Proceedings of the 22nd International Conference on Principles of Knowledge Representation and Reasoning
Main Track

Reasoning about Knowledge on Regular Expressions Is 2EXPTIME-complete

Avijeet Ghosh[1] , **Sujata Ghosh**[2] , **François Schwarzentruber**[3]
[1]Chennai Mathematical Institute, Chennai
[2]Indian Statistical Institute, Chennai
[3]ENS de Lyon, France
avi.ghosh23@gmail.com, sujata@isichennai.res.in, francois.schwarzentruber@ens-lyon.fr

Abstract

Logics for reasoning about knowledge and actions have seen many applications in various domains of multi-agent systems, including epistemic planning. Change of knowledge based on observations about the surroundings forms a key aspect in such planning scenarios. Public Observation Logic (POL) is a variant of public announcement logic for reasoning about knowledge that gets updated based on public observations. Each state in an epistemic (Kripke) model is equipped with a set of expected observations. These states evolve as the expectations get matched with the actual observations. In this work, we prove that the satisfiability problem of POL is 2EXPTIME-complete.

1 Introduction

Intelligent artificial agents are being used for performing various tasks including planning and scheduling in real life, from simple to more complicated ones. For example, a robot may try to move from a point to another by overcoming certain hurdles. Or, it may try to keep an eye over the surroundings without other agents *knowing* about it. Accomplishing a goal in such surveillance activities may involve the robot's knowledge about other agents' knowledge. Automated planning (Ghallab, Nau, and Traverso 2004) is a branch of study in multi-agent systems that involves deciding whether a sequence or plan of actions exists to attain some goal. An extension of such planning studies is termed as *epistemic planning* (Bolander 2017), where the goal, like in the case of surveillance robot, involves knowledge of multiple agents.

Reasoning about knowledge using logical systems has been studied extensively in the domain of modal logic (Blackburn, de Rijke, and Venema 2001), more specifically using epistemic logic (Fagin et al. 1995). In addition, reasoning about the knowledge dynamics of agents has been studied using dynamic epistemic logic (DEL) (van Ditmarsch, van der Hoek, and Kooi 2007), among others. Evidently, a popular approach in epistemic planning is the use of model-checking problem in DEL (Bolander 2017). However, this problem turns out to be undecidable when one considers finite iterations of such actions in DEL (Aucher and Bolander 2013). Moreover, it is shown that for the general epistemic planning tasks, the plan existence problem is already undecidable with two agents.

Public observation logic (POL) (van Ditmarsch et al. 2014) deals with *expected* observations (actions) that are associated with each state in an epistemic model (Fagin et al. 1995), and are represented by regular expressions. The model gets updated depending on the matching of actual and expected observations, and accordingly, agents' knowledge gets updated as well. This dynamic behavior based on some sequence (finite iteration) of observations inherently makes this setting useful in reasoning about various concepts involving knowledge and actions, for example, epistemic planning.

In (Chakraborty et al. 2023), we show that public announcement logic (PAL) with propositional announcements is closely related to the *word* fragment of POL, where the regular expressions describing observations are only in the word form. In this sense, POL can be considered as a dynamic logic-like extension of PAL, taking iteration of the Boolean announcements under its wings. Thus, it is worthwhile to check whether POL is decidable, since PAL with iterated announcements is undecidable (Miller and Moss 2005). Note that the decidability (van Ditmarsch and French 2022) of arbitrary PAL with Boolean announcements (BAPAL) provides a push towards an affirmative response to our query. Before moving forward, let us provide a scenario that can be modeled by POL.

Example 1. *Consider a surveillance drone hovering over the boundary zone between two territories, T_1 and T_2, say, in conflict with each other. Suppose the drone is deployed by T_1, and if it is detected in the airspace of T_2, it might get destroyed. How would the drone differentiate between the two territories so that it can restrict itself from entering T_2? According to its expectations based on the vegetation in the area, if it observes the sequence of (spruce*-pine*-cedar-fir*)* ($*$ denotes the continuance of such sequences), it would know that it is in area T_1, while if it observes (spruce*-pine*-larch-fir*)*, it would know that it is in T_2.*

With regard to the computational behavior of POL, we explored the complexity of the model-checking problem for POL in (Chakraborty et al. 2022). In addition, we investigated the satisfiability problem for the star-free fragment of POL in (Chakraborty et al. 2023), where the observations comprise of star-free regular expressions. In this work, we complete this study by providing an answer to the remaining open problem concerning the decidability of full POL.

We show that the satisfiability problem of POL (with Kleene star) is 2EXPTIME-complete, in contrast to PAL with iterated announcements, making it a more viable option for modelling planning and related problems. The techniques used in our proofs may be of a more general interest.

For the upper bound, we start with the usual filtration argument (Blackburn, de Rijke, and Venema 2001) - if a formula is satisfiable then it is satisfiable in a model of exponential size. Although filtration is usually sufficient to prove decidability, for POL this is not the case, as the expected observations associated with each state in a model may be arbitrary (Section 3). We characterize satisfiability in terms of a finite syntactic structure defined by Hintikka sets (Smullyan 1995), which we call a *bubble transition structure*. We provide a correspondence between the expected observations and the corresponding automata (Section 4) to facilitate our study. Then in exponential time, we reduce the satisfiability problem of deterministic propositional dynamic logic (DPDL) to that of POL by encoding the constraints on the epistemic structures by propositional theories (Section 5). For the lower bound, we encode a superposition of *three* same configurations of a Turing machine in the leaves of a POL model considered as a full binary tree (Section 6). Without further ado, let us start by recalling POL.

2 Public Observation Logic

To begin with, we provide a brief overview of Public Observation Logic (van Ditmarsch et al. 2014). Let $\mathcal{P}$ be a countable set of propositional letters, Ag be a finite set of agents, and Σ be a finite alphabet of atomic actions/observations. We now introduce *observation expressions* as follows:

Definition 2 (Observation Expression)**.** *Given a finite set of observations* Σ*,* observation expressions *are defined recursively as:* $\pi := \emptyset \mid a \mid \pi + \pi \mid \pi;\pi \mid \pi^\star$*, with* $a \in \Sigma$*.*

Note that an observation expression π is a regular expression, and $\mathcal{L}(\pi)$ denotes the language corresponding to the regular expression π. We use these observation expressions to describe observations within formulas (cf. Definition 3) as well as expected observations at states in a model (cf. Definition 4). In the surveillance example, an observation may be *spruce-pine-cedar-fir-spruce* (abbreviated as *spcfs*) or *spruce-pine-larch-fir-spruce* (abbreviated as *splfs*), among others. Let us now describe the language of POL.

Definition 3 (POL Syntax)**.** *The language of* POL *can be recursively defined as:*

$$\varphi := \top \mid p \mid \varphi \vee \varphi \mid \neg\varphi \mid \hat{K}_i\varphi \mid \langle\pi\rangle\varphi$$

where $p \in \mathcal{P}$*,* $i \in Ag$*, and* π *is an observation expression over* Σ*.*

The box formulas are defined as follows: $[\pi]\varphi = \neg\langle\pi\rangle\neg\varphi$, $K_i\varphi = \neg\hat{K}_i\neg\varphi$. The formula $K_i\varphi$ is read as 'agent i knows φ', while $\hat{K}_i\varphi$ is read as 'agent i considers φ as an epistemic possibility'. The formula $\langle\pi\rangle\psi$ expresses that there is a sequence of atomic observations that matches the language of the (regular) expression π and ψ holds after the said sequence is observed publicly. For example, the formula $\langle(s^\star p^\star l f^\star)c\rangle K_d T_2$ expresses that after an observation of finite sequences of spruce and pine followed by a larch and then a finite sequence of fir, the drone knows that it is in the region T_2. We are now ready to describe the POL *models* (van Ditmarsch et al. 2014) that 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 observations.

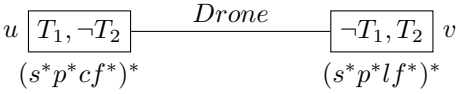


Figure 1: $\mathcal{M}_{sd}$ (the surveillance drone model)

Definition 4 (POL model)**.** *A* POL *model is a tuple* $\mathcal{M} = (S, R, V, Exp)$*, where, (i)* S *is a non-empty set of states, (ii)* $R_i \subseteq S \times S$ *is an equivalence relation for all* $i \in Ag$*.* $R = \{R_i\}_{i\in Ag}$*, (iii)* $V : S \to 2^{\mathcal{P}}$ *is a valuation function, and (iv)* $Exp : S \to RE_\Sigma$ *is an expectation function assigning an observation expression over* Σ *to each state in* S*.*

Figure 1 models the scenario discussed in the introduction. The model consists of the set of states $S = \{u, v\}$, with u representing the state of the drone hovering over the region T_1 and v representing the same for T_2. The expected observations of the drone in the two states are assigned accordingly (cf. the regular expressions assigned to u and v).

To interpret the changes in agent knowledge based on observations, POL semantics involves model updates. The observation expressions associated with the states get updated according to the sequence of actions already observed. To model this idea formally, we first explain the process of *residuation* of observation (regular) expressions with respect to words: Given a word $w \in \Sigma^\star$ and a regular expression π over Σ, $\pi\backslash w$ is a regular expression, called the *residuation* of π with w, where, $\mathcal{L}(\pi\backslash w) = \{u \mid wu \in \mathcal{L}(\pi)\}$. For example, for $a, b \in \Sigma$, $(b^\star aa(a+b)^\star)\backslash ba = a(a+b)^\star$.

Definition 5 (Model update by observation)**.** *Given a word* w *over* Σ*, the model* $\mathcal{M}|_w = (S', R', V', Exp')$ *is defined as follows: (i)* $S' = \{s \in S \mid \mathcal{L}(Exp(s)\backslash w) \neq \emptyset\}$*, (ii)* $R' = R|_{S'\times S'}$*, (iii)* $V' = V|_{S'\times S'}$*, and, (iv)* Exp' *is given by* $Exp'(s) = Exp(s)\backslash w$ *for all* $s \in S'$*.*

We are now ready to give the interpretation of the POL formulas in POL models.

Definition 6 (Truth of a POL formula)**.** *Given a model* $\mathcal{M} = (S, R, V, Exp)$ *and an* $s \in S$*, the truth definition of a* POL *formula* φ *(*$\mathcal{M}, s \vDash \varphi$*), is given as follows:*

- $\mathcal{M}, s \vDash p$ *iff* $p \in V(s)$*, where* $p \in \mathcal{P}$*.*
- $\mathcal{M}, s \vDash \neg\psi$ *iff* $\mathcal{M}, s \nvDash \psi$*.*
- $\mathcal{M}, s \vDash \psi \vee \chi$ *iff* $\mathcal{M}, s \vDash \psi$ *or* $\mathcal{M}, s \vDash \chi$*.*
- $\mathcal{M}, s \vDash \hat{K}_i\psi$ *iff there is* $t \in S$*, s. th.* sR_it *and* $\mathcal{M}, t \vDash \psi$*.*
- $\mathcal{M}, s \vDash \langle\pi\rangle\psi$ *iff there exists* $w \in \mathcal{L}(\pi)$ *such that* $\mathcal{L}(Exp(s)\backslash w) \neq \emptyset$ *and* $\mathcal{M}|_w, s \vDash \psi$*.*

The truth definitions are as usual except for the last one. The formula $\langle\pi\rangle\psi$ holds if there is an observation sequence

(a word w, say) that matches π, and after observing w (publicly), ψ holds. The dual formula $[\pi]\psi$ is interpreted accordingly. As for the drone example and its model (Figure 1) defined above, we can verify:

- $\mathcal{M}_{sd}, s \models [s^*p^*]\neg(K_dT_1 \vee K_d\neg T_1)$. This example corresponds to the drone being uncertain about its whereabouts: observing an arbitrary number of s's followed by p's is compatible with both the expectation $(s^*p^*cf^*)^*$ of the T_1 vegetation, and the expectation $(s^*p^*lf^*)^*$ of the non-T_1 vegetation.
- $\mathcal{M}_{sd}, s \models \langle s^*p^*c\rangle(K_dT_1)$. This example expresses the existence of a sequence of observations that reveals that the drone is in the region T_1.

Satisfiability Problem. The satisfiability problem of POL is as follows: Given a POL formula φ, does there exist a POL model $\mathcal{M}$ and a state s in it such that $\mathcal{M}, s \models \varphi$? In what follows, we show that the POL satisfiability problem is decidable and explore the complexity of the problem.

3 Finite Model Property

As a first step we show the following result:

Theorem 7 (Finite model property)**.** *If φ is satisfiable then φ is satisfied in a* POL *model with $2^{O(|\varphi|)}$ states.*

3.1 Closure Sets

Definition 8 (Fischer-Ladner Closure)**.** *The Fischer-Ladner closure of a formula φ, denoted by $FL(\varphi)$, is the smallest set containing φ and satisfying the following conditions:*

- *if $\psi \in FL(\varphi)$ and ψ not starting with $\neg$ then $\neg\psi \in FL(\varphi)$*
- *if $\neg\psi, K_i\psi, \hat{K}_i\psi, [\pi]\psi$ or $\langle\pi\rangle\psi$ in $FL(\varphi)$ then $\psi \in FL(\varphi)$*
- *if $\psi \wedge \chi$ or $\psi \vee \chi$ are in $FL(\varphi)$ then $\psi, \chi \in FL(\varphi)$*
- *if $\langle\pi_1;\pi_2\rangle\psi \in FL(\varphi)$ then $\langle\pi_1\rangle\langle\pi_2\rangle\psi \in FL(\varphi)$*
- *if $\langle\pi_1 + \pi_2\rangle\psi \in FL(\varphi)$ then $\langle\pi_1\rangle\psi, \langle\pi_2\rangle\psi \in FL(\varphi)$*
- *if $\langle\pi^\star\rangle\psi \in FL(\varphi)$ then $\langle\pi\rangle\langle\pi^\star\rangle\psi \in FL(\varphi)$*
- *if $[\pi_1;\pi_2]\psi \in FL(\varphi)$ then $[\pi_1][\pi_2]\psi \in FL(\varphi)$*
- *if $[\pi_1 + \pi_2]\psi \in FL(\varphi)$ then $[\pi_1]\psi, [\pi_2]\psi \in FL(\varphi)$*
- *if $[\pi^\star]\psi \in FL(\varphi)$ then $[\pi][\pi^\star]\psi \in FL(\varphi)$*

In simple words, the Fischer-Ladner closure of a formula constitutes all subformulas that need to be considered in a satisfiability argument to satisfy the original formula. Note that if $[\pi]\psi \in FL(\varphi)$, we have $[\pi\backslash a]\psi \in FL(\varphi)$. For example, suppose $[b^\star ab]\psi \in FL(\varphi)$. Note that $b^\star ab\backslash a = b$. By definition 8, $[b^\star][ab]\psi \in FL(\varphi)$ implies $[ab]\psi, [b][b^\star][a]\psi \in FL(\varphi)$. Since $[ab]\psi \in FL(\varphi)$, therefore $[a][b]\psi \in FL(\varphi)$, which finally implies $[b]\psi \in FL(\varphi)$. The same is also true for the diamond formulas ($\langle\pi\rangle\psi$) as well.

Example 9. *Consider $\varphi = \langle(a + b)^\star\rangle p$. Then, we have: $FL(\varphi) = \{\varphi, \neg\varphi, p, \langle a + b\rangle\langle(a + b)^\star\rangle p, \ldots\}$.*

Observation 10. *(Harel, Kozen, and Tiuryn 2000) Given φ, $|FL(\varphi)| \leq O(|\varphi|)$.*

3.2 Filtration

A standard approach for showing decidability of the satisfiability problem in modal logics is the filtration technique (Blackburn, de Rijke, and Venema 2001), which goes by proving the small model property: For any satisfiable formula, there exists a finite model satisfying it whose size can be bounded with respect to the input formula. We provide a similar argument using the following construction of the *small model*. We first define an equivalence relation $\sim$ among the states of a model $\mathcal{M} = \langle S, \{R_i\}_{i\in Ag}, V, Exp\rangle$ ($\sim \subseteq S \times S$) with respect to a formula φ as follows:

$$s \sim s' \text{ iff for all } \psi \in FL(\varphi), (\mathcal{M}, s \models \psi \text{ iff } \mathcal{M}, s' \models \psi)$$

Note that, the relation $\sim$ is reflexive, transitive and symmetric, that is, an equivalence relation over S. For any $s \in S$, we denote $[s]$ to be the equivalence class with respect to $\sim$ that contains s. Next we give the small model construction following (Blackburn, de Rijke, and Venema 2001).

Definition 11 (Small Model of a Formula)**.** *Given a model $\mathcal{M} = \langle S, \{R_i\}_{i\in Ag}, V, Exp\rangle$ and a formula φ, a small model $\mathcal{M}^\sim = \langle S^\sim, \{R_i^\sim\}_{i\in Ag}, V^\sim, Exp^\sim\rangle$ is defined as:*

- $S^\sim = \{[s] \mid s \in S\}$
- $([s], [s']) \in R_i^\sim$ *if these conditions hold:*
  1. *there exists $s_1 \in [s]$ and $s_2 \in [s']$ s. t. $(s_1, s_2) \in R_i$.*
  2. *for all $\hat{K}_i\psi \in FL(\varphi)$, $\mathcal{M}, s' \models \psi \vee \hat{K}_i\psi$ implies that $\mathcal{M}, s \models \hat{K}_i\psi$.*
- $V^\sim([s]) = V(s)$
- $Exp^\sim([s]) = Exp(s^c)$ *for some $s^c \in [s]$.*

Note that the above relation $R_i^\sim \subseteq S^\sim \times S^\sim$ is an equivalence relation. The trickier parts are the transitivity and symmetry. The former can be proved using condition (2), as discussed in (Blackburn, de Rijke, and Venema 2001). The latter, that is symmetry, can be proved using (1) and (2). For all proof details, see (Ghosh, Ghosh, and Schwarzentruber 2025). Before proving the small model property, we prove the following lemma, which shows that the choice of s^c in Definition 11 is not important. The main theorem follows.

Lemma 12. *For any formula of the form $\psi' = \langle\pi\rangle\psi \in FL(\varphi)$, if $s \sim s'$, then there exists a $w \in \mathcal{L}(\pi)$ such that [($\mathcal{M}, s \models \psi'$, s survives in $\mathcal{M}|_w$ and $\mathcal{M}|_w, s \models \psi$) iff ($\mathcal{M}, s' \models \psi'$, s' survives in $\mathcal{M}|_w$ and $\mathcal{M}|_w, s' \models \psi$)].*

Proof Sketch. The proof goes by induction on the size of π. Since s and s' satisfy the same formulas in $FL(\varphi)$, the result follows by definition 8. In particular, the inclusion of formulas within $FL(\varphi)$ that take care of the prefixes of π in $\langle\pi\rangle\psi$ plays an important role. We give a detailed proof in (Ghosh, Ghosh, and Schwarzentruber 2025).

Theorem 13. *Given a model $\mathcal{M} = \langle S, \{R_i\}_{i\in Ag}, V, Exp\rangle$ and a formula φ, for any $\psi \in FL(\varphi)$ and $w \in \Sigma^\star$,*

$$\mathcal{M}|_w, s \models \psi \text{ iff } \mathcal{M}^\sim|_w, [s] \models \psi.$$

Corollary 14. *If φ is* POL*-satisfiable, then φ is satisfiable in a* POL*-model with at most $2^{|FL(\varphi)|}$ states.*

Proof. If $\mathcal{M}, s \models \varphi$ then $\mathcal{M}^\sim, [s] \models \varphi$. Note that $\mathcal{M}^\sim$ contains at most $2^{|FL(\varphi)|}$ states. □

Although we have a small model property for POL, it does not give an immediate decidability proof for the satisfiability problem of POL. We note that in POL models, each state is associated with not only a valuation but also a regular expression. Given a satisfiable formula, the popular filtration technique (Blackburn, de Rijke, and Venema 2001) gives us a model where (i) the number of states is bounded above with respect to the size of Fischer-Ladner closure, and (ii) the number of possible valuations is bounded above due to the fact that the number of propositions is bounded by the size of the input formula. However, the size of the regular expression $Exp(s)$ associated with each state s may be arbitrarily large, and it is not straightforward to come up with a bound. We now provide a way to deal with this difficulty.

4 Unraveling the Filtrated Model

We give alternative models to POL in terms of *finite bubble transition structures* (BTS). They are *unraveled* POL models in which the expectation functions are represented with explicit transitions. Furthermore, BTS's are syntactic in nature: epistemic relations are between states labeled by *Hintikka sets* (Blackburn, de Rijke, and Venema 2001). In contrast to states in standard epistemic models, Hintikka sets also contain information about the future, e.g., if a Hintikka set contains $\langle a\rangle\hat{K}_i p$, it says after a is observed, i should consider p as possible. In addition, these BTS's are always finite. Let us now describe them.

4.1 Finite Transition Model

Hintikka sets are sets of POL formulas satisfying some conditions, as defined below.

Definition 15 (Hintikka set of Formulas)**.** *A Hintikka set H is a set of formulas satisfying following conditions:*

1. *If ψ does not start with negation, $\psi \in H$ iff $\neg\psi \notin H$.*
2. *$\psi_1 \wedge \psi_2 \in H$ iff $\{\psi_1, \psi_2\} \subseteq H$.*
3. *$\psi_1 \vee \psi_2 \in H$ iff $\psi_1 \in H$ or $\psi_2 \in H$.*
4. *If $K_i\psi \in H$ then $\psi \in H$.*
5. *If $\langle\pi_1 + \pi_2\rangle\psi \in H$ then $\langle\pi_1\rangle\psi \in H$ or $\langle\pi_2\rangle\psi \in H$.*
6. *If $\langle\pi_1\pi_2\rangle\psi \in H$ then $\langle\pi_1\rangle\langle\pi_2\rangle\psi \in H$.*
7. *If $\langle\pi^\star\rangle\psi \in H$ then either $\psi \in H$ or $\langle\pi\rangle\langle\pi^\star\rangle\psi \in H$.*
8. *If $[\pi_1 + \pi_2]\psi \in H$ then $\{[\pi_1]\psi, [\pi_2]\psi\} \subseteq H$*
9. *If $[\pi_1\pi_2]\psi \in H$ then $[\pi_1][\pi_2]\psi \in H$*
10. *If $[\pi^\star]\psi \in H$ then $\{\psi, [\pi][\pi^\star]\psi\} \subseteq H$*

Besides the usual boolean conditions in Definition 15, (4) corresponds to reflexivity of the knowledge relation. To satisfy diamond observation formulas $\langle\pi\rangle\psi$, $\mathcal{L}(\pi)$ must contain a word. Thus, (5) corresponds to the word coming from π_1 or π_2, (6) corresponds to a word from π_1, followed by one from π_2, and (7) takes care of the iteration. In (8-10), the box formulas are considered in a dual manner.

Example 16. *Consider $\varphi := K_i(\langle a\rangle(p\vee q)\wedge[a^\star]\langle a\rangle(p\vee q))$. A Hintikka set containing φ is given as follows:*

$$H = \{\varphi, \langle a\rangle(p \vee q) \wedge [a^\star]\langle a\rangle(p \vee q), \langle a\rangle(p \vee q), [a^\star]\langle a\rangle(p \vee q), [a][a^\star]\langle a\rangle(p \vee q)\}$$

The second formula in H comes from (4), the third and fourth ones are due to (2), and the last one is a consequence of (10).

Let $\mathcal{K}_i(H) = \{K_i\psi \mid K_i\psi \in H\}$ be the set of knowledge formulas in H. We now define *(epistemic) bubbles*. They are similar to POL models but differs in three ways. First, there is no expectation function Exp anymore. Second, the information about the future in a state s previously stored in $Exp(s)$ is now provided in the Hintikka set $L(s)$ attached to s. For instance if $\langle a\rangle\top \in L(s)$ it means that a can be observed in s. Note that all the necessary information about the expectation function and indistinguishability are now relational structures on Hintikka sets. Third, a bubble is tailored for the corresponding formula φ: Hintikka sets are given with respect to φ and the number of states is bounded by $2^{|FL(\varphi)|}$ (as in Corollary 14).

Definition 17. *A* bubble *wrt a* POL *formula φ is a labelled relational structure $\langle S, \{R_i\}_{i\in Ag}, L\rangle$ such that:*

1. *S is a set of (abstract) states such that $0 \leq |S| \leq 2^{|FL(\varphi)|}$*
2. *$L : S \to 2^{FL(\varphi)}$ is a labelling function such that for every $s \in S$, $L(s)$ is a Hintikka set.*
3. *$R_i \subseteq S \times S$, is a binary equivalence relation such that:*
   (a) *For any $s \in S$, any formula $\hat{K}_i\psi \in L(s)$, there exists an $s' \in S$ such that $\psi \in L(s')$ and $(s, s') \in R_i$.*
   (b) *For all $s', s'' \in [s]_i$, the equivalence class of s under R_i, $\mathcal{K}_i(L(s')) = \mathcal{K}_i(L(s''))$.*

Point 3 describes the interaction between the indistinguishability relation R_i and the Hintikka sets. If a state satisfies $\hat{K}_i\psi$, then there is an i-indistinguishable state satisfying ψ (3a). Also, knowledge of agent i (formulas of the form $K_i\psi$) are the same in all the i-indistinguishable states (3b). We now introduce the notion of *a-successor*. Considering a bubble B that intuitively corresponds to a POL model $\mathcal{M}$, an observation successor of B is a bubble B' which corresponds to $\mathcal{M}|_a$.

Definition 18. *Let $B = \langle S, \{R_i\}_{i\in Ag}, L\rangle$ and $B' = \langle S', \{R'_i\}_{i\in Ag}, L'\rangle$ be two bubbles wrt to φ. Let $a \in \Sigma$. B' is an* a-successor *of B ($B \xrightarrow{a} B'$) if the following conditions hold:*

1. *$S' \subseteq S$ and for all $s \in S'$, $L(s) \cap \mathcal{P} = L'(s) \cap \mathcal{P}$.*
2. *For all $s \in S$, $\langle a\rangle\psi \in L(s)$ iff ($s \in S'$ and $\psi \in L'(s)$).*
3. *For all $s \in S'$, $[a]\psi \in L(s)$ iff $\psi \in L'(s)$.*
4. *(Perfect Recall): $R'_i = R_i \cap (S' \times S')$.*

Point 1 says that the set of states is decreasing when an observation a is made, and that the valuations do not change (for the surviving states). Point 2 says that any state s satisfying a diamond formula $\langle a\rangle\psi$ must survive after observing a, and then, must satisfy ψ. Point 3 says that if a state survives, same rule should apply for box formulas as well. Point 4 says that if an agent considers a state possible from the state s in the projected (residuated) model, she should consider it possible before the projection (Perfect Recall: $\langle a\rangle\hat{K}_i\psi \to \hat{K}_i\langle a\rangle\psi$). Now we introduce the notion of *bubble transition structure* (BTS) of a formula φ, which is a deterministic automaton where nodes are bubbles.

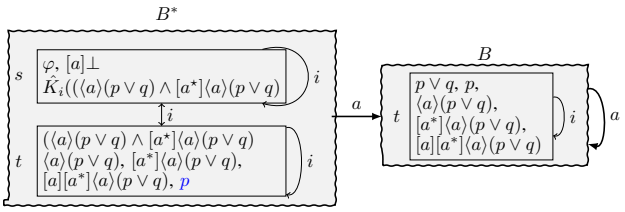


Figure 2: A BTS for $\varphi := [a]\bot \wedge \hat{K}_i(\langle a\rangle(p\vee q)\wedge[a^\star]\langle a\rangle(p\vee q))$. There are two bubbles: B^* and B. There are two abstract states: s and t. If a state appears in a bubble, it is labelled by a Hintikka set: for instance, s in B^* is labelled by $L^*(s) = \{\varphi, [a]\bot, \dots\}$.

Definition 19 (BTS). *Given a* POL *formula φ, a* bubble transition structure *(BTS) of φ is a tuple $\mathbb{B} = \langle \mathcal{B}, \delta\rangle$ where $\mathcal{B}$ is the (finite) set of all bubbles wrt φ, and $\delta : \mathcal{B}\times\Sigma \to \mathcal{B}\cup\{\ddagger\}$ is the transition function such that:*

1. *Some bubble $B^* = \langle S^*, \{R_i^*\}_{i\in Ag}, L^*\rangle \in \mathcal{B}$, called the initial bubble, is s. th. there is an $s \in S^*$, with $\varphi \in L^*(s)$*
2. *For all $B = \langle S, \{R_i\}_{i\in Ag}, L\rangle \in \mathcal{B}$, either $\delta(B,a) = \ddagger$ or, $\delta(B,a) = B^a$ where B^a is an a-successor of B.*
3. *For every $\langle\pi\rangle\psi \in L(s)$ for any $s \in S$ of any node $B = \langle S, \{R_i\}_{i\in Ag}, L\rangle \in \mathcal{B}$ in $\mathbb{B}$, there is a k length word $a_1a_2\dots a_k \in \mathcal{L}(\pi)$, a sequence of nodes (bubbles) $B^0 = B, B^1, \dots, B^k$, where each $B^j \in \mathcal{B}$, such that:*
   (a) $B^j = \delta(B^{j-1}, a_j)$, for all $1 \le j \le k$.
   (b) $s\in S^k$ and $\psi\in L^k(s)$, where $B_k = \langle S^k, \{R_i^k\}_{i\in Ag}, L^k\rangle$.

In other words, a BTS can be thought of as a structure where each node represents some residue of the model represented by the bubble B^* given in (1), which represents a model satisfying φ. (2) assumes that a model can have at most one residue structure for every letter $a \in \Sigma$. (3) says that if $\langle\pi\rangle\psi$ is satisfied at a state in a model, then there exists a model which is residuated on some $w \in \mathcal{L}(\pi)$ and the same state in the residuated model satisfies ψ. Note that a formula may have zero or multiple BTS's.

Example 20. *Consider the BTS $\mathbb{B}$ in Figure 2, where $\varphi := [a]\bot \wedge \hat{K}_i(\langle a\rangle(p\vee q)\wedge[a^\star]\langle a\rangle(p\vee q))$.*

- *All formulas appearing in the labels are in $FL(\varphi)$*
- *φ appears in the label of s in B^* (Definition 19 (1)).*
- *The bubble B is an a-observation successor of B^* as per Definition 18.*

By Definition 15, since $[a^\star]\langle a\rangle(p\vee q) \in L^*(t)$, $[a][a^\star]\langle a\rangle(p\vee q) \in L^*(t)$. Now by Definition 18 (2), since $\langle a\rangle(p\vee q) \in L^*(t)$, we have $(p\vee q) \in L(t)$. We also have $[a^\star]\langle a\rangle(p\vee q) \in L^*(t)$, which again, by definition of Hintikka set gives rise to $\{\langle a\rangle(p\vee q), [a][a^\star]\langle a\rangle(p\vee q)\} \subseteq L^*(t)$.

4.2 Completeness

Theorem 21. *If φ is satisfiable, then there is a BTS of φ.*

Proof Sketch. Suppose φ satisfiable. By Corollary 14, there exists $\mathcal{M}$ with at most $2^{|FL(\varphi)|}$ states and s such that $\mathcal{M}, s \models \varphi$. We create a transition system $\mathbb{B} = \langle\mathcal{B}, \delta\rangle$, where $\mathcal{B}$ contains exactly the bubbles B^w corresponding to $\mathcal{M}|_w$ for some $w \in \Sigma^\star$. The transition function δ is defined by

$$\delta(B,a) = \begin{cases} B^{wa}, \exists\ w \text{ such that } B = B^w \text{ and } \mathcal{M}|_{wa} \text{ exists} \\ \ddagger, \text{ otherwise} \end{cases}$$

It can be shown that $\mathbb{B}$ is a BTS of φ. For complete proof, see (Ghosh, Ghosh, and Schwarzentruber 2025).

4.3 Soundness

Theorem 22. *If there is a BTS of φ, then φ is satisfiable.*

Consider a BTS $\mathbb{B} = \langle\mathcal{B}, \delta\rangle$ of φ. The proof of Theorem 22 constitutes the construction of a pointed POL model $\mathcal{M}^\varphi, s_0$ satisfying φ out of $\mathbb{B}$. We construct the POL model $\mathcal{M}^\varphi = \langle S^\varphi, \{R_i^\varphi\}_{i\in Ag}, V^\varphi, Exp\rangle$ as follows: $S^\varphi = S^*$, $R_i^\varphi = R_i^*$ for all agents i, $V^\varphi(s) = L^*(s)\cap\mathcal{P}$, that is, the propositions of φ that labels in s, for any $s \in S^*$. The state s_0 is some state such that $\varphi \in L^*(s_0)$, where $B^* = \langle S^*, \{R_i^*\}_{i\in Ag}, L^*\rangle$ is the initial bubble.

It remains to define the expectation function Exp. For each state s, we define $Exp(s)$ as a regular expression that characterizes the language of the automaton $\mathcal{A}_s$ defined in the following:

- The set of states of $\mathcal{A}_s$ is $\mathcal{B}$;
- The transition function of $\mathcal{A}_s$ is given by δ;
- The initial state of $\mathcal{A}_s$ is B^*;
- The final states of $\mathcal{A}_s$ are the bubbles B containing s.

As a state s cannot resurrect, non-final states in $\mathcal{A}_s$ are absorbing: when a non-final state is reached by reading some word, it is impossible to reach a final state again.

Example 23. *Consider Figure 2. In $\mathcal{A}_t$, both B^* and B are final. We set $Exp(t) = aa^*$. In $\mathcal{A}_s$, only B^* is final. We set $Exp(s) = \epsilon$.*

The proof of Theorem 22 ends with the following claim, implying that φ is satisfiable.

Claim 24. $\mathcal{M}^\varphi, s_0 \models \varphi$.

Proof. We prove a more general result. We prove the following property $\mathcal{P}(\psi)$ by inducting on $\psi \in FL(\varphi)$:

$$\mathcal{P}(\psi): \begin{array}{l}\text{for all bubbles } B \in \mathcal{B}, \text{ for all states } s \in B, \text{ if} \\ \psi \in L(s) \text{ then } \mathcal{M}^\varphi|_w, s \vDash \psi \text{ for all words } w \in \Sigma^\star \\ \text{such that } B^* \xrightarrow{w}{}^* B\end{array}$$

where $B^* \xrightarrow{w}{}^* B$ means that there is a path from B^* to B by reading the word w in $\mathbb{B}$.

Base Case. The case $\psi = p$ is implied by the construction and point 1 of Definition 18.

Inductive Step. Based on the syntax of ψ:

- $\psi = \hat{K}_i\chi$. As $\psi \in L(s)$, by Definition 17, there is some t such that $(s,t) \in R_i$ and $\chi \in L(t)$. By IH, $\mathcal{M}^\varphi|_w, t \vDash \chi$. Due to point 4 (Perfect Recall) of Definition 18, the relation $(s,t) \in R_i$ is retained from B^* along the path $B^* \xrightarrow{w}{}^* B$. Consider the construction of $\mathcal{A}_s$. Since all bubbles having s in it are marked final, hence $Exp(s)\backslash w \neq \emptyset$ since $w \in \mathcal{L}(Exp(s))$. Hence

by construction and the previous argument, the relation is retained in $\mathcal{M}^\varphi$ as well since s is still retained in $\mathcal{M}^\varphi|_w$. Therefore $\mathcal{M}^\varphi, s \vDash \hat{K}_i\chi$.

- $\psi = \langle\pi\rangle\chi$. By point 3 of Definition 19, there is some $w' \in \Sigma^\star$ such that $\chi \in L'(s)$, where L' is the labelling function for a bubble B' and $B \xrightarrow{w'}{}^{*} B'$. Hence by IH, $\mathcal{M}^\varphi|_{ww'}, s \vDash \chi$. This implies $\mathcal{M}^\varphi|_w, s \vDash \langle\pi\rangle\chi$.

Hence the property $\mathcal{P}(\psi)$ is proved for any formula $\psi \in FL(\varphi)$ for any state s of any bubble B in BTS. In particular, $\mathcal{P}(\varphi)$ implies $\mathcal{M}^\varphi, s_0 \vDash \varphi$ since $\varphi \in L^*(s_0)$ of B^*. □

5 POL Satisfiability by DPDL

Finally, to provide an algorithm for checking satisfiability of POL, we now provide a translation of POL formulas to Deterministic Propositional Dynamic Logic (DPDL) formulas and use the complexity results of DPDL (Ben-Ari, Halpern, and Pnueli 1982a). Before proceeding further, let us have a brief look at the syntax and semantics of DPDL.

5.1 On DPDL

Given a countable set of atomic propositions $\mathcal{P}$, and a finite set of actions Σ, the language of DPDL is given by the following:

$$\varphi := \top \mid p \in \mathcal{P} \mid \varphi \vee \psi \mid \neg\varphi \mid \langle\pi\rangle\psi,$$

where π is regular expression over Σ.

A DPDL model is given by: $M = \langle W, \{\rightarrow_a\}_{a\in\Sigma}, V\rangle$, where, W is a finite set of states, $\rightarrow_a \subseteq W \times W$ is a binary relation such that for every $a \in \Sigma$, and for every $w \in W$, if $(w, w_1) \in \rightarrow_a$ and $(w, w_2) \in \rightarrow_a$ then $w_1 = w_2$, and $V : W \rightarrow 2^{\mathcal{P}}$ is the valuation function.

We extend the relation $\rightarrow_a$ to $\rightarrow_\pi$ for any general regular expression π in the usual way:

- $\rightarrow_{\pi_1+\pi_2} = \rightarrow_{\pi_1} \cup \rightarrow_{\pi_2}$
- $\rightarrow_{\pi_1;\pi_2} = \{(w, v) \mid \exists u \in W : (w, u) \in \rightarrow_{\pi_1} \wedge (u, v) \in \rightarrow_{\pi_2}\}$
- $\rightarrow_{\pi^\star} = \bigcup_{k\geq 0} \rightarrow_{\pi^k}$ where $\pi^k = \underbrace{\pi; \pi; \ldots; \pi}_{k \text{ times}}$

Given a DPDL model $M = \langle W, \{\rightarrow_a\}_{a\in\Sigma}, V\rangle$ and a DPDL formula φ, we define $M, s \vDash \varphi$ for some $s \in W$ as usual by induction on φ. We mention the modal case below that involves the language of the regular expression.

- $M, s \vDash \langle\pi\rangle\psi$ iff there is a word $w \in \mathcal{L}(\pi)$ such that $s \rightarrow_w t$ and $M, t \vDash \varphi$

To clarify the distinction between DPDL and PDL (Harel, Kozen, and Tiuryn 2000), let us consider the formula, $\langle a\rangle p \wedge \langle a\rangle\neg p$. The formula is satisfiable in PDL, but not in DPDL.

5.2 Translation of POL into DPDL

We introduce a translation from POL-formulas into DPDL-formulas. The idea is as follows. On the one hand, each observation operator $\langle\pi\rangle$ is directly simulated by its DPDL-dynamic operator counterpart. On the other hand, we encode each epistemic structure as a propositional theory.

Thanks to the filtration result (see Theorem 13), we know that a satisfiable formula φ has a POL model with at most $2^{|FL(\varphi)|}$ worlds. We can then pinpoint the worlds by labels $\ell, \ell', \ldots$ from the set of labels $\mathfrak{L}_0^\varphi = \{1, \ldots, 2^{|FL(\varphi)|}\}$.

We introduce special atomic formulas of the form $@\ell.\psi$ whose intuitive meaning is 'subformula ψ is true in the ℓ-th world'. We also introduce atomic propositions $R_i(\ell, \ell')$ whose intuitive meaning is 'the ℓ-th world is linked to the the ℓ'-th world by the relation R^i'. In addition, we introduce atomic proposition $surv(\ell)$ that intuitively says that the ℓ-th world has survived so far (meaning that the ℓ-th world is still compatible with the observations that have been seen so far).

Given a POL-formula φ, we define a DPDL-formula $sem(\varphi)$ that encodes the semantics of φ:

- $sem(p) = \bigwedge_{l\in\mathfrak{L}_0^\varphi}(@\ell.p \leftrightarrow \neg@\ell.\neg p)$
- $sem(\neg\psi) = \bigwedge_{l\in\mathfrak{L}_0^\varphi}(@\ell.\neg\psi \leftrightarrow \neg@\ell.\psi)$
- $sem(\psi \vee \psi') = \bigwedge_{l\in\mathfrak{L}_0^\varphi}(@\ell.(\psi \vee \psi') \leftrightarrow (@\ell.\psi \vee @\ell.\psi'))$
- $sem(\hat{K}_i\psi) = \bigwedge_{l\in\mathfrak{L}_0^\varphi}(@\ell.\hat{K}_i\psi \leftrightarrow (\bigvee_{l'\in\mathfrak{L}_0^\varphi}(R_i(\ell, \ell') \wedge surv(\ell') \wedge @\ell'.\psi)))$
- $sem(\langle\pi\rangle\psi) = \bigwedge_{l\in\mathfrak{L}_0^\varphi}(@\ell.\langle\pi\rangle\psi \leftrightarrow \langle\pi\rangle(@\ell.\psi \wedge surv(\ell)))$

Formula $sem(\varphi)$ is about the 'local' semantics of φ. For example, $sem(\psi \vee \psi')$ explains the semantics of $\vee$ in the stage of $\psi \vee \psi'$; the semantics of ψ and ψ' are taken care of by $sem(\chi)$ for the subformulas χ of ψ and ψ'. Formula $sem(\langle\pi\rangle\psi)$ reflects the fact that the semantics of $\langle\pi\rangle\psi$ is given by the DPDL-operator $\langle\pi\rangle$ for a given world ℓ that should survive. Formula $sem(\hat{K}_i\psi)$ expresses the semantics of $\hat{K}_i\psi$ (see Definition 6) in the propositional theory: the existence of a state is represented by the disjunction $\bigvee_{l'\in\mathfrak{L}_0\varphi}$, the relation constraint is represented by the propositional formula $R_i(\ell, \ell') \wedge surv(\ell')$, and the truth of ψ in the possible world by the proposition $@\ell'.\psi$.

Given a POL-formula φ, we also define the DPDL-formula $\mathcal{S}_\varphi$ as the conjunction of the following expressions:

1. $\bigwedge_{\psi\in FL(\varphi)}[\Sigma^\star]sem(\psi)$
2. $\bigwedge_{\ell\in\mathfrak{L}_0^\varphi}((@\ell.p \rightarrow [\Sigma^\star]@\ell.p) \wedge ((@\ell.\neg p \rightarrow [\Sigma^\star]@\ell.\neg p)))$
3. $\bigwedge_{\ell,\ell'}((R_i(\ell, \ell') \rightarrow [\Sigma^\star]R_i(\ell, \ell')) \wedge (\neg R_i(\ell, \ell')) \rightarrow [\Sigma^\star]\neg R_i(\ell, \ell'))$
4. $\bigwedge_{i\in Agt}\bigwedge_{\ell\in\mathfrak{L}_0^\varphi} R_i(\ell, \ell)$
5. $\bigwedge_{i\in Agt}\bigwedge_{\ell\in\mathfrak{L}_0^\varphi}\bigwedge_{\ell'\in\mathfrak{L}_0^\varphi}(R_i(\ell, \ell') \rightarrow R_i(\ell', \ell))$
6. $\bigwedge_{i\in Agt}\bigwedge_{\ell,\ell',\ell''\in\mathfrak{L}_0^\varphi}((R_i(\ell, \ell') \wedge R_i(\ell', \ell'') \rightarrow R_i(\ell, \ell''))$
7. $[\Sigma^\star]\bigwedge_\ell\bigwedge_{a\in\Sigma}\langle a\rangle\top$
8. $[\Sigma^\star]\bigwedge_\ell\bigwedge_{a\in\Sigma}(\neg surv(\ell) \rightarrow \langle a\rangle\neg surv(\ell))$

Point 1 says that for all subformulas ψ, the semantics of ψ is enforced anywhere in the DPDL model. Point 2 says that the truth value of any atomic proposition p does not change at a given label/state ℓ when observations are made. Point 3 says that the epistemic relations do not change. Note that labels ℓ, ℓ' may not survive, that is handled by the proposition $surv(\ell)$. Points 4, 5, and 6 encode, respectively, the

reflexivity, symmetry, and transitivity of the epistemic relation. Point 7 says that an a-successor always exists (even if no labels survive! The surviving mechanism is fully handled by propositions $surv(\ell)$). Point 8 says that a label ℓ that is dead remains dead.

Definition 25 (Translation from POL to DPDL)**.** *Given a* POL*-formula φ, we define* $tr(\varphi) := surv(\ell_0) \wedge @\ell_0.\varphi \wedge \mathcal{S}_\varphi$.

In the translation $tr(\varphi)$ above, we say that the state ℓ_0 must have survived (the empty list of observations so far), formula φ should be true in ℓ_0, and $\mathcal{S}_\varphi$ forces the semantics to be well-behaved.

Proposition 26. *Given* POL *formula φ, $tr(\varphi)$ is computable in exponential time in the size of φ.*

Proof. Given a POL formula φ, there are at most $O(|\varphi|)$ many formulas in $FL(\varphi)$, and by filtration (Theorem 13) there can be at most $O(2^{|\varphi|})$ many unique labels. Therefore the time taken and the size of the formula $tr(\varphi)$ is at most $O(|\varphi| \times 2^{|\varphi|})$. □

Proposition 27. *φ is* POL*-satisfiable iff $tr(\varphi)$ is* DPDL*-satisfiable.*

From Propositions 26 and 27, and from the fact that DPDL satisfiability is EXPTIME-complete (Ben-Ari, Halpern, and Pnueli 1982a) we get:

Theorem 28. POL*-satisfiability is in* 2EXPTIME.

Proof. Here is a double-exponential algorithm for testing the satisfiability of φ in POL:

- compute $tr(\varphi)$ (exponential-time in φ)
- test whether $tr(\varphi)$ is DPDL-satisfiable (exponential time in $tr(\varphi)$, so double-exponential time in φ).

The result follows from the algorithm above. □

6 Lower Bound

In the following we provide a lower bound for the satisfiability problem of POL, and show that:

Theorem 29. POL *satisfiability is* 2EXPTIME*-complete.*

The upper bound was shown in Theorem 28. To prove 2EXPTIME-hardness in Theorem 29, consider any 2EXPTIME problem A. As AEXPSPACE $=$ 2EXPTIME (Chandra, Kozen, and Stockmeyer 1981), there is an alternating Turing machine M deciding A in exponential space $e(|x|)$ where x is the input, and $|x|$ is its length/size. The accepting and rejecting state are respectively denoted by q_{acc} and q_{rej}.

We represent a configuration as a word of symbols. A *symbol* can be either a letter (0, 1, or ␣) written on the tape, or a pair qa where q is a state of M and a is a letter, or a special symbol #. The set of symbols is denoted by Sym. We suppose that the state symbol precedes the letter on which the head is. All configuration words start and finish by the special symbol #. For instance,

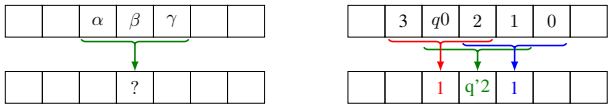
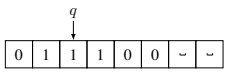
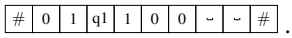


Figure 3: Successor function in a Turing machine: from three consecutive symbols $\alpha\beta\gamma$ the successor function tells the symbol written in the middle (?). On the right, we give an example of a a-transition 'if the machine is in state q with 0 under the head, then write 1 and move to right and go into state q''. In particular, we have $\mathsf{succ}_a(3\ q0\ 2) = 1, \mathsf{succ}_a(q0\ 2\ 1) = q'$ and $\mathsf{succ}_a(021) = 2$.

the configuration  where the tape contains $011100␣␣\ldots$, the machine is in state q, and the head is under the third left-most cell is represented by the word  .

Without loss of generality (w.l.o.g.), we suppose that the machine M switches between universal and existential states and starts with an existential state. Also, w.l.o.g. each configuration has at most two successor configurations. So we use two functions succ_a and succ_b such that given three consecutive symbols $\alpha\beta\gamma \in Sym^3$, $\mathsf{succ}_a(\alpha\beta\gamma)$ (resp. $\mathsf{succ}_b(\alpha\beta\gamma)$) is the symbol in the middle position after the first transition (resp. the second transition), see Figure 3. Taking a transition is modeled by the program $a \cup b$. Figure 4 explains the idea behind the reduction, namely how to represent a computation tree inside a POL model.

Encoding a configuration. We encode a superposition of *three* configurations into an epistemic structure, and then say that they are equal.

As there are two agents, thus two epistemic modalities K_i and K_j we can simulate a standard K modal logic $\Box$. For the rest of the proof, we consider such a modality $\Box$ and its dual $\Diamond$. We also introduce $\Box^k\varphi$ for $\Box \ldots \Box\varphi$ where $\Box$ is repeated k times, and $\Box^{\leq k}\varphi$ for $\varphi \wedge \Box\varphi \wedge \cdots \wedge \Box^k\varphi$.

We create a formula that ensures the existence of a binary tree in the epistemic structure. A *position* in a word is a number in $\{0, 1, \ldots, e(|x|)\}$. Each leaf is tagged with a 3-tuple of positions pos_1, pos_2, pos_3 in the tape, and a 3-tuple of symbols α, β, γ, written respectively at position pos_1, j, k, in respectively the first, second and third configuration – the values of pos_1, pos_2, k by the truth values of propositions $p_1, \ldots, p_n, p_{n+1}, \ldots, p_{2n}$ and $p_{2n+1}, \ldots, p_{3n}$, respectively. A position in the tape is a number between 0 and $e(|x|) - 1$. As $e(|x|)$ is exponential in $|x|$, the integer n above is polynomial in $|x|$. Our binary tree branches over the values for atomic propositions $p_1, \ldots, p_{3n}$. To do so, we use the modal logic formula given in (Blackburn, de Rijke, and Venema 2001) and already used in (Chakraborty et al. 2023):

$$\bigwedge_{\ell<6n} \Box^\ell \left(\Diamond p_\ell \wedge \Diamond \neg p_\ell \wedge \bigwedge_{m<\ell} \begin{array}{l} (p_m \rightarrow \Box p_m) \wedge \\ (\neg p_m \rightarrow \Box \neg p_m) \end{array} \right) \quad (1)$$

In order to select some positions, we make the values of propositions p_m 'observable'. To do that we introduce new

observation symbol $p_m, \bar{p}_m \in \Sigma$ and the constraints:

$$[(a\cup b)^*]\Box^{3n}[\Sigma^*](p_m \leftrightarrow \langle p_m\rangle\top \wedge [\bar{p}_m]\bot) \quad (2)$$

$$[(a\cup b)^*]\Box^{3n}[\Sigma^*](\neg p_m \leftrightarrow \langle \bar{p}_m\rangle\top \wedge [p_m]\bot) \quad (3)$$

In other words, being able to observe $p_m \in \Sigma$ (resp. $\bar{p}_m \in \Sigma$) means that p_m is true (resp. false).

For each symbol α, we introduce observations $1{:}\alpha$, $2{:}\alpha$, $3{:}\alpha$. They are observable when symbol α is written in the current cell of respectively the first, second, and third configuration. The following three formulas say that there is a unique symbol written at each position pos_1, pos_2, pos_3:

$$[(a\cup b)^*]\Box^{3n} \bigoplus_{\alpha\in Sym} \langle i{:}\alpha\rangle\top \quad (4)$$

for $i = 1..3$, and where $\bigoplus$ is the XOR operator.

The two following formulas say that the symbols of the cells do not change when observing some positions:

$$[(a\cup b)^*]\Box^{3n}\Big(\bigwedge_{\alpha\in Symb} \langle i{:}\alpha\rangle\top \rightarrow [(p_1 \cup \bar{p_1} \cup \cdots \cup p_{3n} \cup \bar{p}_{3n})^*]\langle i{:}\alpha\rangle\top\Big) \quad (5)$$

$$[(a\cup b)^*]\Box^{3n}\Big(\bigwedge_{\alpha\in Symb} \neg\langle i{:}\alpha\rangle\top \rightarrow [(p_1 \cup \bar{p_1} \cup \cdots \cup p_{3n} \cup \bar{p}_{3n})^*]\neg\langle i{:}\alpha\rangle\top\Big) \quad (6)$$

We now say that for all $i = 1..3$, all leafs with the same position pos_i contains the same symbol in the i-th configuration:

$$[(a\cup b)^*][\text{choose } pos_i] \bigvee_{\alpha\in Sym} \Box^{3n}\langle i{:}\alpha\rangle\top \quad (7)$$

where choose pos_1, choose pos_2, choose pos_3 are respectively the programs $(p_1 \cup \bar{p}_1)\ldots(p_n \cup \bar{p}_n)$, $(p_{n+1} \cup \bar{p}_{n+1})\ldots(p_{2n} \cup \bar{p}_{2n})$ and $(p_{2n+1} \cup \bar{p}_{2n+1})\ldots(p_{3n} \cup \bar{p}_{3n})$.

Finally we say that the *three* configurations are equal. When two positions - say pos_1 and pos_2 - are equal, then the symbol located at pos_1 in the first configuration and the symbol located at pos_2 in the second configuration are equal. More generally, for $i, j = 1..3$, $i < j$:

$$[(a\cup b)^*]\Box^{3n}(pos_i{=}pos_j) \rightarrow \bigvee_{\alpha\in Sym} \langle i{:}\alpha\rangle\top \wedge \langle j{:}\alpha\rangle\top \quad (8)$$

where $pos_1{=}pos_2$ is a Boolean formula saying that $p_1, \ldots, p_{n-1}$ and $p_{n+1}, \ldots, p_{2n}$ encode the same number, and similar others.

Presence of the complete binary tree after transitions. The following formula says that the existence of the a-transition (a can be observed at the root) implies that a can be observed at all nodes of the binary tree. Also if there is no a-transition, then a is not observable at all nodes of the tree. Same for b. This is captured by the following scheme for ψ being $\langle a\rangle\top$, $\neg\langle a\rangle\top$, $\langle b\rangle\top$, $\neg\langle b\rangle\top$:

$$[(a\cup b)^*](\psi \rightarrow \Box^{\leq 6n}\psi) \quad (9)$$

Leftmost and rightmost cells. We impose that the leftmost and rightmost cells always contain symbol $\#$.

$$[(a\cup b)^*]\Box^{3n}(pos_1 = 0 \rightarrow \langle 1{:}\#\rangle\top) \quad (10)$$

$$[(a\cup b)^*]\Box^{3n}(pos_1 = e(|x|) - 1 \rightarrow \langle 1{:}\#\rangle\top) \quad (11)$$

Initial configuration. At position 1, there is the initial state q_0. At position 2 is the first letter x_1 of x. The last letter x_n is at position $n + 1$.

$$\Box^{3n}(pos_1{=}1) \rightarrow \langle 1{:}q_0x_1\rangle\top \wedge (pos_1{=}2) \rightarrow \langle 1{:}x_2\rangle\top \cdots \wedge (pos_1{=}n) \rightarrow \langle 1{:}x_n\rangle\top \quad (12)$$

After the word, we have the blank symbol $\sqcup$:

$$\Box^{3n}(pos_1 \geq n + 1 \wedge pos_1 < e(|x|) - 1) \rightarrow \langle 1{:}\sqcup\rangle\top \quad (13)$$

Transitions. W.l.o.g. we suppose the machine stops after writing $\sqcup$. We define the two formulas $\mathsf{accepted} := \Diamond^{3n}\langle 1{:}q_{acc}\sqcup\rangle\top$ and $\mathsf{rejected} := \Diamond^{3n}\langle 1{:}q_{rej}\sqcup\rangle\top$ meaning that the current configuration is respectively an accepting or rejecting one. We set $\mathsf{final} := \mathsf{accepted} \vee \mathsf{rejected}$.

We encode a- and b-transitions as follows. We focus on leaves containing consecutive cells: a cell at position pos_1 in the 1st configuration, the cell at $pos_1 + 1$ in the 2nd configuration, and the cell at $pos_1 + 2$ in the 3rd configuration. The superposition of the three configurations helps us to have access to three symbols α, β, γ in consecutive cells. We then use succ_a and succ_b to get the next middle symbol (at position $pos_1 + 1$ in the 2nd configuration).

$$\begin{array}{l}[(a\cup b)^*](\neg\mathsf{final} \rightarrow \Box^{3n}\begin{pmatrix} pos_2{=}pos_1{+}1\wedge \\ pos_3{=}pos_2{+}1\end{pmatrix} \rightarrow \\ \bigwedge_{\alpha\beta\gamma\in Sym^3}(\langle 1{:}\alpha\rangle\top \wedge \langle 2{:}\beta\rangle\top \wedge \langle 3{:}\gamma\rangle\top) \\ \rightarrow (\langle a\rangle\langle 2{:}\mathsf{succ}_a(\alpha\beta\gamma)\rangle\top \wedge \langle b\rangle\langle 2{:}\mathsf{succ}_b(\alpha\beta\gamma)\rangle\top)\end{array} \quad (14)$$

In a terminal configuration, the execution stops meaning that we do not take transitions anymore ($[a\cup b]\bot$).

$$[(a\cup b)^*](\mathsf{final} \rightarrow [a\cup b]\bot) \quad (15)$$

Universal and existential configurations. We introduce an observation symbol $\exists$, which is observable iff the current configuration is existential. Existential and universal configurations are always alternating:

$$[(a\cup b)^*](\langle\exists\rangle\top \rightarrow [a\cup b]\neg\langle\exists\rangle\top) \quad (16)$$

$$[(a\cup b)^*](\neg\langle\exists\rangle\top \rightarrow [a\cup b]\langle\exists\rangle\top) \quad (17)$$

Winning condition. The two following formulas explain what winning means at the final configurations:

$$[(a\cup b)^*](\mathsf{accepted} \rightarrow \langle win\rangle\top) \quad (18)$$

$$[(a\cup b)^*](\mathsf{rejected} \rightarrow \neg\langle win\rangle\top) \quad (19)$$

The two following formulas explain what winning means at a non-terminal configuration:

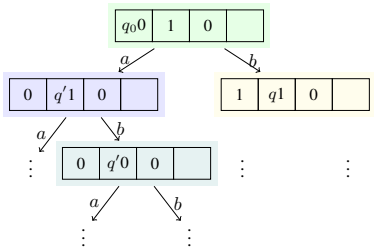
$$[(a\cup b)^*]((\neg\mathsf{final} \wedge \neg\langle\exists\rangle\top) \rightarrow (\langle win\rangle\top \leftrightarrow [a\cup b]\langle win\rangle\top) \quad (20)$$

$$[(a\cup b)^*]((\neg\mathsf{final} \wedge \langle\exists\rangle\top) \rightarrow (\langle win\rangle\top \leftrightarrow \langle a\cup b\rangle\langle win\rangle\top) \quad (21)$$

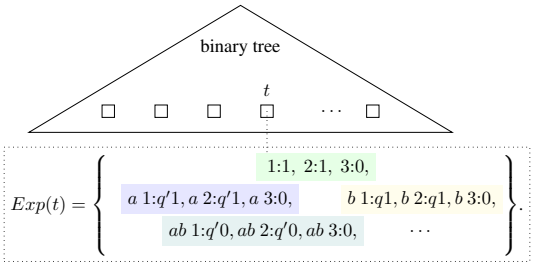
The following formula says that the initial configuration should be winning and is existential:

$$\langle win\rangle\top \wedge \langle\exists\rangle\top \quad (22)$$

We define $tr(x)$ to be the conjunction of formulas (1-22) and the two following propositions conclude the proof.



(a) Computation tree of M on input $x = 01$. Each node of the three is a configuration of the machine. The root contains the initial configuration. The branching directions are called a and b.



(b) The corresponding POL model. The epistemic structure encodes a binary tree. Each leaf t corresponds to three cell positions. The figure examplifies a leaf t where the first and second position are both the second cell while the third position is the third cell.

Figure 4: A computation tree and its corresponding POL model.

Proposition 30. *$tr(x)$ is computable in poly-time in $|x|$.*

Proposition 31. *x is A-positive iff $tr(x)$ is* POL*-satisfiable.*

We thus reduce any 2EXPTIME-problem A to the satisfiability problem of POL, proving the latter to be 2EXPTIME-hard, thus proving Theorem 29.

7 Related Work

Propositional Dynamic Logics. In POL, a residuated model $\mathcal{M}|_w$ can have at most a unique successor $\mathcal{M}|_{wa}$ for a letter a in the alphabet. This gives rise to a Σ-labelled transition structure equivalent to a model where each node is an epistemic skeleton of some $\mathcal{M}|_w$. Thus we get a natural connection with a Deterministic Propositional Dynamic Logic (DPDL) (Ben-Ari, Halpern, and Pnueli 1982b) model structure. But, there is a significant difference: each node in our constructed transition structure is an epistemic model, whereas in DPDL, it is a propositional valuation. To the best of our knowledge, no such deterministic structures involving epistemic constructions have been studied beforehand.

Epistemic Propositional Dynamic Logic (EPDL) has been studied in (Li 2018). As in POL, each node of an EPDL model is an epistemic structure, and it assumes *perfect recall*. However, the transitions in EPDL are non-deterministic ones. Moreover, in EPDL, valuations may change when actions are executed, while that is not the case when observations are made in POL.

Another study on a PDL-like logic with epistemic operators (Heinemann 2007) concerns single agent knowledge formulas, where, verification of *only* knowledge formulas in the regular expressions is considered. Although, they have given a hardness result (EXPTIME-hard) that follows from PDL, they do not have any matching upper bound.

Temporal Logics. Since we are dealing with sequences of observations and how the expected observation expressions get residuated, there is a subtle temporal aspect to POL. In $\mathsf{LTL_K}$, the interaction between time and knowledge assumes *perfect recall* and *synchronous* rules, which also hold in POL. But, there are some differences in the *linearity* aspects. More precisely, since POL deals with multiple letters in the transition alphabet, knowledge does not change linearly, in the sense that, knowledge can change depending on the observations that occur. At a single point of occurrence this can be any one of the letters in the alphabet. We note here that a valuation in a state in the next temporal transition in $\mathsf{LTL_K}$ may change, whereas such a valuation remains consistent in POL. Thus, $\neg p \wedge Xp$ is satisfiable in $\mathsf{LTL_K}$, whereas $\neg p \wedge \langle a \rangle p$ is not satisfiable in POL, which leads to a difference in expressive power. From the complexity viewpoint, the satisfiability problem of $\mathsf{LTL_K}$ is non-elementary, whereas for POL, it is 2EXPTIME-complete. It is interesting to note that both $\mathsf{LTL_K}$ and POL can express that *after some finite sequence of observations, φ holds*, but only POL can express that *after some even occurrences of observations, φ holds*. Resolving the intricate relationship between $\mathsf{LTL_K}$ and POL needs further study.

Since knowledge updates *branch out* in POL depending upon the action observed, Computational Tree Logic with epistemic operators (CTLK) (Dima 2008) forms a close neighbour, whose satisfiability problem turns out to be undecidable. We note here that unravelled POL models create indistinguishability within the nodes (distinct POL models) themselves, whereas, indistinguishability in a CTLK model occurs among nodes across the tree.

8 Perspectives

Our main contribution has been to show that the satisfiability problem of POL (with Kleene star) is 2EXPTIME-complete. Now, we plan to study more tractable fragments of POL, and also more expressive extensions of POL, for instance, when expectations are context-free grammars.

A dynamic extension of POL mentioned in (van Ditmarsch et al. 2014), Epistemic Protocol Logic (EPL) is yet to be studied from the computational viewpoint. This logic is similar to DEL: EPL also comes with operators that have pointed event or action model like DEL, which *assigns* expectations to the POL models. Such action models deal with PDL-like regular expressions with Boolean verifiers. Complexity studies for the logic are open problems.

Acknowledgements

This work was supported by the ANR EpiRL project ANR-22-CE23-0029.

References

Aucher, G., and Bolander, T. 2013. Undecidability in epistemic planning. In *Proceedings of the Twenty-Third International Joint Conference on Artificial Intelligence, IJCAI 2013, Beijing, China, 3-9 August 2013*, 27–33. ijcai.org.

Ben-Ari, M.; Halpern, J. Y.; and Pnueli, A. 1982a. Deterministic propositional dynamic logic: Finite models, complexity, and completeness. *J. Comput. Syst. Sci.* 25(3):402–417.

Ben-Ari, M.; Halpern, J. Y.; and Pnueli, A. 1982b. Deterministic propositional dynamic logic: Finite models, complexity, and completeness. *J. Comput. Syst. Sci.* 25(3):402–417.

Blackburn, P.; de Rijke, M.; and Venema, Y. 2001. *Modal Logic*, volume 53 of *Cambridge Tracts in Theoretical Computer Science*. Cambridge University Press.

Bolander, T. 2017. A gentle introduction to epistemic planning: The DEL approach. In Ghosh, S., and Ramanujam, R., eds., *Proceedings of the Ninth Workshop on Methods for Modalities, M4M@ICLA 2017, Indian Institute of Technology, Kanpur, India, 8th to 10th January 2017*, volume 243 of *EPTCS*, 1–22.

Chakraborty, S.; Ghosh, A.; Ghosh, S.; and Schwarzentruber, F. 2022. On verifying expectations and observations of intelligent agents. In *Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence, IJCAI 2022, Vienna, Austria, 23-29 July 2022*, 2568–2574. ijcai.org.

Chakraborty, S.; Ghosh, A.; Ghosh, S.; and Schwarzentruber, F. 2023. On simple expectations and observations of intelligent agents: A complexity study. In Marquis, P.; Son, T. C.; and Kern-Isberner, G., eds., *Proceedings of the 20th International Conference on Principles of Knowledge Representation and Reasoning, KR 2023, Rhodes, Greece, September 2-8, 2023*, 136–145.

Chandra, A. K.; Kozen, D.; and Stockmeyer, L. J. 1981. Alternation. *J. ACM* 28(1):114–133.

Dima, C. 2008. Revisiting satisfiability and model-checking for CTLK with synchrony and perfect recall. In Fisher, M.; Sadri, F.; and Thielscher, M., eds., *Computational Logic in Multi-Agent Systems, 9th International Workshop, CLIMA IX, Dresden, Germany, September 29-30, 2008. Revised Selected and Invited Papers*, volume 5405 of *Lecture Notes in Computer Science*, 117–131. Springer.

Fagin, R.; Halpern, J. Y.; Moses, Y.; and Vardi, M. Y. 1995. *Reasoning About Knowledge*. MIT Press.

Ghallab, M.; Nau, D. S.; and Traverso, P. 2004. *Automated planning - theory and practice*. Elsevier.

Ghosh, A.; Ghosh, S.; and Schwarzentruber, F. 2025. Reasoning about knowledge on regular expressions is 2exptime-complete. *CoRR* abs/2508.09784.

Harel, D.; Kozen, D.; and Tiuryn, J. 2000. *Dynamic Logic*. MIT Press.

Heinemann, B. 2007. A pdl-like logic of knowledge acquisition. In Diekert, V.; Volkov, M. V.; and Voronkov, A., eds., *Computer Science – Theory and Applications*, 146–157. Berlin, Heidelberg: Springer Berlin Heidelberg.

Li, Y. 2018. Tableaux for a combination of propositional dynamic logic and epistemic logic with interactions. *J. Log. Comput.* 28(2):451–473.

Miller, J. S., and Moss, L. S. 2005. The undecidability of iterated modal relativization. *Stud Logica* 79(3):373–407.

Smullyan, R. 1995. *First-order Logic*. Dover books on advanced mathematics. Dover.

van Ditmarsch, H., and French, T. 2022. Quantifying over boolean announcements. *Log. Methods Comput. Sci.* 18(1).

van Ditmarsch, H.; Ghosh, S.; Verbrugge, R.; and Wang, Y. 2014. Hidden protocols: Modifying our expectations in an evolving world. *Artif. Intell.* 208:18–40.

van Ditmarsch, H. P.; van der Hoek, W.; and Kooi, B. P. 2007. Dynamic epistemic logic and knowledge puzzles. In Priss, U.; Polovina, S.; and Hill, R., eds., *Conceptual Structures: Knowledge Architectures for Smart Applications, 15th International Conference on Conceptual Structures, ICCS 2007, Sheffield, UK, July 22-27, 2007, Proceedings*, volume 4604 of *Lecture Notes in Computer Science*, 45–58. Springer.