *Given a CGS* M *and an ATL* formula* φ*, the global model checking problem concerns determining whether* $M \models \varphi$.

3 MODELLING BIAS

This section introduces the foundational model for formal bias analysis in MASs. We propose a novel definition of *bias* and *bias policy* inspired by the concept of non-interference [15], and formalise these concepts using equivalence classes.

3.1 Bias-sensitive Model

Definition 6. *A* bias-sensitive model *is a tuple of the form* $\mathcal{G}$=(M, η)*, where:*

- M *is a CGS;*
- $\eta : \mathsf{Ap} \mapsto \mathcal{L}$ *is the bias labelling function mapping each atomic proposition to a bias level in* $\mathcal{L}$ *representing the complete lattice of the bias levels.*

Let $\mathsf{Path}_\mathcal{G}$ denote the set of $\mathcal{G}$-paths and $\mathsf{Paths}_\mathcal{G}(s_I)$ denote the set of $\mathcal{G}$-paths, starting from any state in s_I.

Example 1. *Consider a simple scenario of college admission system with three agents: a student applicant A, an AI reviewer B, and a human reviewer C. Each application is evaluated based on the following attributes: 'merit' (*1*-high,* 0*-low), 'origin' (*1*-national,* 0*-overseas), 'demographic group' (*1*-over represented group* G_1*,* 0*-under represented group* G_2*). The set of atomic propositions also includes: 'decision' (*1*-accepted,* 0*-unaccepted). We assume a three-tier bias structure:* $\eta(\mathtt{merit}) = \eta(\mathtt{decision}) = 0$, $\eta(\mathtt{origin}) = 1$, *and* $\eta(\mathtt{group}) = 2$. *This hierarchy reflects that demographic group carries the highest*

bias sensitivity, origin has a moderate bias level, and the remaining attributes are considered bias-free. The available actions for each agent are defined as follows:

$$\text{Act}_A = \{\texttt{apply}, \texttt{receive}, \texttt{idle}\}$$
$$\text{Act}_B = \{\texttt{accept}, \texttt{reject}, \texttt{idle}\}$$
$$\text{Act}_C = \{\texttt{approve}, \texttt{override}, \texttt{idle}\}$$

*If agent C choose '*approve *', the decision made by the AI reviewer is retained; if C chooses '*override *', the decision is flipped. The set of atomic propositions is given by:*

$$\text{Ap} = \{\texttt{merit}, \texttt{origin}, \texttt{group}, \texttt{decision}\}$$

with each proposition taking values in $\{0, 1\}$*. Figure 1 visualises the model. To simplify the presentation, we divide the procedure into two*

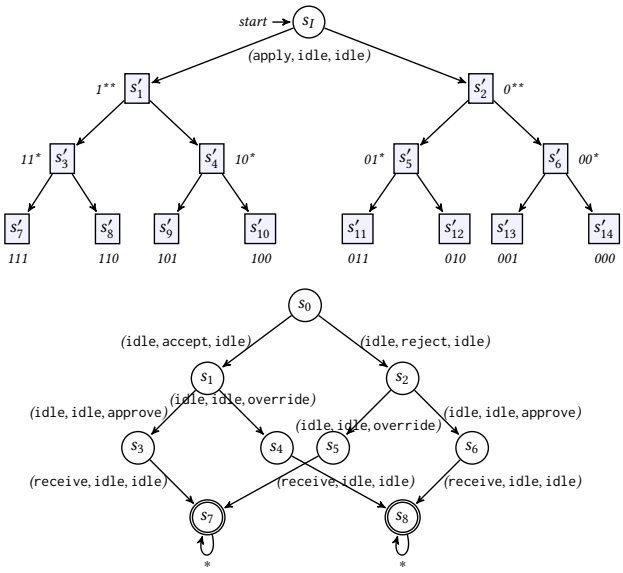

Figure 1: Example: College Admission System. The top part of the figure represents a pre-phase submission process that is not explicitly included in the formal model while the bottom part of the figure represents the model in which we abstract the set of initial states with s_0. $*$ indicates decisions that have not yet been specified in the current state, e.g., "1" represents** `merit` $= 1$ **but** `origin` **and** `group` **are still pending or undefined at this stage.**

parts. The top part of the figure represents the application submission process, capturing different scenarios based on the applicant's profile, where each possible state $\{s'_i \mid 7 \le i \le 14\}$ *corresponds to a different condition of the applicant:*

$$V(s'_7) = \{\texttt{merit}, \texttt{origin}, \texttt{group}\} \quad V(s'_8) = \{\texttt{merit}, \texttt{origin}\}$$
$$V(s'_9) = \{\texttt{merit}, \texttt{group}\} \quad V(s'_{10}) = \{\texttt{merit}\}$$
$$V(s'_{11}) = \{\texttt{origin}, \texttt{group}\} \quad V(s'_{12}) = \{\texttt{origin}\}$$
$$V(s'_{13}) = \{\texttt{group}\} \quad V(s'_{14}) = \emptyset$$

They are represented as s_0 *(i.e.,* $s_0 \in \{s'_i \mid 7 \le i \le 14\}$*) and passed to the bottom part of the figure, which handles application processing and decision-making. Note that the AI reviewer may exhibit bias: e.g., if* `group` $= 1$ *and* `merit` $= 1$ *he recommend accept, if* `group` $= 0$ *and* `merit` $= 1$ *he may recommend reject (which can be considered being biased). The human reviewer may confirm or override AI's decision. After an application is received, the system can be in one of* 8 *possible states. State* $s_0 = \{s'_i \mid 7 \le i \le 12\}$ *represents the stage where an application has been received and is pending evaluation. States* s_1 *and* s_2 *correspond to the AI reviewer recommending "accept" and "reject" respectively. States* s_3 *and* s_6 *represent the human reviewer confirming the AI reviewer's recommendation of "accept" and "reject", respectively. States* s_4 *and* s_5 *capture the cases where the human reviewer "overrides" the AI reviewer's recommendation of "accept" and "reject", resulting in "reject" and "accept", respectively. States* s_7 *and* s_8 *denote the final admission outcomes "accept" and "reject" respectively. Consider example paths with input* `merit` $= 1$, `origin` $= 1$, `group` $= 0$*:*

$$\pi_1 = (s'_8 \in s_0) \rightarrow s_2 \rightarrow s_6 \rightarrow s_8$$
$$\pi_2 = (s'_8 \in s_0) \rightarrow s_2 \rightarrow s_5 \rightarrow s_7$$

and example paths with input `merit` $= 1$, `origin` $= 1$, `group` $= 1$*:*

$$\pi_3 = (s'_7 \in s_0) \rightarrow s_1 \rightarrow s_3 \rightarrow s_7$$
$$\pi_4 = (s'_7 \in s_0) \rightarrow s_1 \rightarrow s_4 \rightarrow s_8$$

In a naive analysis, π_1 *might suggest that a qualified candidate from group* G_2 *is rejected due to the AI reviewer's bias, while* π_2 *implies that the human reviewer corrects this bias. Similarly,* π_3 *may indicate that a qualified candidate from group* G_1 *is accepted, whereas* π_4 *suggests the human reviewer may (wrongly) override the AI's correct decision.*

3.2 Formalising Bias

Note that bias-sensitive attribute information is essentially contained in the atomic state propositions in our model. To reason about the property of bias in MASs, we assign bias-sensitive levels with ordering to state atomic propositions, through $\eta : \text{Ap} \mapsto \mathcal{L}$, where $\mathcal{L}$ denotes the complete lattice of the bias-sensitive levels. The powerset of atomic state propositions therefore forms a complete lattice $\mathcal{L}$, whose partial ordering is regarding to the bias-sensitive levels of the atomic propositions Ap:

$$\forall v, v' \in \text{Ap},\ v \preceq v' \text{ iff } \eta(v) \le \eta(v')$$

where $\preceq$ denotes the partial ordering on Ap and $\eta(v), \eta(v') \in \mathcal{L}$ denotes the bias-sensitive levels of v and v', respectively.

Using a *lattice* to define bias levels, rather than a strictly ordered set, allows for a more flexible and expressive representation of bias relationships. A strict total order assumes that all biases can be ranked along a single dimension, which may not accurately capture the complexity of real-world biases. In contrast, a *lattice structure* enables the representation of multiple dimensions of bias, accommodating both hierarchical sensitivity levels and category-based distinctions. For instance, the bias structure may consist of two components: classification levels (H) and categories (C). The classification levels H represent an ordered hierarchy (e.g., $\{0, 1, 2\}$ or {*low, medium, high*}), while the categories C capture specific categories (e.g., {*White, Mixed, Asian, Black*}) with a partial order defined by the *inverse of the subset inclusion*, since more specific category would be more sensitive to bias. A *partial order* ($\le$) is

defined over the set of bias labels such that:

$$(h_1, c_1) \leq (h_2, c_2) \quad \text{iff} \quad h_1 \leq h_2 \text{ and } c_1 \supseteq c_2.$$

For instance, clearly: $(\texttt{merit}, \{\textit{Black, White}\}) \leq (\texttt{group}, \{\textit{White}\})$. since the sensitive classification level of `merit` is less than that of `group` and $\{\textit{Black, White}\} \supseteq \{\textit{White}\}$. This lattice-based approach allows for a structured yet adaptable representation, ensuring that bias levels reflect both sensitivity degrees and group-specific considerations, thereby enhancing the accuracy and applicability of bias assessments in decision-making systems. The focus of this paper is the formalisation of bias properties within a logical framework, which necessitates a structured approach to categorising bias-sensitive attributes. For clarity, we adopt a two-level classification system: $\mathcal{B}$ (bias-sensitive, e.g., age, gender, demographic group), and $\mathcal{NB}$ (neutral, non-bias-sensitive, e.g., userID), with ordering $\mathcal{NB} \leq \mathcal{B}$; however, this approach can be naturally extended to accommodate multiple levels if needed.

Intuitively, a model $\mathcal{G}$ is considered "biased" if the behaviours or outcomes of the model satisfy a *bias policy*. An outcome can be defined as the set of paths that satisfy an ATL formula φ. We denote $\mathsf{Paths}_{\mathcal{G}}(s, \varphi)$ as the set of paths starting from s that lead to an outcome characterised by φ. We now propose the bias policy that a biased model should satisfy, to do so, we first present the definition of $\mathcal{NB}$-equivalent state and $\mathcal{B}$-equivalent state.

Definition 7 ($\mathcal{NB}$-equivalent state). *We say state $s, s' \in S$ are $\mathcal{NB}$-equivalent, written as $s =_{\mathcal{NB}} s'$, iff:*

$$\forall x \in \mathsf{Ap}.\eta(x) \leq \mathcal{NB} \Rightarrow (x \in V(s) \wedge x \in V(s')) \vee (x \notin V(s) \wedge x \notin V(s')).$$

Intuitively, two states are $\mathcal{NB}$-equivalent, if they are equivalent to each other when bias-sensitive atomic propositions are removed, focusing solely on the non-bias-sensitive atomic propositions.

Example 2. *Continue to consider scenario presented in Example 1, let $\eta(\texttt{merit}) = \eta(\texttt{origin}) = \eta(\texttt{decision}) = \mathcal{NB}$, and $\eta(\texttt{group}) = \mathcal{B}$, and $\mathcal{NB} \preceq \mathcal{B}$, i.e., demographic group is bias-sensitive attribute while merit, origin, and final decision are non-bias-sensitive. We have:*

$$s'_7 =_{\mathcal{NB}} s'_8,\ s'_9 =_{\mathcal{NB}} s'_{10},\ s'_{11} =_{\mathcal{NB}} s'_{12},\ s'_{13} =_{\mathcal{NB}} s'_{14}.$$

Definition 8 (Weak and Strong $\mathcal{NB}$-equivalent path). *Given a model $\mathcal{G}$, and two paths $\pi, \pi' \in \mathsf{Path}_{\mathcal{G}}$. We say that two paths π and π' are* weakly $\mathcal{NB}$-equivalent, *written as $\pi \sim_{\mathcal{NB}_{\mathrm{w}}} \pi'$, iff:*

If $|\pi| < |\pi'|$: $\pi_0 =_{\mathcal{NB}} \pi'_0 \Rightarrow \exists j \geq |\pi|.\forall k \geq j.(last(\pi) =_{\mathcal{NB}} \pi'_k)$;

If $|\pi'| < |\pi|$: $\pi_0 =_{\mathcal{NB}} \pi'_0 \Rightarrow \exists j \geq |\pi'|.\forall k \geq j.(\pi_k =_{\mathcal{NB}} last(\pi'))$;

Else $|\pi| = |\pi'|$: $\pi_0 =_{\mathcal{NB}} \pi'_0 \Rightarrow last(\pi) =_{\mathcal{NB}} last(\pi')$.

We say that π and π' are strongly $\mathcal{NB}$-equivalent, *written as $\pi \sim_{\mathcal{NB}_{\mathrm{s}}} \pi'$, iff:*

$$(|\pi| = |\pi'|) \wedge (\pi_0 =_{\mathcal{NB}} \pi'_0 \Rightarrow \forall j \in (0, |\pi|].(\pi_j =_{\mathcal{NB}} \pi'_j)).$$

Intuitively, two paths are weakly $\mathcal{NB}$-equivalent if, whenever they start from non-bias-equivalent initial states, any differences caused by bias-sensitive attributes do not persist to the end of the execution, so that their final states are indistinguishable with respect to non-bias-sensitive attributes. In particular, even if the paths have different lengths, any divergence caused by bias-sensitive attributes is required to wash out after some point: once the shorter path terminates, the longer path must eventually reach (and remain in) a state that is non-bias-equivalent to the final state of the shorter path. Thus, bias-sensitive differences may affect intermediate steps, but they are not allowed to influence the eventual outcome. In contrast, two paths are *strongly $\mathcal{NB}$-equivalent* if they have the same length and, whenever their initial states are non-bias-equivalent, all corresponding states along the paths keep non-bias-equivalent. This enforces that bias-sensitive attributes never affect non-bias-sensitive attributes (e.g., certain decisions) at any point in time, guaranteeing bias-free behaviour throughout the entire execution rather than only in the final result.

Proposition 1. *Given a model $\mathcal{G}$, for any two path $\pi, \pi' \in \mathsf{Path}_{\mathcal{G}}$, we have:*

$$\pi \sim_{\mathcal{NB}_{\mathrm{s}}} \pi' \Rightarrow \pi \sim_{\mathcal{NB}_{\mathrm{w}}} \pi'$$

Proof. Directly from Definition 8, as the conditions for strong $\mathcal{NB}$-equivalence are stricter than those for weak one. □

Example 3. *Continue to consider Example 1, it is easy to see that, $\pi_1 \nsim_{\mathcal{NB}_{\mathrm{w}}} \pi_3$, since the initial states of the two path are $\mathcal{NB}$-equivalent but final states of them are not $\mathcal{NB}$-equivalent: $s_7 \neq_{\mathcal{NB}} s_8$. Similarly, we have: $\pi_1 \sim_{\mathcal{NB}_{\mathrm{w}}} \pi_4$, $\pi_2 \sim_{\mathcal{NB}_{\mathrm{w}}} \pi_3$, $\pi_2 \nsim_{\mathcal{NB}_{\mathrm{w}}} \pi_4$.*

Definition 9 (Bias policy). *Let ψ be an outcome formula and $\mathcal{G} = (\mathsf{Ag}, S, s_I, \mathsf{Act}, \rightarrow, \mathsf{Ap}, L, \eta)$ a model.*

- *The model $\mathcal{G}$ is said to be* weakly biased *with respect to ψ, written $\mathcal{G} \Vdash_{\psi} \phi_{\mathcal{B}_{\mathrm{w}}}$, iff*

$\mathsf{Paths}_{\mathcal{G}}(s_I, \psi) = \emptyset \ \vee \ \exists \pi, \pi' \in \mathsf{Paths}_{\mathcal{G}}(s_I, \psi)$ *implies* $\pi \nsim_{\mathcal{NB}_{\mathrm{w}}} \pi'$.

- *The model $\mathcal{G}$ is said to be* strongly biased *with respect to ψ, written $\mathcal{G} \Vdash_{\psi} \phi_{\mathcal{B}_{\mathrm{s}}}$, iff*

$\mathsf{Paths}_{\mathcal{G}}(s_I, \psi) = \emptyset \ \vee \ \exists \pi, \pi' \in \mathsf{Paths}_{\mathcal{G}}(s_I, \psi)$ *implies* $\pi \nsim_{\mathcal{NB}_{\mathrm{s}}} \pi'$.

Here, $\mathsf{Paths}_{\mathcal{G}}(s_I, \psi)$ denotes the set of complete paths from the initial states s_I that satisfy ψ, and $\emptyset$ denotes the empty set.

A *bias policy* characterises whether a system exhibits biased behaviour w.r.t. a given outcome ψ. Intuitively, a model is considered *weakly biased* w.r.t. ψ if either the outcome ψ is unreachable from the initial states, or there exist two executions leading to ψ that are not weakly $\mathcal{NB}$-equivalent. In the latter case, although both executions achieve the same outcome, they differ in a way that cannot be accounted for by non-bias-sensitive attributes alone. This indicates that bias-sensitive attributes influence the *final outcome*. The notion of *strong bias* is more demanding. A model is strongly biased w.r.t. ψ if either ψ is unreachable, or there exist two executions leading to ψ that are not strongly $\mathcal{NB}$-equivalent, i.e., are distinguishable at some point along their execution with respect to non-bias-sensitive attributes. This means that the executions become disinguishable at some point along their evolution w.r.t. non-bias-sensitive attributes, implying that bias-sensitive attributes affects the system's *behaviour during the execution*, not merely the final result.

Proposition 2. *Given a model $\mathcal{G}$ and an outcome ψ, we have:*

$$\mathcal{G} \Vdash_{\psi} \phi_{\mathcal{B}_s} \Rightarrow \mathcal{G} \Vdash_{\psi} \phi_{\mathcal{B}_w}$$

Proof. This follows directly from Proposition 1 and Definition 9. Since strong $\mathcal{NB}$-equivalence is stricter than weak $\mathcal{NB}$-equivalence, any violation of the strong condition also violates the weak one. □

Example 4. *Continue to consider Example 1, let $\psi = \Diamond(\texttt{decision})$, clearly there exists $\pi_1, \pi_3 \in \mathsf{Paths}_{\mathcal{G}}(s_I, \psi)$ s.t. $\pi_1 \nsim_{\mathcal{NB}_w} \pi_3$, so $\mathcal{G}$ is a weakly biased model regarding ψ. This meets our intuition since the bias-sensitive attribute of demographic group impacts the final decision while non-bias-sensitive attributes remains the same, i.e., the final result is influenced by bias-sensitive attributes, throughout the path.*

Remark 1 (Model-level vs. strategy-level bias). *Note that the bias policy in Definition 9 is defined at the level of the model rather than with respect to a fixed strategy profile. This allows us to capture* structural bias*, namely whether the system admits biased behaviour under some strategic interaction. A strategy-level notion would instead characterise bias of a particular joint policy. Our model-level definition thus identifies potential bias inherent in the system design, independently of how agents resolve their choices. If desired, a strategy-dependent notion of bias can be obtained by restricting $\mathsf{Paths}_{\mathcal{G}}(s_I, \psi)$ to paths induced by a given strategy profile.*

4 THE BIAS-SENSITIVE LOGIC

This section introduces bATL*, a logical framework for specifying bias in our game model, as an extention of ATL*.

Definition 10 (bATL* syntax). *The syntax of* bATL* *includes two classes of formulae:* state formulae *and* path formulae *ranged over by φ and ψ, respectively.*

$$\begin{aligned} \varphi &::= q \mid \neg\varphi \mid \varphi \wedge \varphi \mid \langle\langle\Gamma\rangle\rangle\psi \mid \langle\langle\Gamma\rangle\rangle\mathfrak{B}_w[\psi] \mid \langle\langle\Gamma\rangle\rangle\mathfrak{B}_s[\psi] \\ \psi &::= \varphi \mid \neg\psi \mid \psi \wedge \psi \mid \bigcirc\psi \mid (\psi \mathbf{U} \psi) \end{aligned}$$

where $q \in \mathsf{Ap}$ and $\Gamma \subseteq \mathsf{Ag}$.

The formulas $\langle\langle\Gamma\rangle\rangle\mathfrak{B}_w[\psi]$ and $\langle\langle\Gamma\rangle\rangle\mathfrak{B}_s[\psi]$ express that the coalition Γ has a strategy to ensure ψ under different bias conditions. Specifically, $\langle\langle\Gamma\rangle\rangle\mathfrak{B}_w[\psi]$ indicates that Γ can enforce ψ when the system exhibits weak bias, while $\langle\langle\Gamma\rangle\rangle\mathfrak{B}_s[\psi]$ states that Γ can enforce ψ even under strong bias. In other words, these formulas capture whether the system's decision-making is considered weakly or strongly biased concerning the outcome ψ when following a strategy chosen by coalition Γ. Just as ATL is a fragment of ATL*, bATL is a fragment of bATL*, by restricting path formulas as follows:

$$\psi ::= \bigcirc\varphi \mid (\varphi \mathbf{U} \varphi) \mid (\varphi \mathbf{R} \varphi)$$

Let us consider some illustrating example formulae:

- $\neg\langle\langle A\rangle\rangle\mathfrak{B}_w[\Diamond(\texttt{policyPassed})]$ captures that bias-sensitive factors (e.g., media influence, institutional discrimination) prevent party A from ensuring the policy's success through any strategy;
- $\langle\langle\Gamma\rangle\rangle\mathfrak{B}_w[\Box\,\neg\texttt{error}]$ states that under weak bias consideration, coalition Γ can ensure that no execution path leads to an error state.

Definition 11 (bATL* semantics). *The satisfaction relation $\models$ for a model $\mathcal{G}$, state $s \in S$, $x \in \{s, w\}$, and* bATL* *formula φ follows:*
$(\mathcal{G}, s) \models \langle\langle\Gamma\rangle\rangle\mathfrak{B}_x[\psi]$ *iff there exists a joint strategy $\vec{\sigma}_\Gamma$ s.t.*

$$[\exists \pi, \pi' \in \mathsf{out}(s, \vec{\sigma}_\Gamma).((\mathcal{G}, \pi) \models \psi \wedge (\mathcal{G}, \pi') \models \psi)] \Rightarrow (\pi \nsim_{\mathcal{NB}_x} \pi')$$

Example 5. *Consider Example 1, and the following two example properties:*

1) to express whether C has a strategy ensuring the system is biased when a final decision is made, we propose:

$$\varphi = \langle\langle C\rangle\rangle\mathfrak{B}_w[\Diamond\,(\texttt{decision})]$$

the result is true *as the agent C could choose to override the AI reviewer's decision whenever a non-bias answer is proposed.*

2) to express whether the system is biased regarding the final acceptance being made, we propose:

$$\varphi = \langle\langle A, B\rangle\rangle\mathfrak{B}_s[\Diamond\,(\texttt{decision})]$$

clearly φ is true*, i.e., the system is strongly biased, as we can find π, π' (e.g., π_1 and π_3 in Example 3) s.t. $\big((\mathcal{G}, \pi) \models \Diamond(\texttt{decision}) \wedge (M, \pi') \models \Diamond(\texttt{decision})\big) \Rightarrow \pi \nsim_{\mathcal{NB}_w} \pi'$. This also meets the results presented in Example 4.*

Theorem 3 (Correctness). *Given a model $\mathcal{G}$=(Ag, S, s_I, Act, $\rightarrow$, Ap, L, η), let ψ be the outcome aimed to achieve. We evaluate whether the system exhibits bias in its behaviours as it reaches the outcome ψ. We have:*

$$\mathcal{G} \Vdash_{\psi} \phi_{\mathcal{B}_w} \text{ iff } (\mathcal{G}, s) \models \langle\langle\mathsf{Ag}\rangle\rangle\mathfrak{B}_w[\psi].$$

Proof. We show both directions.

($\Rightarrow$) Assume $\mathcal{G} \Vdash_{\psi} \phi_{\mathcal{B}_w}$. By Definition 9, either (i) $\mathsf{Paths}_{\mathcal{G}}(s_I, \psi) = \emptyset$, or (ii) there exist $\pi, \pi' \in \mathsf{Paths}_{\mathcal{G}}(s_I, \psi)$ such that $\pi \nsim_{\mathcal{NB}_w} \pi'$.

In case (i), for any joint strategy profile $\vec{\sigma}$ and any $\pi \in \mathsf{out}(s_I, \vec{\sigma})$, we have $(\mathcal{G}, \pi) \not\models \psi$, and thus $(\mathcal{G}, s_I) \models \langle\langle\mathsf{Ag}\rangle\rangle\mathfrak{B}_w[\psi]$ holds trivially.

In case (ii), since every complete path is compatible with some joint strategy profile, there exists $\vec{\sigma}$ such that $\pi, \pi' \in \mathsf{out}(s_I, \vec{\sigma})$. Both paths satisfy ψ and are not weakly $\mathcal{NB}$-equivalent, hence $(\mathcal{G}, s_I) \models \langle\langle\mathsf{Ag}\rangle\rangle\mathfrak{B}_w[\psi]$ by Definition 11.

($\Leftarrow$) Assume $(\mathcal{G}, s_I) \models \langle\langle\mathsf{Ag}\rangle\rangle\mathfrak{B}_w[\psi]$. Then there exists a joint strategy profile $\vec{\sigma}$ such that either (i) at most one $\vec{\sigma}$-outcome satisfies ψ, or (ii) there exist $\pi, \pi' \in \mathsf{out}(s_I, \vec{\sigma})$ satisfying ψ with $\pi \nsim_{\mathcal{NB}_w} \pi'$. In both cases, by Definition 9, $\mathcal{G} \Vdash_{\psi} \phi_{\mathcal{B}_w}$ holds. □

Similar result can be obtained for strong bias property by replacing $\mathcal{NB}_w$ with $\mathcal{NB}_s$. This theorem shows that if the starting state of the model satisfies the bias formula following $\Gamma \subseteq$ Ag's enforcing strategy, then the system is considered biased according to the bias policy defined in Definition 9.

5 MODEL CHECKING bATL

The model checking problem for bATL formulae involves determining, given a model $\mathcal{G}$ and a bATL formula φ, the set of states of $\mathcal{G}$ that satisfy φ. Verification of bias operator $\langle\langle\Gamma\rangle\rangle\mathfrak{B}_w[\psi]$ answers the question "whether the system is biased or not, regarding an outcome expressed as $[\psi]$ following the strategy of a coalition $\Gamma \subseteq$ Ag?" As we have seen, we need to consider all possible paths reaching φ to compare relevant executions to check the satisfaction of the bias operator. The verification problem for bATL formulae is defined in the following way:

Definition 12 (Verification problem for bATL formulae.). *Given a bias-sensitive model $\mathcal{G}$=(Ag, Ap, $S, s_I, \{\mathrm{Act}_i\}_{i\in\mathrm{Ag}}\delta, V, \eta$) and a* bATL *formula φ, determine whether it is the case that every initial state $s \in s_I$ satisfies φ.*

Theorem 4. *The model checking problem for* bATL *in our model is decidable.*

To show the decidability of the problem, we describe the model checking procedure in Algorithm 2. Given $\mathcal{G}$, we denote by $[[\varphi]]$ the set of states that satisfy φ, i.e., $[[\varphi]] = \{s \in S \mid (\mathcal{G}, s) \models \varphi\}$. Let $\mathrm{Pre}(\Gamma, \mathcal{G}, S')$ denote the set of states from which there is a Γ-action $\vec{a}_\Gamma$ such that Γ can reach, by executing $\vec{a}_\Gamma$, only states in S', that is:

$$\mathrm{Pre}(\Gamma, \mathcal{G}, S') = \{s \in S \mid \exists \vec{a}_\Gamma \in \mathrm{Act}_\Gamma,\ \mathrm{Post}(s, \vec{a}_\Gamma) \subseteq S'\}$$

where $\mathrm{Post}(s, \vec{a}_\Gamma)$ is the immediate state successors of s by executing $\vec{a}_\Gamma$. Algorithm 1 is proposed to generate $\mathrm{Pre}(\Gamma, \mathcal{G}, S')$. Given a formula, the function $\mathrm{Sub}(\varphi)$ returns a queue of syntactic subformulae of φ such that if φ_1 is a subformula of φ and φ_2 is a subformula of φ_1, then φ_2 precedes φ_1 in the queue $\mathrm{Sub}(\varphi)$.

Algorithm 1 Pre($\Gamma, \mathcal{G}, S_x$)

$S_y \leftarrow \emptyset$
for each $s \in S$ **do**
  for each $\vec{a} \in \mathrm{ACT}_\Gamma$ **do**
    if $\mathrm{Post}(s, \vec{a}) \subseteq S_x$ **then**
      $S_y \leftarrow S_y \cup \{s\}$
return S_y

Algorithm 2 Model Checking bATL formula φ: MC($\mathcal{G}, s, \varphi$)

for each $\varphi' \in \mathrm{Sub}(\varphi)$ **do**
  match φ':
    case $q \in$ Ap: $[[\varphi]] \leftarrow \{s \in S \mid q \in V(s)\}$
    case $\neg\varphi_1$: $[[\varphi]] \leftarrow S \setminus [[\varphi_1]]$
    case $\varphi_1 \wedge \varphi_2$: $[[\varphi]] \leftarrow [[\varphi_1]] \cap [[\varphi_2]]$
    case $\langle\langle\Gamma\rangle\rangle \bigcirc \varphi_1$: $[[\varphi]] \leftarrow \mathrm{Pre}(\Gamma, \mathcal{G}, [[\varphi_1]])$
    case $\langle\langle\Gamma\rangle\rangle\varphi_1 \mathbf{U} \varphi_2$:
      $X \leftarrow \emptyset \quad Y \leftarrow [[\varphi_2]]$
      while $Y \not\subseteq X$ **do**
        $X \leftarrow X \cup Y$
        $Y \leftarrow \mathrm{Pre}(\Gamma, \mathcal{G}, X) \cap [[\varphi_1]]$
      $[[\varphi]] \leftarrow X$
    case $\langle\langle\Gamma\rangle\rangle\varphi_1 \mathbf{R} \varphi_2$:
      $X \leftarrow [[true]] \quad Y \leftarrow [[\varphi_2]]$;
      while $X \neq Y$ **do**
        $X \leftarrow Y$
        $Y \leftarrow (\mathrm{Pre}(\Gamma, \mathcal{G}, X) \cup [[\varphi_1]]) \cap [[\varphi_2]]$;
      $[[\varphi]] \leftarrow X$;
    case $\langle\langle\Gamma\rangle\rangle\mathfrak{B}_x[\psi]$: $[[\varphi]] \leftarrow \mathrm{SatBias}(\mathcal{G}, s_I, \Gamma, \psi, x)$
return $[[\varphi]]$

Theorem 5 (Complexity). *The model-checking problem for* bATL *in the CGS is PTIME-complete.*

Algorithm 3 Bias satisfaction checking: SatBias($\mathcal{G}, s_I, \Gamma, \psi, x$)

$\sigma_{\mathrm{sat}} \leftarrow \mathrm{GenStrat}(\Gamma, \mathcal{G}, [[\psi]])$
if $\sigma_{\mathrm{sat}} = \emptyset$ **then**
  return $\emptyset$ ▷ Trivially unbiased (no ψ-satisfying paths)
$\sigma_{\mathrm{sat}} \leftarrow \mathrm{CompleteStrategy}(\sigma_{\mathrm{sat}})$
for each $s \in s_I$ **do**
  for each $\pi, \pi' \in out(s, \sigma_{\mathrm{sat}})$ **do**
    if $\pi \nsim_{\mathcal{NB}_x} \pi'$ **then**
      return $\{s\}$ ▷ Bias detected
return $\emptyset$ ▷ No bias found

Algorithm 4 GenStrat($\Gamma, \mathcal{G}, T$)

for each $t \in T$ **do**
  $\sigma \leftarrow \emptyset$
  for each $s \in S$ **do**
    for each $\vec{a} \in \mathrm{Act}_\Gamma$ **do**
      if $\mathrm{Post}(s, \vec{a}) \subseteq T$ **then**
        $\sigma \cup \{(s, \vec{a})\}$
  return σ
return $\emptyset$

Proof. The proof consists of two parts. First, we show that model checking bATL formula without $\mathfrak{B}$ operators is in PTIME-complete. This follows from their correspondence to ATL formulas [1], for which model checking over CSGs is known to be PTIME-complete. Second, for formulas of the form $\langle\langle\Gamma\rangle\rangle\mathfrak{B}_x[\psi]$, model checking is based on Algorithm 3. The latter checks for bias violations via pairwise comparison of strategy-compatible paths under the equivalence relation $\sim_{\mathcal{NB}_x}$. Since $\sim_{\mathcal{NB}_x}$ depends only on some features of the paths, and the pairwise comparison is a quadratic problem, then this part can be performed in polynomial time. As all steps in both parts can be performed in polynomial time, we conclude that model checking for bATL is PTIME-complete. □

6 MODEL CHECKING BATL *

The model checking problem for bATL * extends that of ATL* by incorporating reasoning about bias. We apply a Nondeterministic Tree Büchi Automaton (NTBA) construction, which offers a natural way to capture both temporal evolution and strategic behaviour in MASs.

Theorem 6. *The model checking problem for* bATL * *in CGSs is decidable.*

Satisfaction sets for strategy quantified property $\langle\langle\Gamma\rangle\rangle\psi$ are defined as:

$$\mathrm{Sat}(\langle\langle\Gamma\rangle\rangle\psi) = \{s \in S \mid \exists\vec{\sigma_\Gamma}.\, \forall\pi \in \mathrm{Paths}_{\mathcal{G}}(s, \vec{\sigma_\Gamma}).\, (\mathcal{G}, \pi) \models \psi\}$$

This involves checking the intersection of two tree automata as described in [1]: one that accepts trees where all paths satisfy ψ and another that accepts trees corresponding to possible strategies of the coalition $\Gamma \subseteq$ Ag.

Thus, we focus on the model checking process for the new bias formula $\langle\langle\Gamma\rangle\rangle\mathfrak{B}_s[\psi]$. Similar to that for $\langle\langle\Gamma\rangle\rangle\psi$, the process involves checking the non-emptiness of the intersection of two tree automata: one that accepts trees where all paths satisfy $\mathfrak{B}_s[\psi]$ and

another that accepts trees corresponding to possible strategies of the coalition Γ. Since the latter automaton has the same definition as the one described in [1], we focus on the construction of the automaton that accepts trees where all paths satisfy $\mathfrak{B}_s[\psi]$, i.e., each tree accepted by the automaton has all paths that satisfy ψ and there are at least two paths π and π' that are not strongly equivalent to each other: $\pi \nsim_{\mathcal{NB}_s} \pi'$.

Furthermore, we need to identify a set $F_{\mathfrak{B}_s[\psi]}$ of nodes in the automaton $A_{\mathfrak{B}_s[\psi]}$ s.t. in each tree accepted by $A_{\mathfrak{B}_s[\psi]}$ there is at most one path reaching $F_{\mathfrak{B}_s[\psi]}$, or there exists two infinite paths (π, π') reaching $F_{\mathfrak{B}_s[\psi]}$ infinitely often iff: $((\mathcal{G}, \pi) \models \psi \wedge (\mathcal{G}, \pi') \models \psi) \Rightarrow (\pi \nsim_{\mathcal{NB}_s} \pi')$.

The formula ψ can be considered as an LTL formula over atomic propositions $\Sigma = 2^{\max(\psi)}$, where $\max(\psi)$ is the set of maximal state subformula of ψ. Note that we assume this set of atomic propositions because we apply the classic bottom-up approach, in which, after solving the innermost subformula, we replace it with the corresponding atomic proposition.

So we can construct a Nondeterministic Tree Büchi Automaton (NTBA).

Definition 13. *An NTBA A is a tuple $(\Sigma, Q, q^*, D, \Delta, F)$ where Σ is the alphabet, Q is a finite set of states, $q^* \in Q$ is the initial state, D is a finite set of directions, $\Delta : Q \times \Sigma \to \mathcal{B}^+(D \times Q)$ is a transition function, where $\mathcal{B}^+(D \times Q)$ is the set of all positive Boolean combinations of pairs (d, q) with d direction and q state, and $F \subseteq Q$ is the set of accepting states.*[1]

To well understand this class of automata, assume that A, being in a state q, is reading a node x of the input tree labelled by ξ. Assume also that $\Delta(q, \xi) = (((0, q_1) \wedge (1, q_1)) \vee (1, q_2))$. Then, there are two ways along which the construction of the run can proceed. In the first option, one copy of the automaton proceeds in direction 0 to state q_1 and one copy proceeds in direction 1 to state q_1. In the second option, one copy of A proceed in direction 1 to state q_2. Hence, $\vee$ and $\wedge$ in $\Delta(q, \xi)$ represent, respectively, choice and concurrency. A run is *accepting* if all infinite paths reach infinitely often accepting states. An input tree is accepted if there exists a corresponding accepting run. By $L(A)$ we denote the set of trees accepted by A. We say that A is not empty if $L(A) \neq \emptyset$.

Given the class of automata, we need to construct an NTBA $A_{\mathfrak{B}_s[\psi]} = (\Sigma, Q_{\mathfrak{B}_s[\psi]}, q^*_{\mathfrak{B}_s[\psi]}, D_{\mathfrak{B}_s[\psi]}, \Delta_{\mathfrak{B}_s[\psi]}, F_{\mathfrak{B}_s[\psi]})$ over alphabet Σ recognising trees in which every path satisfies ψ, and there exist two paths π, π' such that $\pi \nsim_{\mathcal{NB}_s} \pi'$. To construct such an automaton, we first need to formally introduce the concept of tree.

Definition 14. *Let Υ be a set. An* Υ-tree *is a prefix closed subset $T \subseteq \Upsilon^*$. The elements of T are called* nodes *and the empty word ε is the* root *of T. For $v \in T$, the set of* children *of v (in T) is $child(T, v) = \{v \cdot x \in T \mid x \in \Upsilon\}$. Given a node $v = y \cdot x$, with $y \in \Upsilon^*$ and $x \in \Upsilon$, we define $anc(v)$ to be y, i.e., the ancestors of v, and $last(v)$ to be x. We also say that v* corresponds *to x. The complete Υ-tree is the tree Υ^*. For $v \in T$, a (full) path π of T from v is a* minimal *set $\pi \subseteq T$ such that $v \in \pi$ and for each $v' \in \pi$ such that $child(T, v') \neq \emptyset$, there is exactly one node in $child(T, v')$ belonging to π. Note that every word $w \in \Upsilon^*$ can be thought of as a path in the tree Υ^*, namely the path containing all the prefixes of w. For an alphabet Σ, a Σ-labeled Υ-tree is a pair $< T, V >$ where T is an $\Upsilon-$tree and $V : T \to \Sigma$ maps each node of T to a symbol in Σ.*

[1]Note that, formulas are in disjunctive normal form, and in each conjunctive clause every direction appears at most once.

Our aim is to construct an automaton $A_{\mathfrak{B}_s[\psi]}$ whose vertex set is:

$$Q_{\mathfrak{B}_s[\psi]} = S \times Q_\psi \times \{\bot, \top\}$$

Note that with Q_ψ we assume the set of states of the automaton A_ψ that accepts all the trees satisfying ψ. Since this automaton as the same structure as the one described in [1], we do not describe it further. Thus, in the rest, we will focus in the solution of the $\mathfrak{B}$ operator.

Thus, the root of the accepting trees need to be the following:

$$q^*_{\mathfrak{B}_s[\psi]} = (s_I, q_0, \top)$$

For any state $q = (s, q_\psi, b)$ and label $\xi \in \Sigma$, define the transition set $\Delta(q, \xi)$ to include all valid successor configurations. For each possible tuple of actions $\alpha, \alpha' \in \mathrm{ACT}$:

- compute successors $t = \delta(s, \alpha)$ and $t' = \delta(s, \alpha')$
- compute new automaton state: $q_1 \in \Delta(q_\psi, \xi_s)$, where $\xi_s = \{\varphi \in \max(\psi) \mid (M, s) \models \varphi\}$
- update bit b: if $(t =_{\mathcal{NB}} t')$, then $b_1 = b$ else $b_1 = \bot$;
- add combinations of directions and successor states:

$$\bigvee_{\text{any valid d,d' and t, t'}} [(d, (t, q_1, b_1)) \wedge (d', (t', q_1, b_1))]$$

This encodes two synchronised paths progressing through the model and formula automata, tracking divergence of bias-sensitive equivalence.

Finally, we consider the accepting set as

$$F_{\mathfrak{B}_s[\psi]} = \{(s, q, \bot) \in Q_{\mathfrak{B}_s[\psi]} \mid q \in F_\psi\},$$

That is, accepts runs where both paths satisfy ψ (i.e., their automaton states visit accepting states infinitely often), and eventually diverge in bias-sensitive equivalence (i.e., $b = \bot$ is seen infinitely often).

A similar process can be applied to the weak bias operator, which only checks state equivalence for final states when the preceding states are bias-sensitive equivalent.

Theorem 7 (Complexity). *The model-checking* bATL * *can be done in 2EXPTIME-complete.*

Proof. For hardness, note that bATL * strictly extends ATL* by introducing bias-aware operators such as $\langle\langle\Gamma\rangle\rangle\mathfrak{B}_x[\psi]$, which impose additional constrains on strategy profiles based on fairness-based equivalence. Since ATL* model checking is already 2EXPTIME-hard, and bATL * subsumes ATL*, it follows immediately that bATL * is 2EXPTIME-hard. For membership, the model checking procedure for bATL * extends that of ATL* by evaluating formulae over strategy trees of exponential size. Evaluating nested path quantifiers and checking bias-awareness across strategy-compatible paths can be done in exponential time, but due to the exponential size of the strategy tree and potential nesting of modalities, the total runtime becomes double exponential in the size of the formula. Thus, bATL * model checking is 2EXPTIME-complete. □

7 RELATED WORK

Active research has investigated bias in AI systems from various perspectives, focusing on issues such as discriminatory outcomes and fairness in decision-making processes. For example, Burnett et al. [7] examined how agents form and communicate stereotypical reputations based on observed features and behaviours, facilitating the detection of biases in reputational opinions. Ryu et al. [23] proposed an approach that leverages biased action information to improve policy learning, achieving enhanced performance in mixed environments. Similarly, Alvim et al. [2] extended the DeGroot model to incorporate individual cognitive biases in social networks, illustrating how societies reach consensus or unanimity under such influences. While these works offer valuable insights, our approach diverges by introducing a formal framework for automatically reasoning about bias in system behaviours. Our novel notion of bias, based on the interference between bias-sensitive inputs and public outputs, is intuitive and general. Additionally, our bias policy, built upon equivalence classes, provides a structured and robust foundation for systematic analysis. Such formulation is foundational, as it rigorously formalises state equivalence - a critical requirement for fairness in dynamic and interactive models. Simpler definitions, such as identifying bias through paths where biased states affect outcomes, may seem intuitive but hard to capture the nuanced manifestations of bias in dynamic multiagent systems, particularly when state equivalence is overlooked. Drawing inspiration from the concept of non-interference, our use of non-bias-equivalent behaviours ensures that bias-sensitive attributes do not unfairly influence outcomes, fairness across equivalence classes within MASs.

Our work also relates to logical verification techniques in multi-agent systems. We propose bATL with bias operators to specify and reason about our notion of bias. Numerous extensions of the ATL family have been developed for reasoning about properties in MASs. For instance, PATL [12, 16] incorporates probabilistic operators to reason about quantitative behaviours, while rPATL [19] includes quantified reward formulae to address reward-based reasoning. oPATL [21] introduce operators for (quantitative) opacity and observability analysis of agents. Beyond probabilistic and reward-based operators, a number of logics have been proposed to account for different dimensions of agency. For example, RB-ATL [22] and RAB-ATL [8] address bounded rationality by incorporating explicit resource constraints, while OL [9] and its extensions [10, 11] enrich the verification framework to model obstruction-based reasoning. Similarly, ATLF [14] integrates fuzzy operators to reason about uncertainty, NatATL [17, 18] and NatSL [5] capture the use of natural strategies, and Cap-ATL [4] incorporates capacity constraints. These contributions underline the richness of the ATL family, which continues to evolve to address diverse reasoning requirements in multi-agent systems. However, our bATL framework introduces new bias operators specifically designed to assess bias induced by AI system behaviours during HAI interactions. This unique focus allows us to rigorously analyse and potentially mitigate bias in a structured and formalised manner, contributing the field of bias detection and correction in AI systems. Additionally, [20] assessed AI system fairness through a logical perspective, formalising key criteria like skewness and dependency on data, and defining metrics for group and individual fairness. Furthermore, [13] introduces a method for the quantitative analysis of fairness in AI systems, using the BRIO tool to assess social unfairness and ethically undesirable behaviours, with a particular focus on credit scoring applications. While related, our work differs by introducing formal verification with bATL, which operates within an alternating-time logic framework and includes a model checking algorithm tailored for MAS-based AI systems. This framework is specifically designed for reasoning about bias in agent interactions, with dynamic and interactive settings and providing a formal approach for rigorous bias analysis.

8 CONCLUSIONS AND FUTURE WORKS

In this paper, we have introduced a novel notion of bias and a corresponding bias policy for MASs, formalised through the bATL logic framework upon a game structure model. We also presented verification precess for reasoning about bias within this framework.

For future work, we plan to extend our framework to incorporate quantitative measures of bias. Developing metrics and algorithms to quantify bias across different contexts and scenarios would enhance the practical applicability of our approach. Additionally, we aim to investigate how strategic interactions among multiple agents affect bias, integrating strategic elements into our logical framework to better capture the complexities of decision-making in competitive and cooperative environments. Lastly, we plan to extend our formal bias analysis to evaluate large language models (LLMs) in interactive settings. By modelling LLM-driven systems as decision-tree prompts within our bATL framework, we can systematically identify and verify bias-related properties using model-checking techniques. This approach enables dynamic analysis of LLM behaviours, pinpointing where bias-sensitive attributes improperly influence outcomes. The insights gained can inform targeted mitigation strategies, such as refining prompts or adjusting fine-tuning processes, ensuring fairness and transparency in LLM interactions across diverse applications.

ACKNOWLEDGMENTS

The authors acknowledge the support of the National Decommissioning Centre (NDC) project on “Decision-Making for Oil & Gas Decommissioning: A Formal AI-Driven Approach”.

REFERENCES

[1] R. Alur, T. A. Henzinger, and O. Kupferman. 2002. Alternating-timeTemporal Logic. *J. ACM* 49, 5 (2002), 672–713.

[2] Mário S. Alvim, Artur Gaspar da Silva, Sophia Knight, and Frank Valencia. 2024. A Multi-agent Model for Opinion Evolution in Social Networks Under Cognitive Biases. In *FORTE (Lecture Notes in Computer Science, Vol. 14678)*, Valentina Castiglioni and Adrian Francalanza (Eds.). Springer, 3–19.

[3] Christel Baier and Joost-Pieter Katoen. 2008. *Principles of Model Checking (Representation and Mind Series).*

[4] Gabriel Ballot, Vadim Malvone, Jean Leneutre, and Youssef Laarouchi. 2024. Strategic Reasoning under Capacity-constrained Agents. In *Proceedings of the 23rd International Conference on Autonomous Agents and Multiagent Systems, AAMAS 2024, Auckland, New Zealand, May 6-10, 2024*, Mehdi Dastani, Jaime Simão Sichman, Natasha Alechina, and Virginia Dignum (Eds.). International Foundation for Autonomous Agents and Multiagent Systems / ACM, 123–131. https://doi.org/10.5555/3635637.3662859

[5] Francesco Belardinelli, Wojtek Jamroga, Vadim Malvone, Munyque Mittelmann, Aniello Murano, and Laurent Perrussel. 2022. Reasoning about Human-Friendly Strategies in Repeated Keyword Auctions. In *21st International Conference on*

Autonomous Agents and Multiagent Systems, AAMAS 2022, Auckland, New Zealand, May 9-13, 2022, Piotr Faliszewski, Viviana Mascardi, Catherine Pelachaud, and Matthew E. Taylor (Eds.). International Foundation for Autonomous Agents and Multiagent Systems (IFAAMAS), 62–71. https://doi.org/10.5555/3535850.3535859

[6] Francesco Belardinelli, Alessio Lomuscio, Aniello Murano, and Sasha Rubin. 2018. Alternating-time Temporal Logic on Finite Traces. In *IJCAI*, Jérôme Lang (Ed.). 77–83.

[7] Chris Burnett, Timothy J. Norman, and Katia P. Sycara. 2013. Stereotypical trust and bias in dynamic multiagent systems. *ACM Trans. Intell. Syst. Technol.* 4, 2 (2013), 26:1–26:22.

[8] Davide Catta, Angelo Ferrando, and Vadim Malvone. 2024. Resource Action-Based Bounded ATL: A New Logic for MAS to Express a Cost Over the Actions. In *PRIMA 2024: Principles and Practice of Multi-Agent Systems - 25th International Conference, Kyoto, Japan, November 18-24, 2024, Proceedings (Lecture Notes in Computer Science, Vol. 15395)*, Ryuta Arisaka, Víctor Sánchez-Anguix, Sebastian Stein, Reyhan Aydogan, Leon van der Torre, and Takayuki Ito (Eds.). Springer, 206–223. https://doi.org/10.1007/978-3-031-77367-9_16

[9] Davide Catta, Jean Leneutre, and Vadim Malvone. 2023. Obstruction Logic: A Strategic Temporal Logic to Reason About Dynamic Game Models. In *ECAI 2023 - 26th European Conference on Artificial Intelligence, September 30 - October 4, 2023, Kraków, Poland - Including 12th Conference on Prestigious Applications of Intelligent Systems (PAIS 2023) (Frontiers in Artificial Intelligence and Applications, Vol. 372)*, Kobi Gal, Ann Nowé, Grzegorz J. Nalepa, Roy Fairstein, and Roxana Radulescu (Eds.). IOS Press, 365–372. https://doi.org/10.3233/FAIA230292

[10] Davide Catta, Jean Leneutre, Vadim Malvone, and Aniello Murano. 2024. Obstruction Alternating-time Temporal Logic: A Strategic Logic to Reason about Dynamic Models. In *Proceedings of the 23rd International Conference on Autonomous Agents and Multiagent Systems, AAMAS 2024, Auckland, New Zealand, May 6-10, 2024*, Mehdi Dastani, Jaime Simão Sichman, Natasha Alechina, and Virginia Dignum (Eds.). International Foundation for Autonomous Agents and Multiagent Systems / ACM, 271–280. https://doi.org/10.5555/3635637.3662875

[11] Davide Catta, Jean Leneutre, Vadim Malvone, and James Ortiz. 2025. Coalition Obstruction Temporal Logic: A New Obstruction Logic to Reason About Demon Coalitions. In *Proceedings of the Thirty-Fourth International Joint Conference on Artificial Intelligence, IJCAI 2025, Montreal, Canada, August 16-22, 2025*. ijcai.org, 21–28. https://doi.org/10.24963/IJCAI.2025/3

[12] T. Chen and J. Lu. 2007. Probabilistic Alternating-time Temporal Logic and Model Checking Algorithm. In *FSKD*. IEEE Computer Society, 35–39.

[13] G. Coraglia, F. A. Genco, P. Piantadosi, E. Bagli, P. Giuffrida, D. Posillipo, and G. Primiero. 2024. Evaluating AI fairness in credit scoring with the BRIO tool. *CoRR* abs/2406.03292 (2024).

[14] Angelo Ferrando, Giulia Luongo, Vadim Malvone, and Aniello Murano. 2024. Theory and Practice of Quantitative ATL. In *PRIMA 2024: Principles and Practice of Multi-Agent Systems - 25th International Conference, Kyoto, Japan, November 18-24, 2024, Proceedings (Lecture Notes in Computer Science, Vol. 15395)*, Ryuta Arisaka, Víctor Sánchez-Anguix, Sebastian Stein, Reyhan Aydogan, Leon van der Torre, and Takayuki Ito (Eds.). Springer, 231–247. https://doi.org/10.1007/978-3-031-77367-9_18

[15] J. A. Goguen and J. Meseguer. 1982. Security Policies and Security Models. In *S & P*. 11–20.

[16] X. Huang, K. Su, and C. Zhang. 2012. Probabilistic Alternating-Time Temporal Logic of Incomplete Information and Synchronous Perfect Recall. In *AAAI*, Jörg Hoffmann and Bart Selman (Eds.). AAAI Press, 765–771.

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

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

[19] M. Kwiatkowska, G. Norman, D. Parker, and G. Santos. 2021. Automatic verification of concurrent stochastic systems. *Formal Methods in System Design* 58, 1 (2021), 188–250.

[20] C. Manganini and G. Primiero. 2023. Reasoning With Bias. In *Proceedings of the 1st Workshop on Fairness and Bias in AI co-located with ECAI (CEUR Workshop Proceedings, Vol. 3523)*. CEUR-WS.org.

[21] C. Mu and J. Pang. 2023. On Observability Analysis in Multiagent Systems. In *ECAI (Frontiers in Artificial Intelligence and Applications, Vol. 372)*. IOS Press, 1755–1762.

[22] Hoang Nga Nguyen, Natasha Alechina, Brian Logan, and Abdur Rakib. 2018. Alternating-time temporal logic with resource bounds. *J. Log. Comput.* 28, 4 (2018), 631–663. https://doi.org/10.1093/LOGCOM/EXV034

[23] Heechang Ryu, Hayong Shin, and Jinkyoo Park. 2021. Cooperative and Competitive Biases for Multi-Agent Reinforcement Learning. In *AAMAS*. ACM, 1091–1099.