On Verifying Expectations and Observations of Intelligent Agents

Sourav Chakraborty[1], **Avijeet Ghosh**[1], **Sujata Ghosh**[2] and **François Schwarzentruber**[3]
[1]Indian Statistical Institute, Kolkata
[2]Indian Statistical Institute, Chennai
[3]Univ Rennes, IRISA, France
{sourav, avijeet_r}@isical.ac.in, sujata@isichennai.res.in, francois.schwarzentruber@ens-rennes.fr

Abstract

Public observation logic (POL) is a variant of dynamic epistemic logic to reason about agent expectations and agent observations. Agents have certain expectations, regarding the situation at hand, that are actuated by the relevant protocols, and they eliminate possible worlds in which their expectations do not match with their observations. In this work, we investigate the computational complexity of the model checking problem for POL and prove its PSPACE-completeness. We also study various syntactic fragments of POL. We exemplify the applicability of POL model checking in verifying different characteristics and features of an interactive system with respect to the distinct expectations and (matching) observations of the system. Finally, we provide a discussion on the implementation of the model checking algorithms.

1 Introduction

Agents have expectations about the world around, and they reason on the basis of what they observe around them, and such observations may or may not match with the expectations they have about their surroundings. Let us first provide two examples showing the diverse nature of such reasoning phenomena.

- Consider a person traveling from Switzerland to France in a car. Here is one way she would know whether she is in France. According to her expectations based on the traffic light signals of the different states, if she observes the sequence of (green*-amber-red*)* ($*$ denotes the continuance of such sequences), she would know that she is in France, whereas if she observes (green*-amber-red*-amber)*, she would know that she is not.
- Consider three agents denoted by Sender (S), Receiver (R) and Attacker (A). Suppose S and R have already agreed that if S wants to convey that some decision has been taken, S would send a message, say m, to R; otherwise, S would send some other message, say m', to R. Suppose also that A has no information about this agreement. Then upon getting a message from S, there would be a change in the knowledge state of R but not A.

The first example concerns a certain rule that we follow in our daily life, and the second example brings in the flavour of coded message-passing under adversarial attacks. Expectations about the moves and strategies of other players also occur naturally in game theory, and possible behaviours of players are represented in these terms. Moving from theory to actual games, in the strategy video game Starcraft[1], one player may know/expect that the other player will attack her base as soon as possible, and thus may play accordingly. Games like Hanabi[2], and Colored Trails [de Weerd *et al.*, 2017] also consider the connection between expectations and observations regarding the moves and strategies of the other players.

The challenge now is to build intelligent systems that are able to reason about knowledge regarding expectations, and plan accordingly. Whereas epistemic logic [Fagin *et al.*, 1995] and more generally, its dynamic extensions, popularly known as *dynamic epistemic logics* (DEL) [van Ditmarsch *et al.*, 2008] help us to build agents that reason about knowledge, they do not offer any mechanism dealing with expectations. In the same way, epistemic planning, based on the model checking of DEL ([Bolander *et al.*, 2020]), extends classical planning with epistemic reasoning, but is unable to take agent expectations into account. Fortunately, following [Wang, 2011], *Public observation logic* (POL) [van Ditmarsch *et al.*, 2014], a variant of DEL, reasons about knowledge regarding expectations. POL provides dynamic operators for verifying whether a given epistemic property holds after observing some sequence of observations matching certain expectations that are modelled by regular expressions π.

However, investigations on algorithmic properties of POL were left open. In this paper, we show that the POL model checking is decidable and PSPACE-complete. Our result relies on automata theory and the careful use of an oracle for deciding the algorithm running in poly-space.

For practical purposes, we investigate syntactic fragments that offer better complexities than reasoning in the full language of POL (see Figure 1), and are suitable for relevant verification tasks:

- the Word fragment, where any regular expression π is a *word*, is sufficient to verify that some given plan leads to a state satisfying some epistemic property;

[1]https://en.wikipedia.org/wiki/StarCraft_(video_game)
[2]https://en.wikipedia.org/wiki/Hanabi_(card_game)

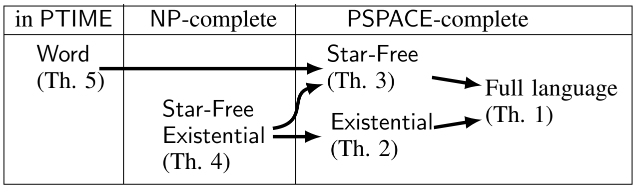

Figure 1: Complexity results of model checking for different fragment of POL. (arrows represent inclusion of fragments).

- the Existential fragment, where the dynamic operators of POL are all *existential*, is suitable for epistemic planning (e.g., does there exist a plan?);
- the Star-Free fragment, where the regular expressions π are *star-free*, embeds *bounded* planning (in which sequences of observations to synthesize are bounded by some constant). In particular, the Star-FreeExistential fragment (i.e. the intersection of the Star-Free and the Existential fragments) is suitable for bounded epistemic planning.

Outline. Section 2 recalls POL with a formal presentation of the two examples mentioned in the introduction. Section 3 deals with all our complexity results about the model checking problem of POL. Section 4 shows the applicability of POL and its fragments in modelling interactive systems. It also includes a discussion on the implementation. Section 5 presents the related work and section 6 concludes the paper.

2 Background and Preliminaries

We first provide an overview of public observation logic (POL) as introduced in [van Ditmarsch *et al.*, 2014]. Let $\mathbf{I}$ be a finite set of agents, $\mathbf{P}$ be a countable set of propositions describing the facts about the world and $\mathbf{\Sigma}$ be a finite set of actions. Below, we will not differentiate between the action of observing a phenomenon and the phenomenon itself.

2.1 Observations

For our purposes, we assume *observations* to be finite strings of actions. In the traffic example, an observation may be *green-amber-red-green* (abbreviated as $garg$) or, *green-amber-red-amber-green* (abbreviated as $garag$), among others, whereas, in the message-passing example, an observation is either m or m'. An agent may expect different (even infinitely many) potential observations at a given state, but to model agent expectations, they are described in a finitary way by introducing the *observation expressions* (as regular expressions over $\mathbf{\Sigma}$):

Definition 1 (Observation expressions)**.** *Given a finite set of action symbols $\mathbf{\Sigma}$, the language $\mathcal{L}_{obs}$ of* observation expressions *is defined by the following BNF:*

$$\pi \quad ::= \quad \emptyset \mid \varepsilon \mid a \mid \pi \cdot \pi \mid \pi + \pi \mid \pi^*$$

where $\emptyset$ denotes the empty set of observations, the constant ε represents the empty string, $\cdot$ denotes concatenation, $+$ is union, $*$ represents iteration and $a \in \mathbf{\Sigma}$.

In the traffic example, the observation expression $(g^*ar^*)^*$ models the traveller's expectation of traffic signals in case she is in France. In the other one, the expression m models the expectation of the receiver in case a decision is made.

The size of an observation expression π is denoted by $|\pi|$. The semantics for the observation expressions are given by *sets of observations* (strings over $\mathbf{\Sigma}$), similar to those for regular expressions. Given an observation expression π, its *set of observations* is denoted by $\mathcal{L}(\pi)$. For example, $\mathcal{L}(m) = \{m\}$, and $\mathcal{L}((g^*ar^*)^*) = \{\varepsilon, a, ga, ar, gar, gargar, \dots\}$. The regular language $\pi\backslash w$ is the set of words given by $\{v \in \mathbf{\Sigma}^* \mid wv \in \mathcal{L}(\pi)\}$. The regular language $\mathit{prefixes}(\pi)$ is the set of prefixes of words in $\mathcal{L}(\pi)$, that is, $w \in \mathit{prefixes}(\pi)$ iff $\exists v \in \mathbf{\Sigma}^*$ such that $wv \in \mathcal{L}(\pi)$ (namely, $\mathcal{L}(\pi\backslash w) \neq \emptyset$).[3]

Example 1. $(g^*ar^*a)^*\backslash(garaga) = r^*(g^*ar^*a)^*$ *denotes the language of words* $\{v : garaga \cdot v \in \mathcal{L}((g^*ar^*)^*)\}$*. The set* $\mathit{prefixes}((g^*ar^*a)^*)$ *contains* $garaga$*. However,* $garg$ *is not in* $\mathit{prefixes}((g^*ar^*a)^*)$ *and* $(g^*ar^*a)^*\backslash(garg)$ *is empty.*

2.2 Models

Epistemic expectation models [van Ditmarsch *et al.*, 2014] capture the expected observations of agents. They can be seen as epistemic models [Fagin *et al.*, 1995] together with, for each world, a set of potential or expected observations. Recall that an epistemic model is a tuple $\langle S, \sim, V\rangle$ where S is a non-empty set of worlds, $\sim$ assigns to each agent in $\mathbf{I}$ an equivalence relation $\sim_i \subseteq S \times S$, and $V : S \to 2^{\mathbf{P}}$ is a valuation function.

Definition 2 (Epistemic expectation model)**.** *An* epistemic expectation model *$\mathcal{M}$ is a quadruple $\langle S, \sim, V, \mathit{Exp}\rangle$, where $\langle S, \sim, V\rangle$ is an epistemic model and $\mathit{Exp} : S \to \mathcal{L}_{obs}$ is an expected observation function assigning to each world an observation expression π such that $\mathcal{L}(\pi) \neq \emptyset$ (non-empty set of finite sequences of observations). A pointed epistemic expectation model is a pair $(\mathcal{M}, s)$ where $\mathcal{M} = \langle S, \sim, V, \mathit{Exp}\rangle$ is an epistemic expectation model and $s \in S$.*

Intuitively, Exp assigns to each world a set of potential or expected observations. We now provide the model definitions of the examples mentioned in the introduction. The traffic light example where only one agent (the traveller) is involved can be depicted by the model $\mathcal{M}_{tl}$ (cf. Figure 2). Unless the traveller (T) observes the respective sequences of traffic signals, she would not know whether she is in France (f) or not ($\neg f$). Her uncertainty is represented by the (bidirectional) link between the two worlds s and t. For the sake of brevity, we do not draw the reflexive arrows. Similar representations are used in the message-passing example as well (cf. Figure 3). Here, the receiver would get to know about the decision depending on the message he receives, whereas, the attacker would be ignorant of the fact irrespective of the message (m or m') she receives.

The main idea for introducing this logic was to reason about agent knowledge via the matching of observations and expectations. In line of public announcement logic [Plaza, 2007], it is assumed that when a certain phenomenon is observed, people delete some impossible scenarios where they

[3]For a more detailed explanation of these concepts, see the full arXiv version [Chakraborty *et al.*, 2022].

Figure 2: $\mathcal{M}_{tl}$ (the traffic light model).

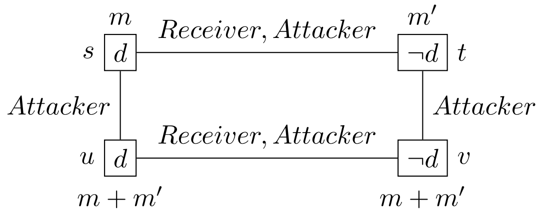

Figure 3: $\mathcal{M}_{mp}$ (the message passing model).

would not expect that observation to happen. To this end, the update of epistemic expectation models according to some observation $w \in \mathbf{\Sigma}^*$ is defined below. The idea behind an updated expectation model is to delete the worlds where the observation w could not happen.

Definition 3 (Update by observation)**.** *Let w be an observation over $\mathbf{\Sigma}$ and let $\mathcal{M} = \langle S, \sim, V, Exp\rangle$ be an epistemic expectation model. The updated model $\mathcal{M}|_w = \langle S', \sim', V', Exp'\rangle$ is defined by: $S' = \{s \in S \mid \mathcal{L}(Exp(s)\backslash w) \neq \emptyset\}$, $\sim'_i = \sim_i|_{S'\times S'}$, $V' = V|_{S'}$, and for all $s \in S'$, $Exp'(s) = Exp(s)\backslash w$.*

In Definition 3, S' is the set of worlds s in S where the word w can be observed, i.e., $\mathcal{L}(Exp(s)\backslash w) \neq \emptyset$. The definitions of $\sim'$ and V' are given by usual restrictions to S'. The expectation at each world in S' gets updated by observing the word w: finite strings of actions that are of the form wu are replaced by u while strings that are not of the form wu get removed because they do not match the expectation.

Example 2. *Consider the model $\mathcal{M}_{tl}$ of Figure 2 and $w = garga$. The updated model $\mathcal{M}_{tl}|_w = \langle S', \sim', V', Exp'\rangle$ is such that $S' = \{s\}$: world t is removed because $garga$ is not a prefix of any word in $\mathcal{L}((g^*ar^*a)^*)$. The expectation $Exp(s)$ is replaced by $Exp'(s) = (g^*ar^*)^*\backslash(garga) = r^*(g^*ar^*)^*$.*

2.3 Public Observation Logic (POL)

To reason about agent expectations and observations, the language for POL is provided below.

Definition 4 (Syntax)**.** *The* formulas φ *of* POL *are given by:*

$$\varphi \quad ::= \quad \top \mid p \mid \neg\varphi \mid \varphi \wedge \varphi \mid K_i\varphi \mid [\pi]\varphi,$$

where $p \in \mathbf{P}$, $i \in \mathbf{I}$, and $\pi \in \mathcal{L}_{obs}$.

Intuitively, $K_i\varphi$ says that 'agent i knows that φ', and $[\pi]\varphi$ says that 'after any observation in π, φ holds'. The other propositional connectives are defined in the usual manner. We also define $\langle\pi\rangle\varphi$ as $\neg[\pi]\neg\varphi$ and $\hat{K}_i\varphi$ as $\neg K_i\neg\varphi$. We will mostly use these modalities in our proofs. The Star-Free **fragment of** POL is the set of formulas in which the π's do not contain any Kleene star $*$. A much more restricted version is the Word **fragment of** POL, where π's are words. The Existential **fragment of** POL is the set of formulas for which there is an odd number of negations in front of K_i and $[\pi]$ modalities. Equivalently, it corresponds to formulas in negative normal form in which only the operators $\langle\pi\rangle$ and $\hat{K}_i$ appear. Finally, we have the Star-Free Existential **fragment of** POL which is the Existential fragment with the extra guarantee that the π's do not contain any Kleene star $*$.

Definition 5 (Truth definition)**.** *Given an epistemic expectation model $\mathcal{M} = \langle S, \sim, V, Exp\rangle$, a world $s \in S$, and a* POL*-formula φ, the truth of φ at s, denoted by $\mathcal{M}, s \vDash \varphi$, is defined by induction on φ as follows:*

$$\begin{array}{rcl}
\mathcal{M}, s \vDash p & \Leftrightarrow & p \in V(s) \\
\mathcal{M}, s \vDash \neg\varphi & \Leftrightarrow & \mathcal{M}, s \nvDash \varphi \\
\mathcal{M}, s \vDash \varphi \wedge \psi & \Leftrightarrow & \mathcal{M}, s \vDash \varphi \text{ and } \mathcal{M}, s \vDash \psi \\
\mathcal{M}, s \vDash K_i\varphi & \Leftrightarrow & \text{for all } t : (s \sim_i t \text{ implies } \mathcal{M}, t \vDash \varphi) \\
\mathcal{M}, s \vDash [\pi]\varphi & \Leftrightarrow & \text{for all observations } w \text{ over } \Sigma, \\
& & w \in \mathcal{L}(\pi) \cap prefixes(Exp(s)) \\
& & \text{implies } \mathcal{M}|_w, s \vDash \varphi
\end{array}$$

The truth of $K_i\varphi$ at s follows the standard possible world semantics of epistemic logic. The formula $[\pi]\varphi$ holds at s if for every observation w in the set $\mathcal{L}(\pi)$ that matches with the beginning of (i.e., is a prefix of) some expected observation in s, φ holds at s in the updated model $\mathcal{M}|_w$. Note that s is a world in $\mathcal{M}|_w$ because $w \in prefixes(Exp(s))$. Similarly, the truth definition of $\langle\pi\rangle\varphi$ can be given as follows:

$$\mathcal{M}, s \vDash \langle\pi\rangle\varphi \text{ iff there exists } w \in \mathcal{L}(\pi) \cap prefixes(Exp(s)) \text{ such that } \mathcal{M}|_w, s \vDash \varphi$$

Intuitively, the formula $\langle\pi\rangle\varphi$ holds at s if there is an observation w in $\mathcal{L}(\pi)$ that matches with the beginning of some expected observation in s, and φ holds at s in the updated model $\mathcal{M}|_w$. For the examples described earlier, we have:

- $\mathcal{M}_{tl}, s \models [g^*]\neg(K_T f \vee K_T \neg f)$. This example corresponds to a safety property: there is no leak of information when observing an arbitrary number of g's because it is compatible with both the expectation $g^*ar^*)^*$ of the French traffic light system, and the expectation $g^*ar^*a)^*$ of the non-French one.
- $\mathcal{M}_{tl}, s \models \langle(garg)^*\rangle(K_T f)$. This example in the Existential fragment shows that we can express the existence of a sequence of observations that reveals that the traveller is in France.
- $\mathcal{M}_{tl}, s \models \langle gar\rangle\neg(K_T f \vee K_T \neg f)$. This example in the Word fragment expresses that the sequence of observations gar would keep the traveller ignorant about her whereabouts.
- $\mathcal{M}_{mp}, s \models \langle m\rangle((K_R d \wedge \neg K_A d)$. This example, also in the Word fragment, expresses that after receiving the message m, the receiver gets to know about the decision but the attacker remains ignorant.

Model Checking for POL**.** Given a finite pointed epistemic expectation model $\mathcal{M}, s$, and a formula φ, does $\mathcal{M}, s \models \varphi$? We are interested in knowing the complexity of this problem. We will also consider restrictions of the model checking when the input formula φ is restricted to be in one of the syntactic fragments: Word, Star-Free, Existential and Star-FreeExistential.

3 Complexity Results

The main complexity result that we prove is given below. For all the proof details, see [Chakraborty *et al.*, 2022].

Theorem 1. POL *model checking is* PSPACE*-complete.*

POL **Model Checking Is in** PSPACE**.** For proving the upper bound result, that is, showing that POL model checking is in PSPACE, we design the algorithm mcPOL (Algorithm 1). It takes as input a POL model $\mathcal{M} = \langle S, \sim, V, Exp \rangle$, an initial starting world $s \in S$, and a POL formula φ and returns True iff $\mathcal{M}, s \models \varphi$. We also prove mcPOL uses polynomial space. The recursive algorithm mcPOL is divided into various cases depending on the structure of φ. The subtle case is the observation modality $\langle \pi \rangle \psi$ (that is dealt with in lines 7 to 11). It follows from the truth definition that $\mathcal{M}, s \vDash \langle \pi \rangle \psi$ iff there exists a $w \in \mathcal{L}(\pi)$ such that $\mathcal{M}|_w, s \vDash \psi$. We observe that for any $\mathcal{M}$ and w the model $\mathcal{M}|_w$ can be represented by a string of size polynomial in the size of $\mathcal{M}$ (This is because $\mathcal{M}$ and $\mathcal{M}|_w$ just differ by their expected observation functions as follows: for any world t, $Exp'(t) = Exp(t) \backslash w$ and $Exp(t)$ share the same Non-deterministic Finite Automata (NFA), just the set of initial states is different.). Thus if we consider the set $\Gamma^{\mathcal{M}} = \{\mathcal{M}|_w \mid w \in \Sigma^*\}$, the set of every updated model $\mathcal{M}|_w$, for a POL model $\mathcal{M}$, over all $w \in \Sigma^*$, we realize that all the models in $\Gamma^{\mathcal{M}}$ has size polynomial in the size of $\mathcal{M}$. Thus, by using both the observations together, when mcPOL has to check if $\mathcal{M}, s \vDash \langle \pi \rangle \psi$ (in the **for** loop in lines 8 to 10) it goes over all models $\mathcal{M}'$ in $\Gamma^{\mathcal{M}}$ and (in line 8) checks if $\mathcal{M}' = \mathcal{M}|_w$ for some $w \in \mathcal{L}(\pi)$ and finally (in line 10) calls mcPOL recursively to check if $\mathcal{M}|_w, s \vDash \psi$. Thus mcPOL needs to call a polynomial space subroutine to check if $\mathcal{M}' = \mathcal{M}|_w$ for some word $w \in \mathcal{L}(\pi)$. To prove that there exists such a polynomial space algorithm we present a slightly convoluted argument. Algorithm 2 provides a non-deterministic procedure running in polynomial space for deciding that $\mathcal{M}' = \mathcal{M}|_w$ for some word $w \in \mathcal{L}(\pi)$. By Savitch's theorem [Savitch, 1970] which states that NPSPACE = PSPACE, we have that a polynomial space algorithm also exists. Algorithm 2 starts by guessing a word of exponential length, sufficiently long enough to explore all subsets of current states for NFAs of $Exp(t)$ for all worlds t in $\mathcal{M}$ and for the NFA of π. Then the algorithm guesses the word w letter by letter and it progresses in the NFAs (note that it does not store the word w as it can be of exponential length). Algorithm 2 accepts when $w \in \mathcal{L}(\pi)$ (i.e., $\epsilon \in \mathcal{L}(\pi')$) and $\mathcal{M} = \mathcal{M}'$. Otherwise, it rejects.

Model Checking for POL **Is** PSPACE**-Hard.** Interestingly, there are two sources for the model checking to be PSPACE-hard: Kleene star in observation modalities as well as alternations in modalities (sequences of nested existential and universal modalities). We prove the PSPACE-hardness of model checking against the Existential fragment and the Star-Free fragment of POL respectively.

Theorem 2. *The model checking for the* Existential *fragment of* POL *is* PSPACE*-hard.*

Theorem 3. *The model checking for* POL *is* PSPACE*-hard, when the* POL *formulas are* Star-Free*.*

Algorithm 1 mcPOL

Input: $\mathcal{M} = \langle S, \sim, V, Exp \rangle$, $s \in S$, φ
Output: True iff $\mathcal{M}, s \vDash \varphi$

1: **if** $\varphi = p$ is a propositional variable **then**
2: return True if $p \in V(s)$; False otherwise
3: **if** $\varphi = \neg\psi$ **then**
4: return not mcPOL$(\mathcal{M}, s, \psi)$
5: **if** $\varphi = \psi' \vee \psi$ **then**
6: return mcPOL$(\mathcal{M}, s, \psi)$ or mcPOL$(\mathcal{M}, s, \psi')$
7: **if** $\varphi = \langle \pi \rangle \psi$ **then**
8: **for** all models $\mathcal{M}'$ in $\Gamma^{\mathcal{M}}$ **do**
9: **if** s is a world in $\mathcal{M}'$ and the oracle claims that $\mathcal{M}' = \mathcal{M}|_w$ for some word $w \in \mathcal{L}(\pi)$ **then**
10: return mcPOL$(\mathcal{M}', s, \psi)$
11: return False
12: **if** $\varphi = \hat{K}_i \psi$ **then**
13: **if** $\exists t \in S$ such that $t \sim_i s$ and mcPOL$(\mathcal{M}, t, \psi)$ **then**
14: return True
15: **else**
16: return False

Algorithm 2 Non-deterministic procedure to decides that $\mathcal{M}' = \mathcal{M}|_w$ for some word $w \in \mathcal{L}(\pi)$

Input: $\mathcal{M} = \langle S, \sim, V, Exp \rangle$, $\mathcal{M}' \in \Gamma^{\mathcal{M}}$, π
Output: has an accepting execution iff $\mathcal{M}' = \mathcal{M}|_w$ for some $w \in \mathcal{L}(\pi)$

$\pi' := \pi$
for $i = 1$ to $2^{\pi} \times \Pi_{t \in S} 2^{|Exp(t)|}$ **do**
if $\epsilon \in \mathcal{L}(\pi')$ and $\mathcal{M} = \mathcal{M}'$ **then**
accept
guess a letter a from Σ
$\pi' := \pi' \backslash a$
for each world t in S **do**
$Exp(t) := Exp(t) \backslash a$ // we modify $\mathcal{M}$ locally
reject

Model Checking for Star-Free Existential **and** Word **Fragment of** POL**.** While Theorems 2 and 3 proved the PSPACE-hardness of the model checking for the Existential fragment and the Star-Free fragment of POL, respectively, if we consider the Star-Free Existential fragment then we can show that the model checking is NP-complete. Finally, we also prove that the model checking for the Word fragment is in P.

Theorem 4. *The model checking problem for the* Star-Free Existential *fragment of* POL *is* NP*-complete.*

Theorem 5. *Model checking for the* Word *fragment is in* P.

4 Application

Let us consider an automatic farming drone that is moving in a field represented as a grid (see Figure 5). Two agents a and b help the farming drone. The system is adaptive so the global behaviour is not hard-coded but learned. We suppose that the drone moves on a grid and agents a and b may observe one of the four directions: $\mathbf{\Sigma} := \{\blacktriangleright, \blacktriangleleft, \blacktriangle, \blacktriangledown\}$. For instance,

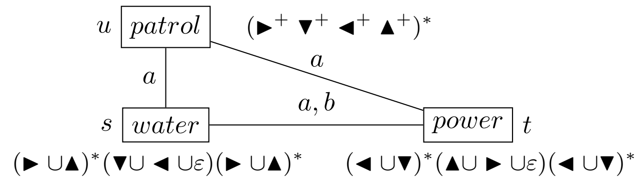

Figure 4: Model describing the initial knowledge of the two agents a and b about the expectation of the automatic farming drone.

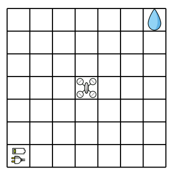

Figure 5: Field and an automatic farming drone.

observing $\blacktriangleleft$ means that the drone moves one-step left. For this example, we suppose that agent a has learned that there are three possible expectations for the drone:

1. the drone may go up-right searching for water, but the drone can make up to one wrong direction ($\blacktriangledown$ or $\blacktriangleleft$). The corresponding set of expectations is captured by the regular expression $(\blacktriangleright \cup \blacktriangle)^*(\blacktriangledown \cup \blacktriangleleft \cup \varepsilon)(\blacktriangleright \cup \blacktriangle)^*$ where ε stands for the empty word regular expression.
2. the drone may go down-left searching for power supply, but the drone can make up to one wrong direction ($\blacktriangle$ or $\blacktriangleright$). The corresponding set of expectations is captured by the regular expression: $(\blacktriangleleft \cup \blacktriangledown)^*(\blacktriangle \cup \blacktriangleright \cup \varepsilon)(\blacktriangleleft \cup \blacktriangledown)^*$.
3. the drone is patrolling making clockwise squares. The expectation is: $(\blacktriangleright^+ \blacktriangledown^+ \blacktriangleleft^+ \blacktriangle^+)^*$.

The regular expressions may be learned by the agents after observing several executions (see for instance [Balcázar *et al.*, 1997]) or might be computed by planning techniques [Bonet *et al.*, 2009]. Agent b has more information and knows that the behaviour of the drone would include either searching for water or power supply. Agent a is programmed so that if she knows that the drone is searching for water ($K_a water$) then she will turn on the valve, and if she knows that the drone is searching for power ($K_a power$), she will prepare the power supply. Agent b is programmed in the same way. The model $\mathcal{M}$, depicted in Figure 4, can be obtained by techniques described in [van Ditmarsch *et al.*, 2014] (they use a mechanism from DEL for constructing the epistemic expectation model, by assigning the expectations at each world).

The verification tasks related to epistemic planning (e.g., verifying whether $K_a water$ is true after some observations) reduce to the POL model checking problem. Let us now discuss the expressivity of the fragments: Word, Existential, Star-Free − Existential and Star-Free.

In the Word fragment, words are fixed sequences of observations. The fragment thus enables to write formulas of the form $\langle w \rangle \varphi$, meaning that φ holds after the sequence w of observations (that can be considered as the observations produced by the plan executed by the system). Thus, this fragment enables to write formulas to verify properties after the execution of a plan.

Example 3 (verification of a plan, Word fragment)**.** *Does agent a know that the drone is searching for water after the sequence $\blacktriangleright\blacktriangleright\blacktriangleright$?*

$$\mathcal{M}, s \models \langle \blacktriangleright\blacktriangleright\blacktriangleright \rangle K_a water$$

Epistemic planning is the general problem of verifying whether there exists a plan leading to a state satisfying a given epistemic formula. In our setting, it can be expressed by a formula of the form $\langle \pi \rangle \varphi$ where π denotes the plan search space (more precisely the search space of sequences of observations produced by a plan).

Example 4 (epistemic planning, Existential fragment)**.** *Does there exist a plan for the drone such that agent b would know that the drone is searching for water while agent a would still consider patrolling a possibility?*

$$\mathcal{M}, s \models \langle (\blacktriangleright \cup \blacktriangledown \cup \blacktriangleleft \cup \blacktriangle)^* \rangle (K_b water \wedge \hat{K}_a patrolling)$$

In planning (and also in epistemic planning), we may ask for the existence of a plan of bounded length, e.g., less than 4 actions. The Star-FreeExistential fragment is sufficiently expressive to tackle the so-called *bounded* epistemic planning.

Example 5 (bounded epistemic planning, Star-Free Existential fragment)**.** *Does there exist a sequence of at most 4 moves such that agent b would know that the drone is searching for water while agent a would still consider patrolling a possibility?*

$$\mathcal{M}, s \models \langle (\blacktriangleright \cup \blacktriangledown \cup \blacktriangleleft \cup \blacktriangle \cup \epsilon)^4 \rangle (K_b water \wedge \hat{K}_a patrolling)$$

Interestingly, the Star-Free fragment and the full language are able to express properties, mixing existence and non-existence of plans, in respectively the bounded and unbounded cases.

Example 6 (Star-Free fragment)**.** *Agent a would not gain the knowledge that the drone will search for water with less than or equal to 2 movements but it is possible with 3 movements:*

$$\mathcal{M}, s \models [(\blacktriangleright \cup \blacktriangledown \cup \blacktriangleleft \cup \blacktriangle)^2] \neg K_a water] \wedge \langle (\blacktriangleright \cup \blacktriangledown \cup \blacktriangleleft \cup \blacktriangle)^3 \rangle K_a water$$

Example 7 (full language)**.** *It is impossible for the agent a to know that the drone is searching for water with only down and left movements but there is a plan if all movements are allowed:*

$$\mathcal{M}, s \models [(\blacktriangledown \cup \blacktriangleleft)^*] \neg K_a water] \wedge \langle (\blacktriangleright \cup \blacktriangledown \cup \blacktriangleleft \cup \blacktriangle)^* \rangle K_a water$$

On Implementation. The model checking for the Word fragment can be implemented in poly-time with a bottom-up traversal of the parse tree of the formula, as for CTL [Baier and Katoen, 2008, Section 6.4]. The model checking for the Star-FreeExistential fragment can be implemented via a reduction to SAT. The idea is to introduce propositional variables to model the Boolean values of the following statements: (i) the t-th letter of the guessed word is equal to a, (ii) a given automaton A is in state q after having read the first t letters of the guessed word, and, (iii) a subformula of the formula φ to check is true at a given world u. The last type of statements are combined in the spirit of the Tseitin transformation [Ben-Ari, 2012, p. 91] (for details, see [Chakraborty *et al.*, 2022]and https://github.com/francoisschwarzentruber/polmc).

5 Related Work

Dynamic Epistemic Reasoning. The model checking of standard epistemic logic (EL) is PTIME-complete [Schnoebelen, 2002]. Public Observation Logic (POL) is quite similar to Public announcement logic (PAL) [Plaza, 2007]. When public announcements are performed, the number of possible worlds reduces, making the model checking of PAL still in PTIME [van Benthem, 2011] as for standard epistemic logic. When actions can be private, the model checking becomes PSPACE-complete for DEL with action models [Aucher and Schwarzentruber, 2013]. In PAL, a possible world is equipped with a valuation, while in POL it is also equipped with a regular expression denoting the expectation in that world. In PAL, the public announcement is fully specified and its effect is deterministic. In POL, we may reason on sets of possible observations represented by regular expressions π. When these sets are singletons, we again obtain a PTIME upper bound (Theorem 5). In this sense, POL is close to Arbitrary PAL (APAL) [French and van Ditmarsch, 2008] whose model checking is also PSPACE-complete [Ågotnes *et al.*, 2010]. In APAL, any epistemic formula can be announced: there are no expectations. However, in POL, we have to reason about the constraints between the possible expectations, and the set of observations (given by π). Our contribution can be reformulated as follows: we prove that (i) reasoning about these constraints can still be done in PSPACE, and, (ii) this reasoning is sufficiently involved for the model checking to be PSPACE-hard. In POL, regular expressions are used to represent sets of observations, while van Benthem et al. [van Benthem *et al.*, 2006] used regular expressions (actually, programs of Propositional dynamic logic (PDL) [Fischer and Ladner, 1979]) to denote epistemic relations. Charrier et al. [Charrier *et al.*, 2019] considered a logic for reasoning about protocols where actions are public announcements and not abstract observations as in POL: in this sense, POL is more general.

Epistemic Temporal Reasoning. It is natural to describe computational behaviours with regular expressions. Finite-state controllers, i.e., automata are used to describe policies in planning [Bonet *et al.*, 2009]. Interestingly, Lomuscio and Michaliszyn [Lomuscio and Michaliszyn, 2016] studied an epistemic logic where formulas are evaluated on intervals and the language provides Allen's operators on intervals: in their setting, the model is an interpreted system, and a propositional variable p is true in an interval I if the trace of I matches a given regular expression associated to p. In contrast, POL is not based on an already set-up model but relies on updates in a model. Bozzelli et al. [Bozzelli *et al.*, 2017] studied the complexity of the model checking of that logic depending on the restrictions on the allowed set of Allen's operators. Their framework is similar to ours because it relies on regular expressions but the approach is orthogonal to model updates and hence, to epistemic planning.

Epistemic Planning. Epistemic planning frameworks (based on DEL [Bolander *et al.*, 2020], or the so-called MEP for Multi-agent Epistemic Planning [Muise *et al.*, 2022]) all provide a mechanism for reasoning about preconditions and effects of actions. Expectations about others or about the world are not dealt with. However, Saffidine et al. [Saffidine *et al.*, 2018] propose a collaborative setup for epistemic planning where each agent executes its own knowledge-based policy/program (KBP) while agents commonly know all the KBPs that are being executed, meaning that agents expect that the other agents follow their own KBP. On the contrary, in POL, observations are public but expectations are in general not commonly known. Reasoning about some epistemic properties that are true after the execution of any kind of KBPs is undecidable, but is PSPACE-complete for star-free KBPs. The complexity is high for different reasons: the initial model is represented symbolically; observations are not already public, and KBPs may contain tests.

Strategic Reasoning. Usually in logics for strategic reasoning (e.g., alternating-time temporal logic [Alur *et al.*, 2002] and strategy logic [Chatterjee *et al.*, 2010]), agents do not have expectations: an agent may consider all possible strategies for the others. Recently, Belnardinelli et al. [Belardinelli *et al.*, 2021] propose a variant of strategy logic (SL) where a player may know completely the strategy of another player. In contrast, in POL agents may have partial information about the expectations. In POL, agents also have higher-order knowledge about these expectations. In SL, strategies are abstract objects in the logical language whereas in POL, observations are represented as composite structures that the agents can reason about, similar to the work on games and strategies presented in [Ghosh and Ramanujam, 2012]. In this sense, POL can be seen as EL extended with PDL operators.

6 Conclusion

In this paper, we showed that the model checking for POL is PSPACE-complete. Such complexity studies were left open in [van Ditmarsch *et al.*, 2014]. We also identified more tractable fragments (see Figure 1) of POL. Finally, we discussed the applicability of our study in verifying various features of interactive systems related to epistemic planning. A discussion on implementation is also provided.

We leave the investigations on model checking for EPL, an extension of POL, also proposed in [van Ditmarsch *et al.*, 2014], for future work. We also aim to study the satisfiability problems of POL and EPL by adapting the techniques from [Aucher and Schwarzentruber, 2013; Lutz, 2006].

Many interesting features of such interactive systems remain to be investigated : private observations, like in DEL with action models [van Ditmarsch *et al.*, 2008]; dynamic aspects (e.g., changing expectations); richer languages of expectations (e.g., context-free grammars for expectations), among others. Symbolic model checking can be considered as well following the trends of [van Benthem *et al.*, 2018] and [Charrier *et al.*, 2019].

This paper also opens up a research avenue for developing variants and extensions for reasoning about expectations and observations that can be expressive enough with reasonable complexities for the model checking problem.

To sum up, POL mixes epistemic logic and language theory for modelling mechanisms of social intelligent agents, and the current investigations on model checking set it up as a useful tool in building social software for AI.

References

[Ågotnes *et al.*, 2010] T Ågotnes, P Balbiani, H van Ditmarsch, and P Seban. Group announcement logic. *J. Appl. Log.*, 8(1):62–81, 2010.

[Alur *et al.*, 2002] R Alur, T A Henzinger, and O Kupferman. Alternating-time temporal logic. *JACM*, 49(5):672–713, 2002.

[Aucher and Schwarzentruber, 2013] G Aucher and F Schwarzentruber. On the complexity of dynamic epistemic logic. *TARK*, 2013.

[Baier and Katoen, 2008] C Baier and J-P Katoen. *Principles of model checking*. MIT Press, 2008.

[Balcázar *et al.*, 1997] J L Balcázar, J Díaz, and R Gavaldà. Algorithms for learning finite automata from queries: A unified view. In *Advances in Algorithms, Languages, and Complexity*, pages 53–72, 1997.

[Belardinelli *et al.*, 2021] F Belardinelli, S Knight, A Lomuscio, B Maubert, A Murano, and S Rubin. Reasoning about agents that may know other agents' strategies. In *IJCAI*, pages 1787–1793, 2021.

[Ben-Ari, 2012] M Ben-Ari. *Mathematical Logic for Computer Science, 3rd Edition*. Springer, 2012.

[Bolander *et al.*, 2020] T Bolander, T Charrier, S Pinchinat, and F Schwarzentruber. DEL-based epistemic planning: Decidability and complexity. *Artif. Intell.*, 287:103304, 2020.

[Bonet *et al.*, 2009] B Bonet, H Palacios, and H Geffner. Automatic derivation of memoryless policies and finite-state controllers using classical planners. In *ICAPS*, 2009.

[Bozzelli *et al.*, 2017] Bozzelli, A Molinari, A Montanari, and A Peron. An in-depth investigation of interval temporal logic model checking with regular expressions. *SEFM*, 10469:104–119, 2017.

[Chakraborty *et al.*, 2022] S Chakraborty, A Ghosh, S Ghosh, and F Schwarzentruber. On verifying expectations and observations of intelligent agents. *CoRR*, abs/2205.00784, 2022.

[Charrier *et al.*, 2019] T Charrier, S Pinchinat, and F Schwarzentruber. Symbolic model checking of public announcement protocols. *JLC*, 29(8):1211–1249, 2019.

[Chatterjee *et al.*, 2010] K Chatterjee, T A Henzinger, and N Piterman. Strategy logic. *Inf. Comput.*, 208(6):677–693, 2010.

[de Weerd *et al.*, 2017] H de Weerd, R Verbrugge, and B Verheij. Negotiating with other minds: the role of recursive theory of mind in negotiation with incomplete information. *AAMAS*, 31(2):250–287, 2017.

[Fagin *et al.*, 1995] R Fagin, J Y Halpern, Y Moses, and M Y Vardi. *Reasoning about Knowledge*. MIT Press, 1995.

[Fischer and Ladner, 1979] M J Fischer and R E Ladner. Propositional dynamic logic of regular programs. *JCSS*, 18(2):194–211, 1979.

[French and van Ditmarsch, 2008] T French and H van Ditmarsch. Undecidability for arbitrary public announcement logic. In *AiML*, pages 23–42, 2008.

[Ghosh and Ramanujam, 2012] S Ghosh and R Ramanujam. Strategies in games: A logic-automata study. *LNCS*, 7388:110–159, 2012.

[Lomuscio and Michaliszyn, 2016] Alessio Lomuscio and Jakub Michaliszyn. Model checking multi-agent systems against epistemic HS specifications with regular expressions. *KR*, pages 298–308, 2016.

[Lutz, 2006] C Lutz. Complexity and succinctness of public announcement logic. *AAMAS*, pages 137–143, 2006.

[Muise *et al.*, 2022] C Muise, V Belle, P Felli, S A McIlraith, T Miller, A R Pearce, and L Sonenberg. Efficient multi-agent epistemic planning: Teaching planners about nested belief. *Artificial Intelligence*, 302, 2022.

[Plaza, 2007] J Plaza. Logics of public communications. *Synthese*, 158(2):165–179, 2007.

[Saffidine *et al.*, 2018] A Saffidine, F Schwarzentruber, and B Zanuttini. Knowledge-based policies for qualitative decentralized pomdps. *AAAI*, pages 6270–6277, 2018.

[Savitch, 1970] W J Savitch. Relationships between nondeterministic and deterministic tape complexities. *JCSS*, 4(2):177–192, 1970.

[Schnoebelen, 2002] P Schnoebelen. The complexity of temporal logic model checking. *AiML*, pages 393–436, 2002.

[van Benthem *et al.*, 2006] J van Benthem, J van Eijck, and B P Kooi. Logics of communication and change. *Information and Computation*, 204(11):1620–1662, 2006.

[van Benthem *et al.*, 2018] J van Benthem, J van Eijck, M Gattinger, and K Su. Symbolic model checking for dynamic epistemic logic - S5 and beyond. *JLC*, 28(2):367–402, 2018.

[van Benthem, 2011] J van Benthem. *Logical dynamics of information and interaction*. Cambridge University Press, 2011.

[van Ditmarsch *et al.*, 2008] H van Ditmarsch, W van der Hoek, and B Kooi. *Dynamic Epistemic Logic*, volume 337 of *Synthese Library*. Springer, 2008.

[van Ditmarsch *et al.*, 2014] H van Ditmarsch, S Ghosh, R Verbrugge, and Y Wang. Hidden protocols: Modifying our expectations in an evolving world. *Artificial Intelligence*, 208:18–40, 2014.

[Wang, 2011] Y Wang. Reasoning about protocol change and knowledge. *ICLA*, 6521:189–203, 2011.