On Observability Analysis in Multiagent Systems

Chunyan Mu[a;*] **and Jun Pang**[b]

[a]Department of Computing Science, University of Aberdeen
[b]Department of Computer Science, University of Luxembourg

Abstract. In multiagent systems (MASs), agents' observation upon system behaviours may improve the overall team performance, but may also leak sensitive information to an observer. A quantified observability analysis can thus be useful to assist decision-making in MASs by operators seeking to optimise the relationship between performance effectiveness and information exposure through observations in practice. This paper presents a novel approach to quantitatively analysing the observability properties in MASs. The concept of opacity is applied to formally express the characterisation of observability in MASs modelled as partially observable multiagent systems. We propose a temporal logic oPATL to reason about agents' observability with quantitative goals, which capture the probability of information transparency of system behaviours to an observer, and develop verification techniques for quantitatively analysing such properties. We implement the approach as an extension of the PRISM model checker, and illustrate its applicability via several examples.

1 Introduction

The multiagent computing paradigm pervades nearly all aspects of the modern intelligent computational world, enabling the creation of net-based solutions to communication, collaboration, and coordination problems in different fields such as commerce, cyber, and conflict prevention. Agents often exploit machine learning methods, which allow them to learn from experience, and to implement decision-making mechanisms. Observation of other agents' behaviours may improve the overall team performance in the learning mechanisms [26]. On the other hand, in practice, due to the frequently adversarial nature of multiagent systems (MASs) such solutions can also bring additional channel threat and information leakage risks. Sensitive information can be leaked to malicious (inside/outside) agents during the process of collaboration and interaction. Information exposure issue should also play a role in making decisions for agents. Therefore, rigorous analysis and verification of (sensitive) information transparency properties constitutes an important challenge. In particular, a *quantified observability* analysis can be useful in MASs design to address such concerns, for instance, for decision-making by operators seeking to optimise the relationship between performance effectiveness and information exposure security risks in MASs, which are the key underpinning elements of a progressive artificially intelligent society.

This paper addresses the problem of specifying, verifying and thus reasoning about observability properties of MASs. Specifically, we specify the observability properties from novel perspective of information transparency in the *opacity* framework, which is formally described in the logic oPATL. With this logic, we can express the degree of transparency of system behaviours to an observer under a coalition of agents' strategy, given predefined observability of atomic actions to the observer. We model the system in partially observable probabilistic game structure, which maps infinite (input) sequences onto partially observable infinite (output) sequences. The properties of observability can then be captured by measurement upon output sequences and input sequences. Intuitively, a transparent system, in which the observability is maximised, reveals most information in the input sequence; while an opaque system, in which the observability is minimised, hides some information (with properties of interest) contained in the input sequence. Probabilistic model checking techniques can be applied to reason about the quantitative observability analysis of the system, and allow us to calculate the degree of the observability of the system behaviours.

The main contributions of the paper are summarised below:

- A partially observable multi-agent system (POMAS) is proposed to model probabilistic action outcomes of system behaviours with characterisation of multi-agents, actions and the relevant observables, and atomic state propositions.
- The logic of oPATL is presented to allow us to express (probabilistic) observability properties.
- Probabilistic verification technique against oPATL is presented to allow for automatic verification of quantified observability properties in MASs modelled as POMAS.
- A prototype of the proposed framework is built upon the PRISM model checker [22].

Related work. In the field of formal methods for artificial intelligence, logics have gained a great importance in expressing properties and providing powerful formalisms for reasoning about agents behaviours in MASs. There have been several multiagent logics proposed to express and reason about agents' observation properties including [19, 5, 12, 17]. These logics have centred on knowledge representation where knowledge is built from what the agents observe. In these logics, the formation of knowledge is modelled via epistemic connectives which can be defined as modal operators of the form $K_i\,\varphi$ specifying "agent i knows property φ", and the observability of agents is modelled via Kripkean accessibility relations with respect to the visibility atoms of propositional variables: agent i cannot distinguish valuation w from w' if every variable that agent i observes has the same value at w and w'. A number of works [18, 17, 31] have studied multiagent planning model to adapt strategy for cooperation and analyse trade-off between local observation and capability of coordination based on estimation of quantified communicating and variant costs. Various methods and accompanying implementations

* Corresponding Author. Email: chunyan.mu@abdn.ac.uk

have also been developed supporting the verification against the logics and their variations [1, 24, 6, 23, 7]. This line of works focused on epistemic logics regarding the knowledge about the *state* of the system, mostly used for producing epistemic planning. It is not natural to apply these approaches to investigate (sensitive) information flow caused by observation and inference. In addition, Huang et al. [20] proposed PATL*, which enables reasoning about MASs with incomplete information. While PATL* can be applied to reason about the possible states of the system and the possible actions of other agents based on their beliefs and strategies, it does not directly address issues related to information exposure resulting from observation of agents' actions and behaviours for flow security concern.

In contrast, this work opens a novel perspective from information transparency based on the concept of *opacity*. It builds upon the frameworks of PATL [14] and probabilistic opacity [11, 29] to investigate the formal expression of agents' observations about the information of actions taken by other agents. We are concerned with studying the effects of agents' behaviours on the overall system, particularly in terms of potential information leakage. Actions are a natural choice for representing the relations between states, which is why we focused on observations over actions rather than over states. However, the distinction between observations over states and actions is often blurred, and the choice between the two depends on the specific research context and goals. Ultimately, our choice of using PATL was driven by its suitability for modelling complex interactions among multiple agents and their strategies, as well as its ability to reason about the effects of actions on the system over time. Intuitively, a concerned behaviour (satisfying a property φ, e.g., reaching a secret state) of a system is considered as *opaque* if, whenever the behaviour has occurred, there is a non-concerned behaviour (violating property φ) that is observationally equivalent. Opacity represents a suitable option for specifying observation and information flow properties in MASs due to the feature of partial observability of agents, agent behaviours, the uncertainty of the environment, and the nature of information. Observation in our work is considered as a modal operator, based on the concept of opacity, with more intuitive semantics. Our approach can capture the information induced/obtained via inference, a direct application is privacy loss/information leakage analysis and assessment.

As a consequence, this work also relates to information flow security awareness analysis and verification. Over the past years there has been a sustained effort in exploring concepts and analyses in quantified information flow for secure computing systems. Indeed, this period has seen significant inroads made into the study of core imperative languages and their probabilistic aspects [15, 21, 25, 29]; and some attempts to study quantified approaches to flow security of system specifications in various interactive settings [3, 2, 8, 10, 28, 29]. However, none of these studies has accounted for multiple agent scenarios, which involve dynamic patterns of collaborations, interactions, and decision-makings. In contrast, in this work we study the observability issues which can be naturally applied to quantified information flow security awareness in MASs, from a novel perspective of information transparency.

2 Partially Observable Multiagent Systems

Let $\mathbb{N}$ be the set of natural numbers with zero, $\mathsf{Ag} = \{1, 2, \ldots, n\}$ be a set of agents. An *alphabet* Σ is a non-empty, finite set of actions, $|\Sigma|$ is its cardinality. Σ^* denotes the set of all *finite* words over Σ including the *empty word* ε, $\Sigma^+ = \Sigma^* \setminus \{\varepsilon\}$, Σ^ω denotes the set of all *infinite* words, Σ^∞ denotes the set of all *finite* and *infinite* words. Subsets $L \subseteq \Sigma^*$ are called languages, and $L \subseteq \Sigma^\infty$ are called ω-languages. Let $\mathsf{Dist}(X)$ denote the set of discrete probability distribution over a set X, i.e., all functions $\mu : X \to [0, 1]$ s.t. $\sum_{x \in X} \mu(x) = 1$ and $\mu(x) \geq 0$. 2^X denotes the power set of X.

2.1 Probabilistic (concurrent) game structure

Definition 1. *A* probabilistic game structure (PGS) *is a tuple* $\mathcal{G} = (S, \mathsf{Act}, \delta)$*, where:*

- S *is a finite set of* states*;*
- $\mathsf{Act} = \mathsf{Act}_1 \times \mathsf{Act}_2 \times \ldots \mathsf{Act}_n = \prod_{j \in \mathsf{Ag}} \mathsf{Act}_j$ *is a finite set of joint actions (decisions) of the agents in* Ag*,* $\mathsf{Act}_j \subseteq \Sigma$ *is the set of actions that* $j \in \mathsf{Ag}$ *can perform;*
- $\delta : S \to 2^{\mathsf{Dist}(\mathsf{Act} \times S)}$ *is the* probabilistic transition relation*; for state* $s \in S$*,* $\delta(s)$ *is the distribution for next state;* s *is a* terminal state *if* $\delta(s) = \emptyset$*.*

We write $s \to \mu$ for $s \in S$ and $\mu \in \delta(s)$. Each agent $j \in \mathsf{Ag}$ chooses action a_j in state $s \in S$, we write $s \xrightarrow{\prod_{j \in \mathsf{Ag}} a_j} s'$ and sometimes $s \xrightarrow{p.\, \prod_{j \in \mathsf{Ag}} a_j} s'$ for $s, s' \in S$ whenever $s \to \mu$ and $\mu(s, \prod_{j \in \mathsf{Ag}} a_j) > 0$, where p denotes the probability of the transition from s to s' through joint action $\prod_{j \in \mathsf{Ag}} a_j$. We use nondeterministic PGS in this paper to accommodate the agents' probabilistic behaviour. While the game structure determines the probability of each action, agents can still make decisions based on their probabilistic strategies or beliefs. To enable a broader range of strategies, an MDP-like transition function that maps a state-action pair to a distribution over the next state would provide greater flexibility for agent behaviour. This extension is a potential area for future work.

Definition 2. *We say* $\mathcal{G}$ *is* circular*, if every state has an outgoing transition, i.e., for all* $s \in S$*, there is* $s \xrightarrow{\prod_{j \in \mathsf{Ag}} a_j} s'$*. We say* $\mathcal{G}$ *is* fully probabilistic *if* $|\delta(s)| \leq 1$ *for all* $s \in S$*. For a fully probabilistic game structure, when* $\delta(s) \neq \emptyset$*, we use* $\delta(s)$ *to denote the distribution outgoing from* s*.*

Definition 3. *A* path *in* $\mathcal{G}$ *is a sequence* $\rho = s_0 \xrightarrow{\alpha_0} s_1 \xrightarrow{\alpha_1} \ldots$ *of states and joint actions, where* $\alpha_i = \prod_{j \in \mathsf{Ag}} a^i_j \in \mathsf{Act}$*,* $a^i_j \in \mathsf{Act}_j(s_i)$ *for* $i \geq 0$ *and* $j \in \mathsf{Ag}$*, for all* $t \geq 0$*,* $s_t \in S$*,* $\alpha_t \in \mathsf{Act}$ *and* $\delta(s_t \xrightarrow{\alpha_t} s_{t+1}) > 0$*. Let* $\rho_s(i)$ *denote the* i^{th} *state of* ρ*, and* $\rho_a(i)$ *denote the* i^{th} *joint action of* ρ*, so for all* i*, we have* $\rho_s(i) \xrightarrow{\rho_a(i)} \rho_s(i+1)$*. Let* ρ^i *denote the prefix of* ρ *up to the* i^{th} *state, i.e.,* $\rho^i = s_0 \xrightarrow{\alpha_0} s_1 \xrightarrow{\alpha_1} \ldots \xrightarrow{\alpha_{i-1}} s_i$*. Let* $\mathsf{Post}(s)$ *denote immediate state successors of* s *in a path, and* $\mathsf{Pre}(s)$ *denote the immediate state predecessors of* s *in a path. A path is* finite *if it ends with a state. A path is* complete *if it is either infinite or finite ending in a terminal state. Given a finite path* ρ*,* $last(\rho)$ *denotes its last state. The length of a path* ρ*, denoted by* $|\rho|$*, is the number of transitions appearing in the path. Let* $\mathsf{Paths}_\mathcal{G}(s)$ *denote the set of* $\mathcal{G}$*-paths,* $\mathsf{Paths}_\mathcal{G}(s)^*$ *denote the set of all* $\mathcal{G}$*'s finite paths,* $\mathsf{CPaths}_\mathcal{G}(s)$ *denote the set of all* $\mathcal{G}$*'s complete paths, starting from state* s*. Paths are ordered by the prefix relations, denoted by* $\leq$*:* $\mathsf{Pref}(\rho') = \{\rho \mid \rho \leq \rho'\}$*.*

Definition 4. *The* trace *of a path is the sequence of joint actions in* $\mathsf{Act}^* \cup \mathsf{Act}^\infty$ *obtained by erasing the states, so for the above* ρ*, we have the corresponding trace of* ρ*:* $tr(\rho) = \alpha_0 \alpha_1 \ldots$*. We use* $\mathsf{Traces}_\mathcal{G}(s)$ *to denote the set of* $\mathcal{G}$*-traces starting from state* s*.*

Let $\mathcal{G} = (S, \mathsf{Act}, \delta)$ be a PGS, $\rho \in \mathsf{Paths}_\mathcal{G}(s)^*$ be a finite path starting from $s \in S$. The *cone* generated by ρ is the set of complete

paths $\langle\rho\rangle = \{\rho' \in \mathsf{CPaths}_{\mathcal{G}}(s) \mid \rho \leq \rho'\}$. Given a $\mathcal{G} = (S, \mathsf{Act}, \delta)$ and a state $s \in S$, we can then calculate the probability value, denoted by $\mathbb{P}_s(\rho)$, of any finite path ρ starting at s as follows:

- $\mathbb{P}_s(s) = 1$, and
- $\mathbb{P}_s(\rho \xrightarrow{\alpha} s') = \mathbb{P}_s(\rho)\mu(s', \alpha)$ for $last(\rho) \to \mu$.

Let $\Omega_s \triangleq \mathsf{CPaths}_{\mathcal{G}}(s)$ be the sample space, and let $\mathcal{G}_s$ be the smallest σ-algebra induced by the cones generated by all the finite paths of $\mathcal{G}$. Then $\mathbb{P}$ induces a unique *probabilistic measure* on $\mathcal{G}_s$ such that: $\mathbb{P}_s(\langle\rho\rangle) = \mathbb{P}_s(\rho)$ for every finite path ρ starting in s.

2.2 Observations

To model the observability of agents, we need to make a distinction between the actions that are observable and those that are not, regarding different agents' view. For each agent, we use a set of *observables*, distinct of the *actions* of the ambient PGS. *Actions* and *observables* are connected by an observation function.

Definition 5. *Let Θ be a finite alphabet for observables, and $\Theta^{\epsilon} = obs \cup \{\epsilon\}$ where ϵ denotes the invisible/hidden action. An* observation function *on paths is a labelled-based function $obs : \mathsf{Paths}_{\mathcal{G}}(s) \to (\Theta_1 \times \Theta_2 \times \cdots \times \Theta_n)^{\infty}$, where $\Theta_j \subseteq \Theta$ denotes a finite set of observables for $j \in \mathsf{Ag}$. Specifically, we consider static observation function, i.e., there is a map $\zeta : \mathsf{Act} \to \Theta^{\epsilon}$ s.t. for every path $\rho = s_0 \xrightarrow{\alpha_0} s_1 \xrightarrow{\alpha_1} \ldots \xrightarrow{\alpha_{t-1}} s_t$ of $\mathcal{G}$: $obs(\rho) = \beta_0\ \beta_1\ \ldots\ \beta_{t-1}$. where for all $0 \leq i < t$, $\alpha_i = \prod_{j \in \mathsf{Ag}} a^i_j$, and $\beta_i = \prod_{j \in \mathsf{Ag}} \zeta(a^i_j)$. Observation functions on traces are defined similarly.*

2.3 Partially observable MASs

Definition 6. *A* partially observable multiagent system (POMAS) *is a tuple $\mathcal{M} = (\mathsf{Ag}, \mathcal{G}, s_0, \mathsf{Ap}, L, \{obs_i\}_{i \in \mathsf{Ag}})$, where:*

- $\mathsf{Ag} = \{1, \ldots, n\}$ *is a finite set of intelligent* agents*;*
- $\mathcal{G} = (S, \mathsf{Act}, \delta)$ *is a* fully PGS *that is circular;*
- $s_0 \in S$ *is the* initial state*;*
- Ap *is a finite set of* atomic propositions*;*
- $L : S \to 2^{\mathsf{Ap}}$ *is the state labelling function mapping each state to a set of atomic state proposition taken from set* Ap*;*
- $obs_i : \mathsf{Paths}_{\mathcal{G}}(s_0) \to (\Theta_1 \times \cdots \times \Theta_n)^{\infty}$ *is an observation function for agent $i \in \mathsf{Ag}$.*

Actions	ζ_1(Action)	ζ_2(Action)	ζ_3(Action)	Descriptions
$\mathtt{Op}_0$	$\mathtt{Op}_0$	$\mathtt{Op}_0$	$\mathtt{Op}_0$	the chair opens a voting session
$\mathtt{Cl}_0$	$\mathtt{Cl}_0$	$\mathtt{Cl}_0$	$\mathtt{Cl}_0$	the chair closes a voting session
$\mathtt{W}_i$	$\mathtt{W}$	$\mathtt{W}$	$\mathtt{W}$	agent i is waiting, $i \in \{0,1,2,3\}$
$\mathtt{V}_1^X$	X_1	ϵ	X	voter 1 votes candidate X
$\mathtt{V}_1^Y$	Y_1	ϵ	Y	voter 1 votes candidate Y
$\mathtt{V}_2^X$	ϵ	X_2	X	voter 2 votes candidate X
$\mathtt{V}_2^Y$	ϵ	Y_2	Y	voter 2 votes candidate Y
$\mathtt{V}_3^X$	ϵ	ϵ	X_3	voter 3 votes candidate X
$\mathtt{V}_3^Y$	ϵ	ϵ	Y_3	voter 3 votes candidate Y

Table 1: Actions and observation functions in Example 1.

Example 1. *In a toy agent-based model of voting, the process goes as follows: 1) The chair initiates the voting procedure; 2) Voters simultaneously propose their votes; 3) Each voter commits his vote once he makes his decision; 4) If a voter is waiting during this process, he can partially observe the behaviour of other voters and gather indications, such as identifying the dominant candidate; 5) Based on their observations, the voter can make their own voting decision; 6) Once all voters have committed their votes, the chair closes the voting session. The set of agents $\mathsf{Ag} = \{0,1,2,3\}$ includes a set of voters $\{1,2,3\}$ and the chair 0. We assume there are two candidates X and Y. The actions and the observation function are described in Table 1, where the column ζ_i(Action) specifies the assumed observation function over Action under voter i's view, action $\mathtt{Op}_0$ ($\mathtt{Cl}_0$) denotes the chair opens (closes) a voting session, $\mathtt{W}_i$ denotes agent $i \in \{0,1,2,3\}$ is waiting, $\mathtt{V}_j^X$ ($\mathtt{V}_j^Y$) denotes voter $j \in \{1,2,3\}$ votes candidate X(Y).*

Assume voter 2 is the observer, voters 1, 2, 3 vote consequently, e.g., voter 2 and 3 are waiting when voter 1 is voting. Table 1 indicates that the actions of voting X by 1 and 3 are not visible to voter 2. Consider a path (with probability of actions) where e.g., voters1, 2 and 3 vote candidate X with probability $\frac{1}{2}$, $\frac{1}{2}$ and $\frac{1}{3}$, respectively:

$$\rho \quad = \quad s_0 \xrightarrow{\mathtt{Op}_0\mathtt{W}_1\mathtt{W}_2\mathtt{W}_3} s_1 \xrightarrow{\frac{1}{2}.\mathtt{W}_0\mathtt{V}_1^X\mathtt{W}_2\mathtt{W}_3} s_2 \xrightarrow{\frac{1}{2}.\mathtt{W}_0\mathtt{W}_1\mathtt{V}_2^Y\mathtt{W}_3} s_3 \xrightarrow{\frac{2}{3}.\mathtt{W}_0\mathtt{W}_1\mathtt{W}_2\mathtt{V}_3^Y} s_4 \xrightarrow{\mathtt{Cl}_0\ \mathtt{W}_1\mathtt{W}_2\mathtt{W}_3} s_5$$

the observation and its probability on the above path from voter 2's view would be:

$$obs_2(\rho) \quad = \quad \mathtt{Op}_0\mathtt{WWW}\ \mathtt{WWW}\ \mathtt{WW}Y_2\mathtt{W}\ \mathtt{WWW}\ \mathtt{Cl}_0\mathtt{WWW}\ \ w.p.\ \frac{1}{6}.$$

Consider another example, assume voter 3 is the observer, voters $1,2$ make their voting concurrently, and voter 3 makes her voting afterwards. The observer's knowledge obtained from her observation might influence her decision on voting. Consider the following path:

$$\rho = s_0 \xrightarrow{\mathtt{Op}_0\mathtt{W}_1\mathtt{W}_2\mathtt{W}_3} s_1 \xrightarrow{\frac{1}{4}.\mathtt{W}_0V_1^XV_2^Y\mathtt{W}_3} s_2 \xrightarrow{\frac{1}{3}.\mathtt{W}_0\mathtt{W}_1\mathtt{W}_2V_3^X} s_3 \xrightarrow{\mathtt{Cl}_0\mathtt{W}_1\mathtt{W}_2\mathtt{W}_3} s_4$$

the observation on the above path from voter 3's view would then be: $obs_3(\rho) = \mathtt{Op}_0\mathtt{WWW}\ \mathtt{W}XY\mathtt{W}\ \mathtt{WWW}X_3\ \mathtt{Cl}_0\mathtt{WWW}\ w.p.\ \frac{1}{12}$.

2.4 Strategies for agents in POMASs

Given a POMAS $\mathcal{M} = (\mathsf{Ag}, \mathcal{G}, s_0, \mathsf{Ap}, L, \{obs_i\}_{i \in \mathsf{Ag}})$, a *mixed strategy* of an agent $i \in \mathsf{Ag}$ specifies a way of choosing actions, based on her observation on the finite path starting with s_0 so far.

Definition 7. *A* mixed strategy *for agent i is a function π_i:*

$$\pi_i \triangleq obs_i(\mathsf{Paths}_{\mathcal{G}}(s_0)) \to \mathsf{Dist}(\mathsf{Act}_i)$$

such that, if $\pi_i(\rho)(a_i) > 0$ then $a_i \in \mathsf{Act}_i(last(\rho))$. The set of all strategies of agent i is denoted Π_i.

Definition 8. *A* strategy profile *for* POMAS$\mathcal{M}$ *is a tuple $\pi = (\pi_1, \ldots, \pi_n) \in \Pi_1 \times \cdots \times \Pi_n$ producing a strategy for each agent of the system.*

Definition 9. *A path ρ is* consistent *with a strategy profile π, denoted by ρ_π, if it can be obtained by extending its prefixes using π. Formally, $\rho = s_0 \xrightarrow{\prod_{j \in \mathsf{Ag}} a_{0j}} s_1 \xrightarrow{\prod_{j \in \mathsf{Ag}} a_{1j}} \ldots$ is consistent with π if for all $t \geq 0$, $i \in \mathsf{Ag}$, under strategy π_i, we have: $a_i^t \in \mathsf{Act}_i(\rho_s(t))$ and $\delta(s_t \xrightarrow{\prod_{j \in \mathsf{Ag}} a_j^t} s_{t+1}) > 0$.*

Definition 10. *Given a* POMAS $\mathcal{M} = (\mathsf{Ag}, \mathcal{G}, s_0, \mathsf{Ap}, L, \{obs_i\}_{i \in \mathsf{Ag}})$, *a* history *is a finite path starting with s_0, the set of histories in $\mathcal{M}$ is written as $\mathsf{Hist}(\mathcal{M})$ and the set of histories in $\mathcal{M}$ starting with history h is written as $\mathsf{Hist}(\mathcal{M}, h)$. For any agent $i \in \mathsf{Ag}$, and two histories h and h', we say h and h' are observationally equivalent to each other from i's view, denoted by $h \sim_i h'$, iff $obs_i(h) = obs_i(h')$.*

Example 2. *Consider the second scenario proposed in Example 1. The observer's information of knowledge obtained from her observation might influence her decision on voting. Assume the observer (voter 3) is not able to see whom other voters have voted, but she can see how many ballots each candidate has received as specified in Table 1 and thus she can indicate the dominant candidate so far. A basic strategy to reflect such an influence is that she will vote the dominant candidate if there is one, otherwise she will vote the candidates under her preferred distribution.*

3 Observability Specification

This section studies the problem of formally specifying observability of an agent on system behaviours modelled in POMAS.

3.1 Observability and opacity

Given a property φ and an observation function obs_i of an agent $i \in \mathsf{Ag}$, we are interested in quantitatively expressing the observability of the agent that a set of agents has a strategy to enforce the property φ. The property can be viewed as a predicate, i.e., a set of execution paths for which it holds. The concept of *Opacity* [27] provides an intuitive approach for this task via distinguishing the observed behaviour and the original one. Intuitively, a property φ is opaque (not observable), provided that for every behaviour (say path ρ) satisfying φ there is another behaviour (say path ρ'), not satisfying φ, such that ρ and ρ' are observationally equivalent. So the observer is not able to determine whether the property in a given path of the system is satisfied or not. More precisely, opacity specifies whether an agent can establish a property φ, enforced by a strategy of a coalition A of agents, at some specific state(s) of the executions of the system, according to her observation on the system behaviours. We use $[\![\varphi]\!]$ to denote the set of paths satisfying property φ.

Definition 11. *Let $\mathcal{M} = (\mathsf{Ag}, \mathcal{G}, s_0, \mathsf{Ap}, L, \{obs_i\}_{i\in\mathsf{Ag}})$. Given a predicate φ over $\mathsf{Paths}_{\mathcal{G}}(s_0)$, we say φ is* opaque *w.r.t. obs_i if for every path $\rho \in [\![\varphi]\!]$, there is a path $\rho' \in [\![\bar{\varphi}]\!]$ s.t. $obs_i(\rho) = obs_i(\rho')$, i.e., all paths satisfying φ are covered by paths in $[\![\bar{\varphi}]\!]$: $obs_i([\![\varphi]\!]) \subseteq obs_i([\![\bar{\varphi}]\!])$ under obs_i, where $[\![\bar{\varphi}]\!] \triangleq \mathsf{Paths}_{\mathcal{G}}(s_0) \setminus [\![\varphi]\!]$.*

3.2 The logic oPATL

To express the observability of an agent, we would consider the transparent paths, i.e., behaviours observable (non-opaque) to her. The level of observability can be considered as the degree of transparency of the property enforced by the strategy of a coalition, which can be measured by calculating the probability of the transparent paths satisfying the property. We now present oPATL, an extension of probabilistic alternating-time temporal logic (PATL) [14], that characterises agents' quantified ability to enforce temporal properties. The key additions of oPATL include an *observability operator* and a *probabilistic (observability) operator*.

Definition 12. *Let $\mathcal{M} = (\mathsf{Ag}, \mathcal{G}, s_0, \mathsf{Ap}, L, \{obs_i\}_{i\in\mathsf{A}})$. The syntax of* oPATL *includes three classes of formulae:* state and path formulae*, and* observability formulae *ranged over by ϕ, ψ and Φ, respectively.*

$$\begin{array}{lcl} \phi & ::= & a \mid \neg\phi \mid \phi \wedge \phi \mid \mathbf{P}_{\bowtie p}\,\langle\mathsf{A}\rangle[\psi] \mid \mathcal{D}_{\bowtie p}\,\langle\mathsf{A}\rangle[\Phi] \\ \psi & ::= & \mathbf{X}\phi \mid \phi\mathbf{U}\phi \mid \phi\mathbf{R}\phi \mid \neg\psi \mid \psi \wedge \psi \\ \Phi & ::= & \mathbf{O}_i\,[\psi] \mid \neg\Phi \mid \Phi \wedge \Phi \end{array}$$

where $a \in \mathsf{Ap}$ is an atomic proposition*, $\mathsf{A} \subseteq \mathsf{Ag}$ is a set of agents, $\langle\mathsf{A}\rangle$ is the strategy quantifier, $\langle\mathsf{A}\rangle[\psi]$ expresses the property that coalition A has a strategy to enforce ψ, $i \in \mathsf{A} \subseteq \mathsf{Ag}$ is an agent, $\bowtie \in \{\leq, <, \geq, >\}$, $p \in [0, 1]$ is a probability bound.*

Note that oPATL formula is defined relative to a state, path formulae are only allowed inside the observability operator $\mathbf{O}_i\,[\cdot]$ and the probabilistic operator $\mathbf{P}_{\bowtie p}\,\langle\mathsf{A}\rangle[\cdot]$. The formula $\mathbf{P}_{\bowtie p}\,\langle\mathsf{A}\rangle[\psi]$ expresses that A has a strategy such that the probability of satisfying path formula ψ is $\bowtie p$, when the strategy is followed. The observability formula $\mathbf{O}_i\,[\psi]$ expresses the property of behaviours satisfying $[\psi]$ are observable to agent i. Intuitively, it is satisfied if for each path ρ satisfying ψ one cannot find a path ρ' violating ψ such that ρ and ρ' observationally equivalent to each other - from agent i's view. This operator would allow us to reason about the observability of agent i on system behaviours to enforce the property ψ. The quantitative observability formula $\mathcal{D}_{\bowtie p}(\langle\mathsf{A}\rangle[\Phi])$ expresses that A has a strategy π_{A} such that the degree of the observability enforcing path property considered in Φ is $\bowtie p$. $\mathsf{Paths}_{\mathcal{G}}(s, \pi_{\mathsf{A}})$ is used to denote the set of all paths of $\mathcal{G}$ starting from s and consistent with π_{A}.

Definition 13. *Let $\mathcal{M} = (\mathsf{Ag}, \mathcal{G}, s_0, \mathsf{Ap}, L, \{obs_i\}_{i\in\mathsf{Ag}})$. Semantics for* oPATL *include three satisfaction relations regarding the three notions of formulae (state, path, observability formulae).*

For a state $s \in S$ of $\mathcal{G}$, the satisfaction relation *$s \models_{\mathcal{M}} \phi$ for state formulae denotes "s satisfies ϕ":*

- *$s \models_{\mathcal{M}} a$ iff $a \in L(s)$.*
- *$s \models_{\mathcal{M}} \neg\phi$ iff $s \not\models_{\mathcal{M}} \phi$.*
- *$s \models_{\mathcal{M}} \phi \wedge \phi'$ iff $s \models_{\mathcal{M}} \phi$ and $s \models_{\mathcal{M}} \phi'$.*
- *$s \models_{\mathcal{M}} \mathbf{P}_{\bowtie p}\langle\mathsf{A}\rangle[\psi]$ iff $\exists\pi_{\mathsf{A}}$, the probability of the consistent paths with π_{A} over the set A, from state s, that ψ is true, satisfies $\bowtie p$, i.e., $\mathsf{Prob}(s, [\![\langle\mathsf{A}\rangle[\psi]]\!]) \bowtie p$, where: $\mathsf{Prob}(s, [\![\langle\mathsf{A}\rangle[\psi]]\!]) = \mathbb{P}_s([\![\langle\mathsf{A}\rangle[\psi]]\!]) = \mathbb{P}_s\{\rho \in \mathsf{Paths}_{\mathcal{G}}(s, \pi_{\mathsf{A}}) \mid \rho \models \psi\}$.*
- *$s \models_{\mathcal{M}} \mathcal{D}_{\bowtie p}\langle\mathsf{A}\rangle[\Phi]$ iff from state s, the probability of outgoing observable paths enforced by Φ that are consistent with π_{A} of a coalition A, satisfies $\bowtie p$: $\mathsf{Prob}(s, [\![\langle\mathsf{A}\rangle\Phi]\!]) \bowtie p$, where for the case of $\Phi = \mathbf{O}_i[\psi]$ and $\Phi' = \mathbf{O}_j[\psi']$:*

$$\begin{aligned} \mathsf{Prob}(s, [\![\langle\mathsf{A}\rangle\Phi]\!]) = \ & \mathbb{P}_s([\![\langle\mathsf{A}\rangle[\psi]]\!] \setminus obs_i^{-1}(obs_i([\![\neg\psi]\!]))) \\ \mathsf{Prob}(s, [\![\neg\Phi]\!]) = \ & \mathbb{P}_s([\![\langle\mathsf{A}\rangle[\psi]]\!] \cap obs_i^{-1}(obs_i([\![\neg\psi]\!]))) \\ \mathsf{Prob}(s, [\![\Phi \wedge \Phi']\!]) = \ & \mathbb{P}_s(([\![\langle\mathsf{A}\rangle[\psi]]\!] \setminus obs_i^{-1}(obs_i([\![\neg\psi]\!]))) \\ & \cap([\![\langle\mathsf{A}\rangle[\psi']]\!] \setminus obs_j^{-1}(obs_j([\![\neg\psi']\!])))). \end{aligned}$$

For a path ρ of $\mathcal{G}$, we define:

- *$\rho \models_{\mathcal{M}} \mathbf{X}\phi$ iff $\rho_s(1) \models \phi$.*
- *$\rho \models_{\mathcal{M}} \phi\mathbf{U}\phi'$ iff there exists $i \in \mathbb{N}$ s.t. $\rho_s(i) \models_{\mathcal{M}} \phi'$ and $\rho_s(j) \models_{\mathcal{M}} \phi$ for all $j < i$.*
- *$\rho \models_{\mathcal{M}} \phi\mathbf{R}\phi'$ iff for all $i \in \mathbb{N}$ at least one of the following is true: i) $\rho_s(i) \models_{\mathcal{M}} \phi'$, ii) $\rho_s(j) \models_{\mathcal{M}} \phi$ for some $j < i$.*
- *$\rho \models_{\mathcal{M}} \neg\psi$ iff $\rho \not\models_{\mathcal{M}} \psi$.*
- *$\rho \models_{\mathcal{M}} \psi \wedge \psi'$ iff $\rho \models_{\mathcal{M}} \psi$ and $\rho \models_{\mathcal{M}} \psi'$.*

Finally, for $s \models_{\mathcal{M}} \Phi$, we define observability formulae Φ:

- *$s \models_{\mathcal{M}} \mathbf{O}_i\,[\psi]$ iff for each path $\rho \in \mathsf{Paths}_{\mathcal{G}}(s)$ s.t. $\rho \models_{\mathcal{M}} \psi$, and for all $\rho' \in \mathsf{Paths}_{\mathcal{G}}(s)$ s.t. $\rho' \not\models_{\mathcal{M}} \psi$: $obs_i(\rho) \neq obs_i(\rho')$.*
- *$s \models \neg\Phi$ iff $s \not\models \Phi$, i.e., for each path $\rho \in \mathsf{Paths}_{\mathcal{G}}(s)$ s.t. $\rho \models_{\mathcal{M}} \psi$, there exists a path $\rho' \in \mathsf{Paths}_{\mathcal{G}}(s)$ s.t. $\rho' \not\models_{\mathcal{M}} \psi$: $obs_i(\rho) = obs_i(\rho')$.*
- *$s \models \Phi \wedge \Phi'$ iff $s \models_{\mathcal{M}} \Phi$ and $s \models_{\mathcal{M}} \Phi'$.*

Example 3. *Consider the model in Example 1. Assume we are interested in analysing the observability of voter 2 regarding her observation function in Table 1. Consider the property* "eventually candidate

X wins", *i.e.,* $\psi = \mathbf{F}(cx > cy \wedge o = 2)$*, where* cx *and* cy *denotes the final ballots* X *and* Y *received, and* $o = 2$ *indicates the state of voting process being closed. The operator* $\mathbf{F}\phi$ *is defined as "*$\texttt{true}\ \mathbf{U}\ \phi$*", so the probabilistic observability property is specified as:* $\mathcal{D}_{\leq p}\langle 1,2,3\rangle[\mathbf{O}_2[\psi]]$*, where* $p = \frac{1}{3}$ *is a probability threshold. It is easy to notice that, the above formula would return* `true`*, since ("wait" actions have been omitted here for simplifying the expression without introducing any confusion):*

$$\mathsf{tr}([\![\psi]\!]) = \{\tfrac{1}{12}.\mathtt{Op}_0 V_1^X V_2^X V_3^X \mathtt{Cl}_0, \tfrac{1}{6}.\mathtt{Op}_0 V_1^X V_2^X V_3^Y \mathtt{Cl}_0, \tfrac{1}{12}.\mathtt{Op}_0 V_1^X V_2^Y V_3^X \mathtt{Cl}_0, \tfrac{1}{6}.\mathtt{Op}_0 V_1^Y V_2^X V_3^X \mathtt{Cl}_0\}$$

and thus,

$$\mathsf{tr}([\![\mathbf{O}_2[\psi]]\!]) = \{\frac{1}{12}.\mathtt{Op}_0 V_1^X V_2^X V_3^X \mathtt{Cl}_0, \frac{1}{6}.\mathtt{Op}_0 V_1^X V_2^X V_3^Y \mathtt{Cl}_0\},$$

this is because under the observation function specified in Table 1, traces $\mathtt{Op}_0 V_1^X V_2^Y V_3^X \mathtt{Cl}_0$ *and* $\mathtt{Op}_0 V_1^Y V_2^X V_3^X \mathtt{Cl}_0$ *are covered by violating* ψ *traces* $\mathtt{Op}_0 V_1^X V_2^Y V_3^Y \mathtt{Cl}_0$ *and* $\mathtt{Op}_0 V_1^Y V_2^X V_3^Y \mathtt{Cl}_0$ *respectively from voter 2's view:*

$$obs_2(\mathtt{Op}_0 V_1^X V_2^Y V_3^X \mathtt{Cl}_0) = obs_2(\mathtt{Op}_0 V_1^X V_2^Y V_3^Y \mathtt{Cl}_0) = \mathtt{Op}_0 Y_2 \mathtt{Cl}_0,$$
$$obs_2(\mathtt{Op}_0 V_1^Y V_2^X V_3^X \mathtt{Cl}_0) = obs_2(\mathtt{Op}_0 V_1^Y V_2^X V_3^Y \mathtt{Cl}_0) = \mathtt{Op}_0 X_2 \mathtt{Cl}_0.$$

Therefore,

$$\mathsf{Prob}([\![\langle 1,2,3\rangle \mathbf{O}_2[\psi]]\!]) = \mathsf{Prob}(\{\frac{1}{12}.\mathtt{Op}_0 V_1^X V_2^X V_3^X \mathtt{Cl}_0, \frac{1}{6}.\mathtt{Op}_0 V_1^X V_2^X V_3^Y \mathtt{Cl}_0\}) = \frac{1}{4}$$

which is less than $p = \frac{1}{3}$ *and thus* $\mathcal{D}_{\leq p}\langle 1,2,3\rangle[\mathbf{O}_2[\psi]]$ *returns* `true`.

4 Verification of Observability Properties

Intuitively, verification of probabilistic observability answers the question "to which degree the system is observable to an agent $i \in \mathsf{Ag}$?", relative to a task expressed as property $[\psi]$ following the strategy of a coalition $\mathsf{A} \subseteq \mathsf{Ag}$, and the observation function of the agent obs_i. Since oPATL is a branching time logic, the overall approach is to recursively compute the satisfaction set $\mathsf{Sat}(\phi)$ of states satisfying formula ϕ over the structure of the formula.

For the *propositional logic fragment* of oPATL, the computation of this set for atomic propositions and logical connectives follows the conventional CTL model checking [4] and is sketched below:
(1) Convert the oPATL formulae in a positive normal form, that is, formulae built by the basic modalities $\mathbf{O}[\mathbf{X}\phi]$, $\mathbf{O}[\phi\mathbf{U}\phi']$, and $\mathbf{O}[\phi\mathbf{R}\phi']$, and successively pushing negations inside the formula at hand: $\neg\texttt{true} \rightsquigarrow \texttt{false}$, $\neg\texttt{false} \rightsquigarrow \texttt{true}$, $\neg\neg\phi \rightsquigarrow \phi$, $\neg(\phi \wedge \phi') \rightsquigarrow \neg\phi \vee \neg\phi'$, $\neg(\phi \vee \phi') \rightsquigarrow \neg\phi \wedge \neg\phi'$, $\neg\mathbf{X}\phi \rightsquigarrow \mathbf{X}\neg\phi$, $\neg(\phi\mathbf{U}\phi') \rightsquigarrow \neg\phi\mathbf{R}\neg\phi'$, $\neg(\phi\mathbf{R}\phi') \rightsquigarrow \neg\phi\mathbf{U}\neg\phi'$;
(2) Recursively compute the satisfaction sets $\mathsf{Sat}(\phi') = \{s \in S \mid s \models \phi'\}$ for all state subformulae ϕ' of ϕ: the computation carries out a bottom-up traversal of the parse tree of the state formula ϕ starting from the leafs of the parse tree and completing at the root of the tree which corresponds to ϕ, where the nodes of the parse tree represent the subformulae of ϕ and the leafs represent an atomic proposition $\alpha \in \mathsf{Ap}$ or `true` or `false`. All inner nodes are labelled with an operator. For positive normal form formulae, the labels of the inner nodes are $\neg$, $\wedge$, $\mathbf{O}[\mathbf{X}]$, $\mathbf{O}[\mathbf{U}]$, $\mathbf{O}[\mathbf{R}]$. At each inner node, the results of the computations of its children are used and combined to build the states of its associated subformula. In particular, satisfaction sets for the propositional logic fragment state formula are given as follows:

- $\mathsf{Sat}(\texttt{true}) = S$,
- $\mathsf{Sat}(\alpha) = \{t \in S \mid \alpha \in \eta(t)\}$,
- $\mathsf{Sat}(\neg\phi) = S \setminus \mathsf{Sat}(\phi)$,
- $\mathsf{Sat}(\phi \wedge \phi') = \mathsf{Sat}(\phi) \cap \mathsf{Sat}(\phi')$,

(3) Check whether $s \in \mathsf{Sat}(\phi)$.

For the treatment of subformulae of the form $\phi = \mathbf{P}_{\bowtie p}\langle\mathsf{A}\rangle[\psi]$, in order to determine whether $s \in \mathsf{Sat}(\phi)$, the probability of consistent paths with π_A under coalition A for behaviour specified by ψ, i.e., $\mathsf{Prob}(s \models_\mathcal{M} \langle\mathsf{A}\rangle[\psi])$, needs to be established, then:

$$\mathsf{Sat}(\mathbf{P}_{\bowtie p}\langle\mathsf{A}\rangle[\psi]) = \{s \in S \mid \mathsf{Prob}(s \models_\mathcal{M} \langle\mathsf{A}\rangle[\psi]) \bowtie p\}$$

The computation of the probability can follow the conventional PATL model checking algorithms, e.g., [14].

We now focus on the treatment state formulae of the form $\mathcal{D}_{\bowtie p}[\mathbf{O}_i\langle\mathsf{A}\rangle\psi]$. The problem reduces to computing the probability of observable paths that are satisfying property ψ and consistent with strategies of coalition A, from agent i's view.

Definition 14. *Given a* POMAS $\mathcal{M} = (\mathcal{G}, s_0, \mathsf{Ag}, \mathsf{Ap}, \{obs_i\}_{i\in\mathsf{Ag}})$*, a task in property* ψ *required to be completed under a strategy* π_A *of a coalition* $\mathsf{A} \in \mathsf{Ag}$*, the probabilistic verification problem of observability property is to decide whether* $s_0 \models_\mathcal{M} \mathcal{D}_{\bowtie p}(\mathbf{O}_i\ \langle\mathsf{A}\rangle[\psi])$*, i.e.,*

$$\mathbb{P}_{s_0}([\![\langle\mathsf{A}\rangle[\psi]]\!] \setminus obs_i^{-1}(obs_i([\![\langle\mathsf{A}\rangle[\neg\psi]]\!]))) \bowtie p.$$

Therefore, we focus on computing $\mathbb{P}_{s_0}(\langle\mathsf{A}\rangle[\psi] \setminus obs_i^{-1}(obs_i(\langle\mathsf{A}\rangle[\neg\psi])))$ for a given POMAS $\mathcal{M}$ and coalition A. We assume that the available actions of agent $i \in \mathsf{Ag}$ of $\mathcal{M}$ in state s are $\{a_{i,1}, \ldots, a_{i,k_i}\}$. The brief procedure for checking $s \models_\mathcal{M} \mathcal{D}_{\bowtie p}(\mathbf{O}_i\ \langle\mathsf{A}\rangle[\psi])$ is sketched as follows.

- Find all consistent paths Π and the corresponding traces Λ, represented in regular-expression-like format (denoted by $\mathsf{Reg}(\cdot)$), satisfying ψ under mixed strategy π_A of coalition A, denoted by:

$$\Pi = \{\mathsf{Reg}(\rho_{\pi_\mathsf{A}}) \mid \rho_{\pi_\mathsf{A}} \models_\mathcal{M} \psi\} \qquad \Lambda = \{\mathsf{erase}(\rho) \mid \rho \in \Pi\}.$$

- Find all consistent paths Π' and the corresponding traces Λ, represented in regular-expression-like format (denoted by $\mathsf{Reg}(\cdot)$), violating ψ under mixed strategy π_A of coalition A:

$$\Pi' = \{\mathsf{Reg}(\rho'_{\pi_\mathsf{A}}) \mid \rho_{\pi_\mathsf{A}} \not\models_\mathcal{M} \psi\} \qquad \Lambda' = \{\mathsf{erase}(\rho') \mid \rho' \in \Pi'\}.$$

- Find all ψ-opaque traces:

$$\Lambda'' = \{\lambda'' \mid \lambda'' \in \Lambda \wedge \exists\lambda' \in \Lambda'.(obs_i(\lambda') = obs_i(\lambda''))\}.$$

- Compute the probability of ψ-observable traces:

$$d = \mathbb{P}_{s_0}([\![\mathbf{O}_i\langle\mathsf{A}\rangle[\psi]]\!]) = \sum_{\xi\in(\Lambda\setminus\Lambda'')} \mathsf{Prob}(\xi).$$

- Return true if $d \bowtie p$, return false otherwise.

We present the detailed procedure of computing the probability of ψ-observable traces starting at s under mixed strategy π_A of coalition A from the observation of $i \in \mathsf{Ag}$, in Algorithm 1. Algorithm 2 computes a set of regular-expression-like formatted paths satisfying $\phi\mathbf{U}\phi'$. Similarly, an algorithm can be proposed to compute a set of regular-expression-like formatted paths satisfying $\phi\mathbf{R}\phi'$. We can thus compute all regular-expression-like formatted paths Π (and Π') starting from s and satisfying (and violating) ψ and consistent with mixed strategy π_A.

Soundness. Given a POMAS $\mathcal{M}$, a probability threshold p, and a task specified in ψ to be completed:

$$s_0 \models_\mathcal{M} \mathcal{D}_{\bowtie p}\langle\mathsf{A}\rangle[\mathbf{O}_i[\psi]] \qquad \text{iff} \qquad \mathbb{P}([\![\langle\mathsf{A}\rangle[\mathbf{O}_i[\psi]]]\!]) \bowtie p.$$

Algorithm 1: Computing the probability of ψ-observable consistent traces under π_{A} from i's view - $\mathcal{D}(\langle \mathsf{A} \rangle \mathbf{O}_i[\psi])$.

Data: $\mathcal{M}, s, i, A, \psi$
Result: The probability $\mathcal{D}(\langle \mathsf{A} \rangle \mathbf{O}_i[\psi])$
switch ψ **do**
  case $\mathbf{X}\phi$:
    $\mathsf{Sat}(\psi) \leftarrow \cup\{s \xrightarrow{\alpha} s' \mid \mathsf{Post}(s) = s' \wedge s' \in \mathsf{Sat}(\phi)\}$,
    $\mathsf{Sat}(\neg\psi) \leftarrow \cup\{s \xrightarrow{\alpha} s' \mid \mathsf{Post}(s) = s' \wedge s' \in \mathsf{Sat}(\neg\phi)\}$;
  case $\phi \mathbf{U} \phi'$: $\mathsf{Sat}(\psi) \leftarrow$ **compU**$(\mathcal{M}, s, \phi, \phi')$,
    $\mathsf{Sat}(\neg\psi) \leftarrow$ **compR**$(\mathcal{M}, s, \neg\phi, \neg\phi')$;
  case $\phi \mathbf{R} \phi'$: $\mathsf{Sat}(\psi) \leftarrow$ **compR**$(\mathcal{M}, s, \phi, \phi')$,
    $\mathsf{Sat}(\neg\psi) \leftarrow$ **compU**$(\mathcal{M}, s, \neg\phi, \neg\phi')$;
end
$p\Lambda \leftarrow \{p\lambda \mid p\lambda.tr \leftarrow \lambda \wedge p\lambda.pr \leftarrow \mathsf{Prob}(\lambda) \wedge \lambda = tr(\rho)$ for $\rho \in \mathsf{Paths}_{\mathcal{G}}(s, \pi_{\mathsf{A}}) \wedge \rho \in \mathsf{Sat}(\psi)\}$;
$p\Lambda' \leftarrow \{p\lambda \mid p\lambda.tr \leftarrow \lambda \wedge p\lambda.pr \leftarrow \mathsf{Prob}(\lambda) \wedge \lambda = tr(\rho)$ for $\rho \in \mathsf{Paths}_{\mathcal{G}}(s, \pi_{\mathsf{A}}) \wedge \rho \in \mathsf{Sat}(\neg\psi)\}$;
$p\Lambda'' = \{\}$;
for *each* $p\lambda \in p\Lambda$ **do**
  for *each* $p\lambda' \in p\Lambda'$ **do**
    if $obs_i(p\lambda.tr) \subseteq obs_i(p\lambda'.tr)$ **then**
      $p\Lambda'' \leftarrow p\Lambda'' \cup \{p\lambda\}$; break;
    end
  end
end
$p\Lambda_{\mathbf{O}} \leftarrow p\Lambda \setminus p\Lambda''$; /* observable traces */
$d = \sum_{\lambda \in p\Lambda_{\mathbf{O}}} \lambda.pr$;
return d.

The satisfaction relation of $\mathcal{D}_{\bowtie p}\langle \mathsf{A} \rangle[\mathbf{O}_i[\psi]]$ and the computation of the probability of ψ-observable consistent traces under mixed strategy π_{A} of coalition A from observer i's view is described in Algorithm 1. The algorithm will terminate since $\mathsf{Sat}(\psi)$ are processed and computed as a set of regular-expression-like formatted traces satisfying ψ (such as Algorithm 2 presented in the technical appendix). Probability of such a trace is calculated by multiplication of the probability of each transition label for non-cycle part, and multiplication of $p/(1-p)$ for a cycle with probability p.

Complexity. The worst case of checking satisfaction of the observability formula, specified in Algorithm 1, is EXPSPACE in general. The formula of observability is essentially in the form of $\forall\forall$, the algorithm traverses all consistent traces under mixed strategy π_{A} satisfying ψ and all traces of those violating ψ, and conducts observation equivalence comparison. So the worst case complexity here follows the complexity of the hyper property model checking problem with two quantifier ($\forall$) alternations, and thus EXPSPACE. We hypothesise the time complexity of checking satisfaction of oPATLformula is exponential to the size of the POMAS, and doubly exponential in the size of the formula itself, similar to model checking HyperLTL [16]. If all pairs of traces are evaluated in parallel, the evaluation of each individual pair corresponds to the evaluation of an LTL formula over a single trace, which can be done in polylogarithmic time on a parallel computer with a polynomial number of processors [9].

Example 4. *The proposed work has been implemented on top of PRISM, which allows to specify properties which evaluate to a value using, e.g., $\mathcal{D}_{=?}\langle A \rangle[\mathbf{O}_i[\psi]]$. The result of Example 3 can be automatically produced below, which meets our calculation by hand.*

```
Result: 0.25.{
    0.08333333333333333:vX1vX2vX3cl0->X2cl0,
    0.16666666666666666:vX1vX2vY3cl0->X2cl0
} (value in the initial state)
```

5 Implementation and Examples

A prototype tool for specifying and verifying the observability problem in MASs has been built on the top of the PRISM model checker [22]. Models are described in an extension of the PRISM modelling language with observations and transition labels, the new model type is denoted as "`pomas`". Properties are described in an extension of the PRISM's property specification language with the observability operator. The tool and examples are available from [30].

Algorithm 2: Computing $\mathsf{Sat}(\phi \mathbf{U} \phi')$: **compU**$(\mathcal{M}, s, \phi, \phi')$

Data: $\mathcal{M}, s, \phi, \phi'$
Result: Regular-expression-like formatted paths satisfying $\phi \mathbf{U} \phi'$
$\Pi \leftarrow \{\}$; $i \leftarrow 0$;
for *each* $t_i \in \mathsf{Sat}(\phi')$ **do**
  $T_i \leftarrow \{t_i\}$; $\Pi_i \leftarrow \{\pi \mid \pi(0) = t_i\}$;
  while
   $\{s_j \in \mathsf{Sat}(\phi) \setminus (T_i \cup \mathsf{Sat}(\phi')) \mid \mathsf{Post}(s_j) \cap T_i \neq \emptyset\} \neq \emptyset$ **do**
    let
      $s_j \in \{s_j \in \mathsf{Sat}(\phi) \setminus (T_i \cup \mathsf{Sat}(\phi')) \mid \mathsf{Post}(s_j) \cap T_i \neq \emptyset\}$;
    if $s_j \in \mathsf{Post}(s_j) \cap T_i$ **then**
      /* There is a self-loop: wrap it with a star and concatenate paths starting from a state in $\mathsf{Post}(s_j) \cap T_i$ */
      for *each* $\pi' \in \Pi_i$ *s.t.* $\pi'(0) \in \mathsf{Post}(s_j) \cap T_i$ **do**
        $\Pi_i \leftarrow \Pi_i \cup \{(s_j \xrightarrow{\alpha} s_j)^* + \pi'[1...]\}$;
      end
    end
    for *each:* $q_1 \in \mathsf{Post}(s_j) \cap T_i, q_2 \in \mathsf{Post}(q_1) \cap T_i, \ldots, q_n \in \mathsf{Post}(q_{n-1}) \cap T_i$ *s.t.* $\mathsf{Post}(q_n) \cap T_i = \emptyset$ **do**
      if $\mathsf{Pre}(s_j) \notin \{q_1, q_2, \ldots q_n\}$ **then**
        for *each* $\pi' \in \Pi_i$ *s.t.* $\pi'(0) \in \mathsf{Post}(s_j) \cap T_i$ **do**
          $\Pi_i \leftarrow \Pi_i \cup \{s_j \xrightarrow{\alpha} q_1 + \pi'[1...]\}$;
        end
      end
      else if $\mathsf{Pre}(s_j) = q_n \wedge s_j \in \mathsf{Post}(q_n)$ **then**
        /* There is a cycle, wrap it with a star and concatenate paths starting from a state in $\mathsf{Post}(s_j) \cap T_i$ */
        for *each* $\pi' \in \Pi_i$ *s.t.* $\pi'(0) \in \mathsf{Post}(s_j) \cap T_i$ **do**
          $\Pi_i \leftarrow \Pi_i \cup \{(s_j \xrightarrow{\alpha_1} q_1 \xrightarrow{\alpha_2} \ldots \xrightarrow{\alpha_n} \mathsf{Pre}(s_j) \xrightarrow{\alpha_{n+1}} s_j)^* + \pi'[1...]\}$;
        end
      end
    end
    $T_i \leftarrow T_i \cup \{s_j\}$;
  end
  $\Pi \leftarrow \Pi \cup \Pi_i$;
  $i \leftarrow i + 1$;
end
return Π.

Example: a simple supply chain. Nowadays supply chain is a core part of businesses concerned with transporting products between different parties such as customers, retailers, coordinators, delivery services, and suppliers. Agents of those parties communicate with each other for buying and selling items. Suppliers compete with each others to obtain more jobs and profit, they might partially observe the procedure of the supply chain and try to induce commercial information. Customer might partially observe the pipeline of the supply chain, and try to learn information about the origin of the products. Such a scenario can be naturally modelled as a POMAS, and we are interested in studying the quantified observability by agents, which may cause information flow and affect future decision-making.

To illustrate our framework and its implementation, we consider a basic commercial supply chain shown in Fig. 1 as an example. Assume there are a number of agents in the system: 1) customer: buying products (denoted by `ordc`) from the retailer; 2) retailer: requesting to order products (denoted by `ordr`) from suppliers through the coordinator; 3) coord: the coordinator, processing requests/orders from the retailer, sending requests to and receiving response from sup-

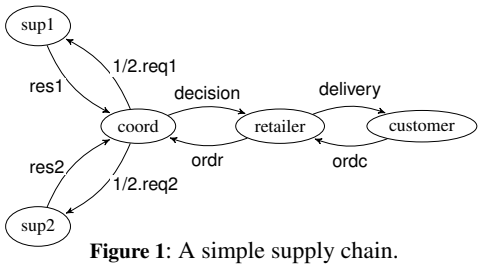

Figure 1: A simple supply chain.

pliers, making decisions such as which supplier provides products, returning decision to the retailers; 4) sup$i$: the i^{th} supplier, receiving requests from/responding availability to the coordinator. The agents and their observability are given as follows: action `ordc` is hidden to *supi* and is observed as `Ordc` from the rest of the agents' view; action `ordr` is hidden to *customer* and is observed as `Ordr` from the rest of the agents' view; action `reqi`, denoting *coord* sending requests to *supi*, is hidden to *customer & retailer*, is observed as `Req` to *supi*, and is observed as `Reqi` to *coord*; action `resi`, denoting *supi* responding to *coord*, is hidden to *customer & retailer*, is observed as `Res` to *supi*, and is observed as `Resi` to *coord*; action `decisioni`, denoting *coord* deciding *supi* to provide the products, is hidden to *customer & retailer*, is observed as `Dec` to *supi*, and is observed as `Deci` to *coord*; action `delivery` is observed `Dlv` to all of the agents.

Let A denote the set of agents defined above. We could ask questions such as "what is the degree of the observability by *sup1* if the product is successfully delivered to the customer but the supplier is not *sup1*?", specified as $\mathbf{P} =_? \langle A \rangle\ [\mathbf{O}_{sup1}\ [\mathbf{F}\ (\texttt{dec!} = 1\ \&\ \texttt{dlv} = 1)]]$, where `dec` is the variable defined in the module to specify the decision made by the coordinator: `dec=i` denotes supplier i will provide the requested products, `dlv` is the variable defined in the module to specify the status of product delivery: `dlv=1` denotes the product has been successfully delivered to the customer. The result generated by the tool is presented as follows:

```
Result: 0.5.{
  0.25:ordcordrreq1res1decision2delivery->OrdrReqResDecDlv,
  0.25:ordcordrreq2res2decision2delivery->OrdrReqResDecDlv
} (value in the initial state)
```

This meets our intuition, since the listed two traces satisfying $\mathbf{F}(\texttt{dec!} = 1 \& \texttt{dlv} = 1)$ are not covered by traces violating the property, are thus observable to *sup1*. If we ask question "what is the degree of the observability by *customer* if the product is successfully delivered to him but the supplier is not *sup1*?", which can be specified as $\mathbf{P} =_? \langle A \rangle\ [\mathbf{O}_{customer}\ [\mathbf{F}\ (\texttt{dec!} = 1\ \&\ \texttt{dlv} = 1)]]$. The result generated by the tool would be:

```
Result: 0.0.{
} (value in the initial state)
```

Example: A peer-to-peer (P2P) file sharing network. This case study considers a variant of a Gnutella-like P2P network for file sharing, allowing users to communicate and access files without the need for a server. The individual users in this network are referred to as peers. Gnutella protocol defines a decentralised approach making use of distributed systems, where the peers are called nodes, and the connection between peers is called an edge between the nodes, thus resulting in a graph-like structure. A peer wishing to download a file would send a query request Qry packet to all its neighbouring nodes under a probability distribution. If those nodes don't have the required file, they pass on the query to their neighbours and so on. When the peer with the requested file is found, the query flooding stops and it sends back a query hit packet Hit following the reverse path. If there are multiple query hits, the client selects one of these peers. The client thus builds a connection with the peer offering the resource and download the resource. Fig. 2 shows an example process of downloading a file using the Gnutella-like P2P network.

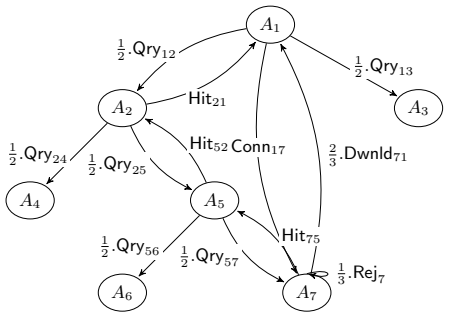

Figure 2: Downloading a file using Gnutella P2P network.

Let $A = \{A_i \mid i \in \{1,2,3,4,5,6,7\}\}$ denote a set of nodes (agents) in the network, Qry_{ij} denote node A_i sends a query request to node A_j, Hit_{ji} denotes node A_j sends a query hit message to A_i, Conn_{ij} denotes node A_i builds a connection with node A_j, Dwnld_{ji} denotes node A_i downloads the file from node A_j. Suppose A_1 is the agent node looking for a resource, A_7 is the agent node willing to offer the requested resource. A malicious node, say A_4, tries to learn some information of the situation of the downloading procedure of A_1. We could ask a question "what is the degree of the observability of A_4 on the procedure of A_1 finally downloading the requested file?", which could be formalised in formula: $\mathbf{P} =_? [\langle A \rangle\ \mathbf{O}_4\ \mathbf{F}$ "A_1 *downloads the requested file*"$]$. Assume the observation function of A_4 is defined as: `Qry12->Q1, Qry13->, Qry24->Q2, Qry25->, Qry56->, Qry57->, Hit21->, Hit52->, Hit75->, Conn17->Con1, Dwnld71->Dwn, Rej71->Fail`

The result generated regarding to the pre-defined observation function is presented as follows:

```
Result: 0.083.{
0.083:Qry12Qry25Qry57Hit75Hit52Hit21Conn17Dwnld71->Q1Con1Dwn
} (value in the initial state)
```

6 Conclusions and Future Work

We have constructed a formal framework for quantitatively specifying and verifying observability properties in MASs. Observability analysis can be used to capture information transparency and thus information leakage in MASs for information flow security concerns. A direct application of this work is privacy loss and information leakage for security analysis in MASs. The focus of this paper is on developing the theory and framework which provides a foundation for future work to assist operators in managing information-leakage risks while optimising performance effectiveness in collaborative multi-agent systems. Although the current implementation is a prototype for testing research ideas and demonstrating the verification framework's feasibility through small case studies, future plans include transitioning to a publicly available software tool.

In future, we intend to integrate our observability operator into Strategic Logic [13] and evaluate its suitability for various scenarios in information security analysis. We will also develop novel approaches to generate policies that capture the trade-off between task completion and gathering/restricting information learned via maximising/minimising agents' observability. Game-theoretic methods can be integrated into the framework to automatically identify an equilibrium between information transparency guarantees and performance objectives based on the quantified results produced by this work. Such an equilibrium can be used to indicate an optimal decision for operators to coordinate behaviours in multiple domains concerning combined effectiveness and information transparency. There is a wide range of applications where such abilities are necessary to balance the integrated capabilities and security risks.

References

[1] N. Alechina, B. Logan, H. N. Nguyen, F. Raimondi, and L. Mostarda, 'Symbolic model-checking for resource-bounded ATL', in *Proc. International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, pp. 1809–1810. ACM Press, (2015).

[2] M. S. Alvim, M. E. Andrés, and C. Palamidessi, 'Quantitative information flow in interactive systems', *Journal of Computer Security*, **20**(1), 3–50, (2012).

[3] M. Backes, 'Quantifying probabilistic information flow in computational reactive systems', in *Proc. European Symposium on Research in Computer Security (ESORICS)*, volume 3679 of *Lecture Notes in Computer Science*, pp. 336–354. Springer-Verlag, (2005).

[4] C. Baie and J.-P. Katoen, *Principles of Model Checking*, The MIT Press, 2008.

[5] P. Balbiani, O. Gasquet, and F. Schwarzentruber, 'Agents that look at one another', *Logic Journal of the IGPL*, **21**(3), 438–467, (2013).

[6] F. Belardinelli, A. Lomuscio, A. Murano, and S. Rubin, 'Verification of multi-agent systems with imperfect information andpublic actions', in *Proc. International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, pp. 1268–1276. ACM Press, (2017).

[7] F. Belardinelli, A. Lomuscio, A. Murano, and S. Rubin, 'Verification of multi-agent systems with public actions against strategy logic', *Artificial Intelligence*, **285**, 103302, (2020).

[8] F. Biondi, A. Legay, B. F. Nielsen, P. Malacaria, and A. Wasowski, 'Information leakage of non-terminating processes', in *Proc. IARCS Annual Conference on Foundations of Software Technology and Theoretical Computer Science (FSTTCS)*, volume 29 of *LIPIcs*, pp. 517–529. Schloss Dagstuhl, (2014).

[9] B. Bonakdarpour and B. Finkbeiner, 'The complexity of monitoring hyperproperties', *CoRR*, **abs/2101.07847**, (2021).

[10] M. Boreale, D. Clark, and D. Gorla, 'A semiring-based trace semantics for processes with applications to information leakage analysis', *Mathematical Structures in Computer Science*, **25**(2), 259–291, (2015).

[11] J. W. Bryans, M. Koutny, and C. Mu, 'Towards quantitative analysis of opacity', in *Proc. International Symposium Trustworthy Global Computing (TGC)*, volume 8191 of *Lecture Notes in Computer Science*, pp. 145–163. Springer-Verlag, (2012).

[12] T. Charrier, A. Herzig, E. Lorini, F. Maffre, and F. Schwarzentruber, 'Building epistemic logic from observations and public announcements', in *Proc. International Conference on Principles of Knowledge Representation and Reasoning (KR)*, pp. 268–277. AAAI Press, (2016).

[13] K. Chatterjee, T. A. Henzinger, and N. Piterman, 'Strategy logic', in *Proc. International Conference on Concurrency Theory (CONCUR)*, pp. 59–73. Springer-Verlag, (2007).

[14] T. Chen and J. Lu, 'Probabilistic alternating-time temporal logic and model checking algorithm', in *Proc. International Conference on Fuzzy Systems and Knowledge Discovery (FSKD)*, pp. 35–39. IEEE CS, (2007).

[15] T. Chothia, Y. Kawamoto, C. Novakovic, and D. Parker, 'Probabilistic point-to-point information leakage', in *Proc. IEEE Computer Security Foundations Symposium (CSF)*, pp. 193–205. IEEE CS, (2013).

[16] M. R. Clarkson, B. Finkbeiner, M. Koleini, K. K. Micinski, M. N. Rabe, and C. Sánchez, 'Temporal logics for hyperproperties', in *Proc. International Conference on Principles of Security and Trust (POST)*, volume 8414 of *Lecture Notes in Computer Science*, pp. 265–284. Springer-Verlag, (2014).

[17] M. C. Cooper, A. Herzig, F. Maffre, F. Maris, and P. Régnier, 'A simple account of multi-agent epistemic planning', in *Proc. European Conference on Artificial Intelligence (ECAI)*, volume 285 of *Frontiers in Artificial Intelligence and Applications*, pp. 193–201. IOS Press, (2016).

[18] L. Gasparini, T. J. Norman, and M. J. Kollingbaum, 'Observation-based multi-agent planning with communication', in *Proc. European Conference on Artificial Intelligence (ECAI)*, volume 285 of *Frontiers in Artificial Intelligence and Applications*, pp. 444–452. IOS Press, (2016).

[19] W. Van Der Hoek, N. Troquard, and M. Wooldridge, 'Knowledge and control', in *Proc. International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, pp. 719–726. IFAAMAS, (2011).

[20] X. Huang, K. Su, and C. Zhang, 'Probabilistic alternating-time temporal logic of incomplete information and synchronous perfect recall', in *Proc. AAAI Conference on Artificial Intelligence (AAAI)*. AAAI Press, (2012).

[21] M. H. R. Khouzani and P. Malacaria, 'Leakage-minimal design: Universality, limitations, and applications', in *Proc. IEEE Computer Security Foundations Symposium (CSF)*, pp. 305–317. IEEE CS, (2017).

[22] M. Kwiatkowska, G. Norman, and D. Parker, 'PRISM 4.0: Verification of probabilistic real-time systems', in *Proc. International Conference on Computer Aided Verification (CAV)*, volume 6806 of *Lecture Notes in Computer Science*, pp. 585–591. Springer-Verlag, (2011).

[23] M. Kwiatkowska, G. Norman, D. Parker, and G. Santos, 'Equilibria-based probabilistic model checking for concurrent stochastic games', in *Proc. International Symposium on Formal Methods (FM)*, volume 11800 of *Lecture Notes in Computer Science*, pp. 298–315. Springer-Verlag, (2019).

[24] A. Lomuscio, H. Qu, and F. Raimondi, 'MCMAS: an open-source model checker for the verification of multi-agent systems', *International Journal on Software Tools for Technology Transfer*, **19**(1), 9–30, (2017).

[25] P. Malacaria, M. H. R. Khouzani, C. S. Pasareanu, Q. Phan, and K. S. Luckow, 'Symbolic side-channel analysis for probabilistic programs', in *Proc. IEEE Computer Security Foundations Symposium (CSF)*, pp. 313–327. IEEE CS, (2018).

[26] M. J. Mataric, *Interaction and intelligent behavior*, Ph.D. dissertation, Massachusetts Institute of Technology, Cambridge, MA, USA, 1994.

[27] L. Mazaré, 'Using unification for opacity properties', in *Proc. Workshop on Issues in the Theory of Security (WITS)*, pp. 165–176, (2004).

[28] C. Mu, 'A language-based approach to analysing flow security properties in virtualised computing systems', in *Proc. International Symposium on Theoretical Aspects of Software Engineering (TASE)*, pp. 185–192. IEEE CS, (2020).

[29] C. Mu and D. Clark, 'Verifying opacity properties in security systems', *IEEE Trans. Dependable Secur. Comput.*, **20**(2), 1450–1460, (2023).

[30] C. Mu and J. Pang. Prototype tool and case studies. https://github.com/cmu777/obs-mas.git, 2022.

[31] J. Zand, J. Parker-Holder, and S. J. Roberts, 'On-the-fly strategy adaptation for ad-hoc agent coordination', in *Proc. International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, pp. 1771–1773. IFAAMAS, (2022).